International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

A Survey on Advanced Encryption Standard

Sandeep Kumar Rao[1], Dindayal Mahto[2], Dr. Danish Ali Khan[3]

[1, 2, 3]National Institute of Technology, Jamshedpur, India

Abstract: ***Rijndael's Advanced Encryption Standard (AES) is the block cipher based symmetric-key cryptography to protect the sensitive information. The key sizes of AES are 128, 192, 256 bits. AES is based on substitution-permutation strategy. It is accepted by NIST in 2001 after the five year of security evaluation. It is highly secured and efficient than Data Encryption Standard (DES) and other symmetric-key cryptographic algorithms. This paper depicts all the valuable work done on the Advanced Encryption Standard since it is accepted by National Institute of Standards and Technology (NIST).***

Keywords: AES, DES, Composite Field Arithmatic(CFA), Field Programmable Gate Array(FPGA), Correlation Power Analysis(CPA)

1. Introduction

In 1997, NIST wished to form a successor of DES after some security flaws in DES [3]. Two conferences were held (AES1 in August 1998 and AES2 in March 1999) and the motive was not only the security but also the performance in various aspects of settings [3]. In October 2000, Rijndael algorithm for encryption/decryption is selected and after the five years of long security and performance testing [4], is accepted by the U.S government in 2001.

The AES was published in 2001 by NIST as the symmetric block cipher algorithm and become the successor of DES as approved standard.

In AES, cipher takes block size of 128 bits for in both hardware and software implementation [5]. AES block size is fixed that is 128 bits and key sizes of 128,192 and 256 bits respectively but Rijndael's block sizes and key sizes are multiple of 32 bits with a minimum of 128 bits [1].

The block sizes have a limitation of 256 bits but key sizes are not fixed theoretically.AES uses 128,192 and 256 bits key sizes and 10, 12 and 14 round respectively. There has been attack on 7 rounds for 128-bit, 8 rounds for 192-bit and 9 rounds for 256-bit keys[6].

AES is highly structured and efficient algorithm to protect the classified information at the highest secure level[7].

2. Structure of Advanced Encryption Standard

NIST accepted the AES as a Federal Information processing Standard (FIPS)-197. The 128 bits inputs are arranged in block of bytes using 4×4 square matrix. The bytes processing is defined in the Galois Field GF(2^8). There are specified repeated steps involved in the each round of encryption and inverse steps are involved to getting back original plaintext[6,7].

The following steps involved in the encryption process:
1) Initial Round: AddRoundKey
2) Rounds: SubBytes, ShiftRows, MixColumns, and AddRoundKey
3) Last Round: SubBytes, ShiftRows, and AddRoundKey

AddRoundKey:
States's each byte is combined with block of the round key XORed with input block operation.

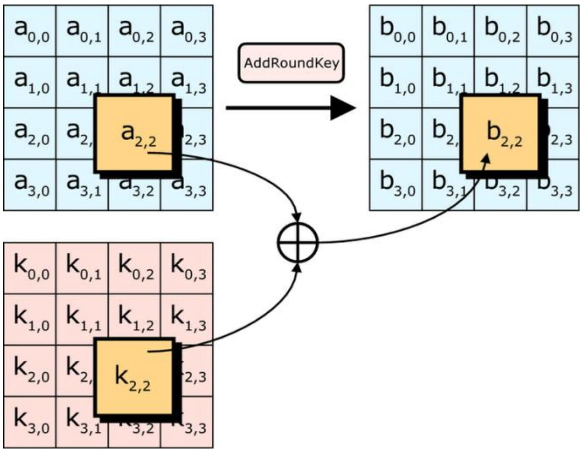

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
AddRoundKey
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
$\oplus$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$
$k_{1,0}$ $k_{1,1}$ $k_{1,2}$ $k_{1,3}$
$k_{2,0}$ $k_{2,1}$ $k_{2,2}$ $k_{2,3}$
$k_{3,0}$ $k_{3,1}$ $k_{3,2}$ $k_{3,3}$

Figure 1: Add round Key

SubByes:
It's a non-linear substitution step in which each byte is replaced with another according to lookup table that is S-Box.

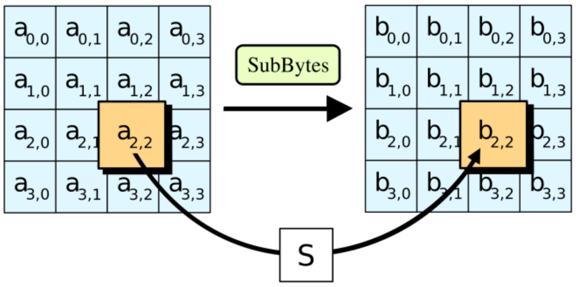
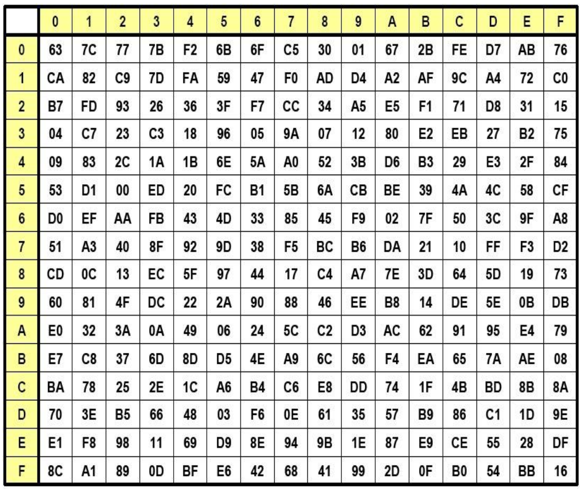

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

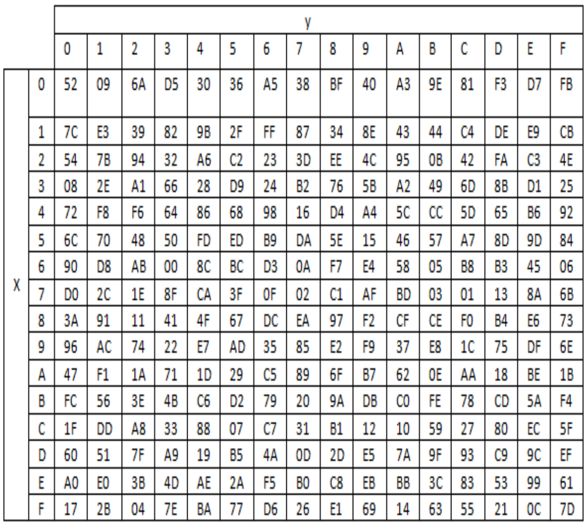
Figure 2: Substitute Bytes



	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
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

(a)S-Box

Figure 3: S-Boxes (use for substitutes bytes in encryption process)



		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
X	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

(b)Inverse S-box

Figure 4: Inverse S-Box (use for substitutes bytes in decryption process)

ShiftRows:
It's a transposition step in which the last three rows of the state is shifted cylically a certain round of steps.

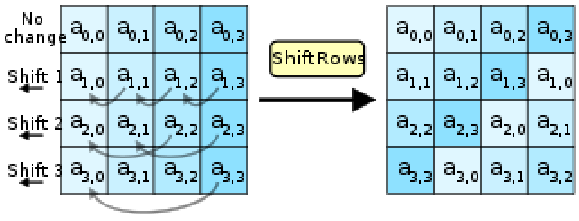

No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

Figure 5: Shifting row

MixColumns:
It's a mixing operation that operates on the Columns of the state, combining the four bytes in each column.

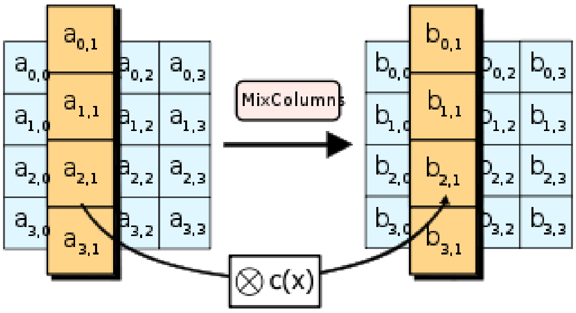

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
MixColumns
$\otimes c(x)$
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

Figure 6: Mixing Of column

AES Parameters:

Key Size(bits)	128	192	256
Plaintext Block Size(bits)	128	128	128
Number of Rounds	10	12	14
Round Key Size(bits)	128	128	128
Expanded Key Size(words)	44	52	60

3. AES Encryption and Decryption

a) AES most important feature is that it is not a Feistal structure. In Feistal structure half the part of data block is used to modify the other half of data block and so then the half are swapped.

b) AES process the entire data block using a single square matrix during each round using substitutions and permutation.

c) The key is expanded into array of 44 (32-bit words),w_i.

d) Four steps are used,one of permutation and three of substitution
   i) Sustitute bytes
   ii)ShiftRows
   iii) Mixcolumns
   iv) AddRound Key

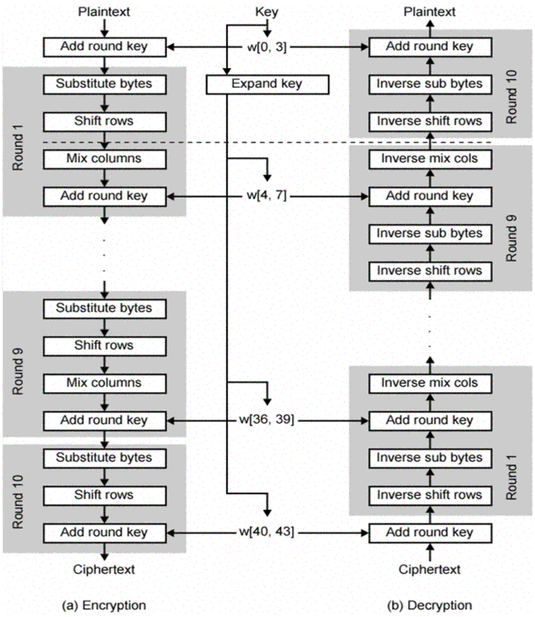


Figure 7: AES encryption and decryption

Fig.7 shows the overall encryption and decryption process

4. S-Box Enhancement and modification:

4.1 General Modification

a) Algabraic representation of Rijndael:
Rijndael block cipher uses the closed algebraic formulae that highly structured and very simple than any other algebraic formulation of block cipher that we know[8].

Rijndael S-Box can be written as an equation form that is

$$S(x)= w_8 + \sum_{d=0}^{7} w_8 x^{255-2^d}$$

W_8 is a constant and varies from w_0 to w_7.

Since in every round except last round, the output of of S-Box is multiplied by Maximum Distance Separable (MDS) matrix and after that key is added.

Due to linearity of MDS matrix and key addition w_8 constant can be replaced by some constant.

Now the equation will be in following form

$$S(x)= \sum_{d=0}^{7} w_d x^{255-2^d}$$

here modified key schedule is used so it gives x^{255} =1,expect for all x=0.

So the equation is simplified as

$$S(x)= \sum_{d=0}^{7} w_d x^{2^d}$$

After byte substitution,ShiftRow operation, Mixcolumns, and key addition the equation
Can be written as

$$a_{i,j}^{(r+1)} = k_{i,j}^{(r)} + \sum_{\substack{e_r \in \mathcal{E} \\ d_r \in D}} \frac{w_{i,e_r,d_r}}{\left(a_{e_r,e_r+j}^{(r)}\right)^{2^{d_r}}}$$

$a_{i,j}^{(r+1)}$ =byte at position (i,j) at final round,

$k_{i,j}^{(r)}$ = round key at position(i,j),
e_r=constant term in MixColumn step,
d_r=constant term in initial round,
$\mathcal{E}$ =(0,…3)
D =(0,…7)

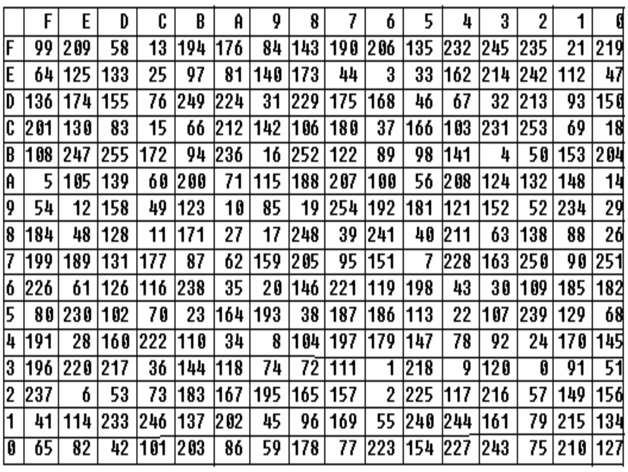
This algebraic representation for Rijndael still do not have any attack and it works properly.

b) S-Box Complexity
Rijandael's S-Box is very simple and it involves only 9 terms without any particular reason. So it is taken as a open challenge.
AES S-Box complexity can be increased to 255 terms to increase high reliable security of AES[9].

Algebraic expression of AES S-box is:

y = '05'x^{254} + '09'x^{253} + 'f9'x^{251} + '25'x^{247}+ 'f4'x^{239}+ '01'x^{223}+ 'b5'x^{191}+ '8f'x^{127}+0x63

After the improvement complexity can be increased from 9 to 255 terms. In this case complexity as well as security and performance of former algorithm is improved. Later algorithm is more resisting against the cryptanalysis attacks.



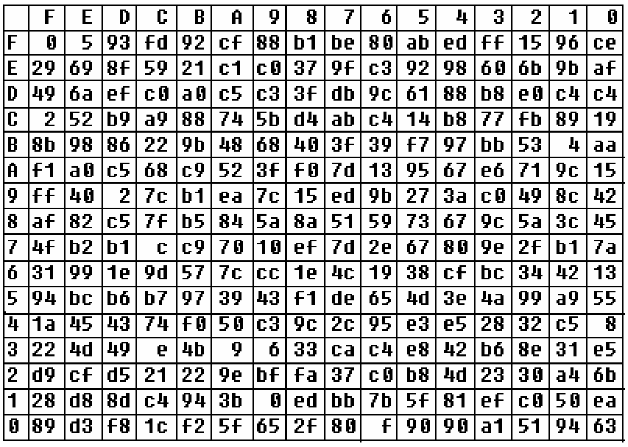
	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
F	99	209	58	13	194	176	84	143	190	206	135	232	245	235	21	219
E	64	125	133	25	97	81	140	173	44	3	33	162	214	242	112	47
D	136	174	155	76	249	224	31	229	175	168	46	67	32	213	93	150
C	201	130	83	15	66	212	142	106	180	37	166	103	231	253	69	18
B	108	247	255	172	94	236	16	252	122	89	98	141	4	50	153	204
A	5	105	139	60	200	71	115	188	207	100	56	208	124	132	148	14
9	54	12	158	49	123	10	85	19	254	192	181	121	152	52	234	29
8	184	48	128	11	171	27	17	248	39	241	40	211	63	138	88	26
7	199	189	131	177	87	62	159	205	95	151	7	228	163	250	90	251
6	226	61	126	116	238	35	20	146	221	119	198	43	30	109	185	182
5	80	230	102	70	23	164	193	38	187	186	113	22	107	239	129	68
4	191	28	160	222	110	34	8	104	197	179	147	78	92	24	170	145
3	196	220	217	36	144	118	74	72	111	1	218	9	120	0	91	51
2	237	6	53	73	183	167	195	165	157	2	225	117	216	57	149	156
1	41	114	233	246	137	202	45	96	169	55	240	244	161	79	215	134
0	65	82	42	101	203	86	59	178	77	223	154	227	243	75	210	127

Figure 8: Improved AES S-Box

Substitute byte operation and its access time is fixed and indestructible. S-Box using combinational logic contains small area and it provides high level efficiency and throughput. In the combinational based S-Box 2 stage pipeline is introduced for the S-Box implementation[10]. Area is taken by this design is 43 slices and maximum clock frequency of 72.155 MHZ.

Substitute byte transformation:
It is computed by taking the multiplicative inverse in GF(2^8) followed by Affine transformation of input byte.

Inverse-Substitute transformation:
It is the reverse process of Substitute byte transformation. Inverse-Substitute transformation is computed by applying inverse of Affine transformation followed by taking the multiplicative inverse in GF(2^8).



	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
F	0	5	93	fd	92	cf	88	b1	be	80	ab	ed	ff	15	96	ce
E	29	69	8f	59	21	c1	c0	37	9f	c3	92	98	60	6b	9b	af
D	49	6a	ef	c0	a0	c5	c3	3f	db	9c	61	88	b8	e0	c4	c4
C	2	52	b9	a9	88	74	5b	d4	ab	c4	14	b8	77	fb	89	19
B	8b	98	86	22	9b	48	68	40	3f	39	f7	97	bb	53	4	aa
A	f1	a0	c5	68	c9	52	3f	f0	7d	13	95	67	e6	71	9c	15
9	ff	40	2	7c	b1	ea	7c	15	ed	9b	27	3a	c0	49	8c	42
8	af	82	c5	7f	b5	84	5a	8a	51	59	73	67	9c	5a	3c	45
7	4f	b2	b1	c	c9	70	10	ef	7d	2e	67	80	9e	2f	b1	7a
6	31	99	1e	9d	57	7c	cc	1e	4c	19	38	cf	bc	34	42	13
5	94	bc	b6	b7	97	39	43	f1	de	65	4d	3e	4a	99	a9	55
4	1a	45	43	74	f0	50	c3	9c	2c	95	e3	e5	28	32	c5	8
3	22	4d	49	e	4b	9	6	33	ca	c4	e8	42	b6	8e	31	e5
2	d9	cf	d5	21	22	9e	bf	fa	37	c0	b8	4d	23	30	a4	6b
1	28	d8	8d	c4	94	3b	0	ed	bb	7b	5f	81	ef	c0	50	ea
0	89	d3	f8	1c	f2	5f	65	2f	80	f	90	90	a1	51	94	63

Figure 9: Coefficient of New(improved) AES S-box

There is relation between data and its correspondence Coefficient in improved AES S-Box.

$$S(x)=\sum_{x,y=0}^{15} c_{16*x+y}\, x^{16*x+y}$$

c) S-Box based on combinational logic
Affine Transformation:

$$\begin{pmatrix} b'0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} = \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Inverse-Affine Transformation:

$$\begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} = \begin{pmatrix} 0&0&1&0&0&1&0&1 \\ 1&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1 \\ 1&0&1&0&0&1&0&0 \\ 0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Note: S-Box can be generated by replacing the Affine Transformation matrix.

Here we can see that both SubByte and inverse SubByte contain a multiplicative inversion operation. Both involves the same multiplicative inversion operation in combined manner. Due to the similarity of SubByte and inverse SubByte than their operation followed by Affine transformation and its inverse, only implementation of SubByte is essential.

d) S-Box rotation

AES can be enhanced in security by applying the key dependent AES using S-box rotation[11]. In this algorithm, Key expansion along with S-box rotation makes the S-Box key dependent. This approach of key dependent S-box is much harder for attackers in doing any analysis.

Fig.10 depicts the New planned key dependent Encryption.

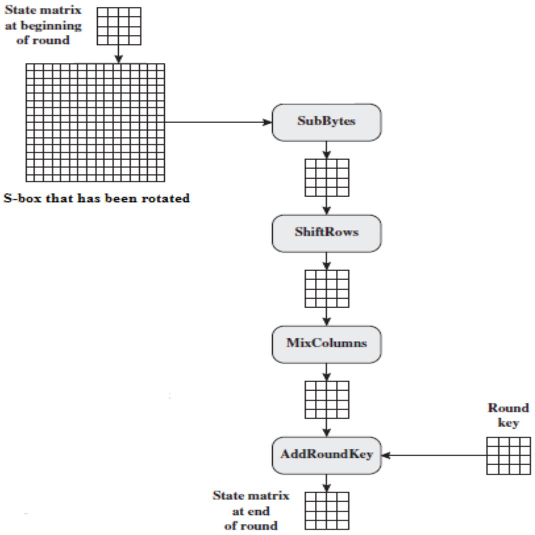


Figure 10: A new approach for key dependent Encryption

Fig.11 shows the Decryption for newly proposed key dependent S-Box.

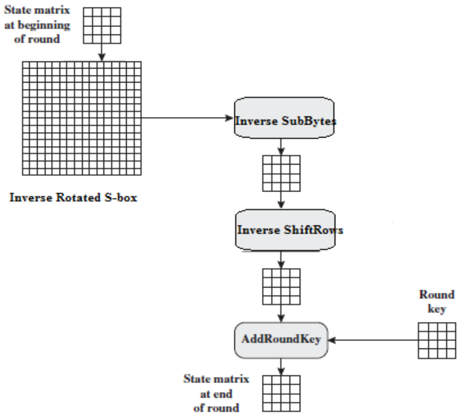


Figure 11: Newly proposed Key dependent Decryption

Round key is generated in each round using cipher key with the help of key scheduling algorithm.

e) S-Box construction

S-Box is the backbone of AES encryption standard. Rijndael's S-Box is created by the first irreducible polynomial out of thirty irreducible polynomial that can be constructed of the same degree over Gf(2^8). The isomorphic field to the core field can be generated by using differential irreducible polynomial of same degree over GF(2^8)[12].

Irreducible polynomial can be made by following Mobius function i.e

$\frac{1}{n}\sum_{d/n}\mu(d)\,p^{n/d}$,where μ is Mobius Function

$$\mu(n)=\begin{cases} 0 & \text{, if n=1 or more prime factor} \\ 1 & \text{, if n=1} \\ (-1)^k & \text{, n=product of k distinct primes} \end{cases}$$

from above equation a total of 30 irreducible polynomial of same degree(8) over GF(2^8) can be generated. Irreducible polynomial that is used in AES originally is ($x^8+x^4+x^3+x+1$).

Some polynomials are listed below

i.e

1) $x^8+x^7+x^5+x^4+1$
2) $x^8+x^6+x^5+x^4+1$
3) $x^8+x^4+x^3+x^2+1$
4) $x^8+x^7+x^6+x^4+x^3+x^2+1$
5) $x^8+x^6+x^5+x^4+x^2+x+1$
6) $x^8+x^5+x^3+x+1$
7) $x^8+x^7+x^5+x^3+1$
8) $x^8+x^7+x^6+x^5+x^4+x^3+1$
9) $x^8+x^5+x^4+x^3+x^2+x+1$
10) $x^8+x^7+x^6+x^4+x^2+x+1$ etc.

Affine matrix:

Affine marix can also be made that can replace the existing affine matrix and still this matrix works in efficient way. Affine matrix A∊$GL_8(2)$, it is a general linear group of degree 8 over the Galois Field,GF(2) and the group order is

$$\prod_{k=0}^{7}(2^8-2^k)\sim 5.3481\times 10^{18}$$

using above equation we can generate numerous Affine matrix. This is well tested on MAT Lab.

Some affine matrix are shown below

$$\begin{bmatrix} 0&0&1&0&1&1&0&1\\ 1&0&0&0&1&1&1&1\\ 1&1&0&1&1&0&0&0\\ 0&0&0&0&0&1&0&1\\ 1&0&1&0&1&1&0&0\\ 0&1&1&0&1&0&0&0\\ 1&1&1&1&1&0&1&0\\ 1&0&0&1&1&1&1&0 \end{bmatrix}, \begin{bmatrix} 1&1&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&1\\ 0&1&1&0&0&1&0&1\\ 1&0&1&1&0&0&1&0\\ 1&0&1&1&1&1&0&0\\ 1&1&1&1&1&1&1&1\\ 1&1&0&1&0&0&1&0\\ 1&0&0&1&1&0&0&1 \end{bmatrix}$$

$$\text{and} \begin{bmatrix} 1&0&1&0&1&0&1&1\\ 1&0&1&0&0&1&1&1\\ 1&1&0&0&1&0&1&1\\ 1&0&1&1&1&0&1&1\\ 0&0&0&1&0&1&1&0\\ 0&1&1&1&0&0&1&1\\ 0&0&1&0&1&0&0&1\\ 1&1&1&0&0&0&1&1 \end{bmatrix}.$$

4.2 Hardware based design for S-Box

a) Compact and High speed hardware design for Rijndael algorithm

Rijndael's algorithm can also be implemented on hardware with high efficiency and speed[13].

S-Box can be optimized by using composite field.

In Encryption and decryption all the arithmetic components are reused and data path is combined.

It can be obtained of enormously small size 5.4Kgates for 128-bit key Rijnadael's circuit by means of 0.11µm CMOS standard.

Cell library.

It occupies only 0.052mm^2 area for both encryption/decryption with 311 mbps throughput. It can be increased upto 2.6Gbps of size 21.3Kgates for high speed implementation.

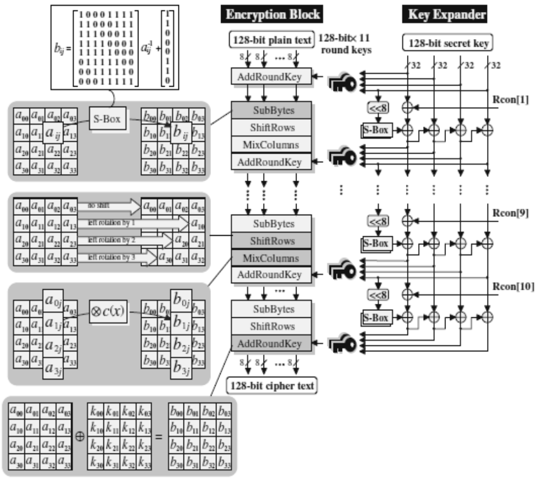


Figure 12: Rijndael's encryption process(source:[13])

Above figure depicts overall Encryption process of Rijndael's algorithm.

In hardware approach both Encryption/Decryption block uses 16-byte data register and ShiftRow operation or Inverse-ShiftRow processed by self.

In this approach Shiftrow(or Inverse-ShiftRow) and SubByte(or Inverse-SubByte) is different but there is no any effect on this algorithm, it works properly.

S-Box is used only once in Key Expansion and 4-times used in Both Encryption and Decryption block

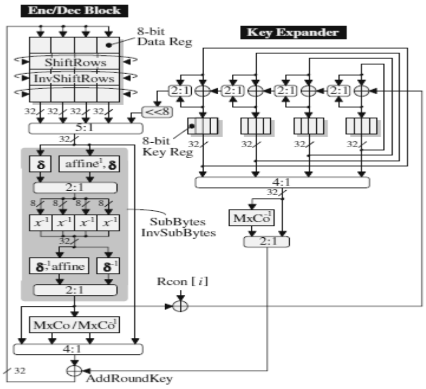


Figure 13: Architecture for Data path(source:[13])

During Key Expansion S-Box and ShiftRow (or Inverse-ShiftRow) operationsare executed simultaneously. Initial round requires 4-cyle because there is no S-Box transformation is needed and remaining round takes 5-cycles.

Total of (4+5*10)=54cycles needed.

b) Multiplexer-Look-Up-Table(MLUT)based S-Box

To protect the AES from side channel attacks(SDAs), MLUT can be used as a countermeasure against SDAs[18]. It requires 256-bytes to 1-byte multiplexer and memory uses 256 bytes[19]. Multiplexer based S-Box is 30 times more secure regarding SDAs than conventional

AES based.

Secret information can be leaked by correlating its transitional data with leaked physical parameter. Leaked physical elements can be power dissipation, electromagnetic radiation, and timestamp information.

SDA can be analyzed by three way:

- Simple Power Analysis(SPA)
- Differential Power Analysis(DPA)
- Correlation Power Analysis(CPA)

CPA is the powerful attack than other. Basically two countermeasure can be applied on CPA[18] i.e

- Hiding (hardware based)
- Masking (software based)

In hiding technique, the dependency between transitional data and physical elements is being reduced. In masking transitional data is being masked for securing against leaked elements.

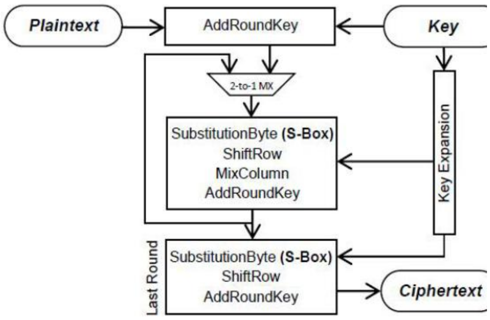


Figure 14: AES encryption

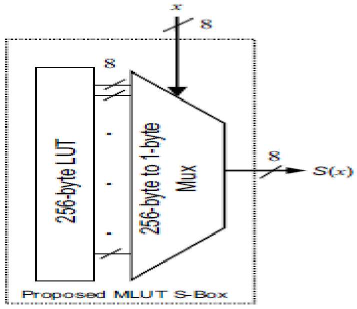


Figure 15: multiplexer based S-Box

c) S-Box over Composite Field Arithmatic (CFA)

As we have seen that S-Box is made over the GF(2^8). Decomposition of GF(2^8) into GF($((2^2)^2)^2$) provides another way of making S-Box.

Implementation of CFA on Field Programmable gate Array(FPGA) reduces the gates count that is used in hardware than the conventional Look Table based S-Box.

S-Box, based on CFA on hardware reduces the area by 50% and also diminishes the power consumption.

This architecture replaces the conventional S-Box based on look Up table to Composite Field Arithmetic design.

Using CFA, S-Box can be constructed in 16 different proposed ways[14].
GF(2^8) can be decomposed into GF($(2^4)^2$) and GF($((2^2)^2)^2$).
GF(2^8) to GF($((2^2)^2)^2$) ,a total of eight isomorphic mapping elements can be constructed.

S-Box can be shown as
$S=MS^{-1}+C$

Here multiplicative inversion over GF(2^8) is followed by Affine transformation matrix.
M= 8×8 binary matrix
C = 8-bit binary vector
CFA that contains irreducible polynomial

Can be represented as:
GF(2^2):x^2+x+1
GF($(2^2)^2$):$x^2+x+\phi$, $\phi=\{10\}_2$
GF($((2^2)^2)^2$):$x^2+x+\lambda$, $\lambda\{1100\}_2$

For $\phi=\{10\}_2$, $\lambda\{1100\}_2$ smallest amount of gate count in hardware and low power consumption can be seen.

d) Gray S-Box

Binary gray encoding technique can be a better option for increasing the complexity of algebraic expression[15]. It provides a easiest implementation in digital communication systems. Gray code is an encoding technique by a single bit difference for the consecutive value.

Binary gray code can calculated as

For a =[0 to n],input array in binary form,
b =[0 to n],output array in binary form,
where [0] is LSB.]
b[n] =a[n];
for k = (n-1) to 0
b[k] = a[k+1] ⊕ a[k].

In GF(2^8) binary number can be converted into linear form as

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{pmatrix} 1&1&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0 \\ 0&0&0&1&1&0&0&0 \\ 0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&1 \end{pmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix}$$

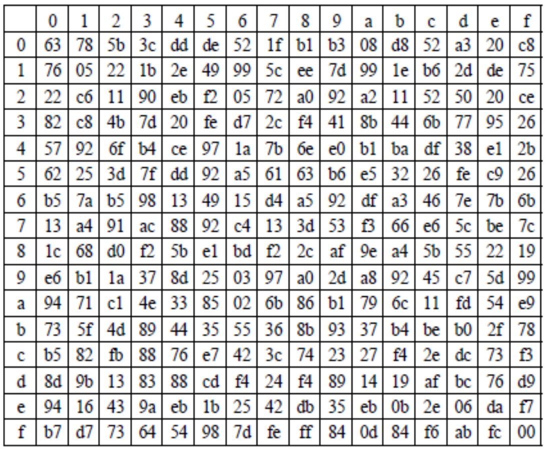
where (a_0 is the LSB bit), and a_k is the kth bit of byte a.
b_k = kth bit of byte b.
let a=x and b=y;

AES polynomial in GF(2^8)[x]/(m(x)) can be represented in linear form as
$L(x)$ = '8f'$x^{(7f)}$+'b5'$x^{(bf)}$+ '01'$x^{(df)}$+ 'f4'$x^{(ef)}$ + '25'$x^{(f7)}$ + 'f9'$x^{(7b)}$ + '09'$x^{(7d)}$ + '05'$x^{(7e)}$+ '63'

Above expression can be replace by gray code conversion as
$G(x)=(98)x^{(80)} + (e5)x^{(40)} +(4e)x^{(20)}+(3c)x^{(20)} +(13)x^{(08)}+(93)x^{(04)}+(9b)x^{(02)}+(15)x$

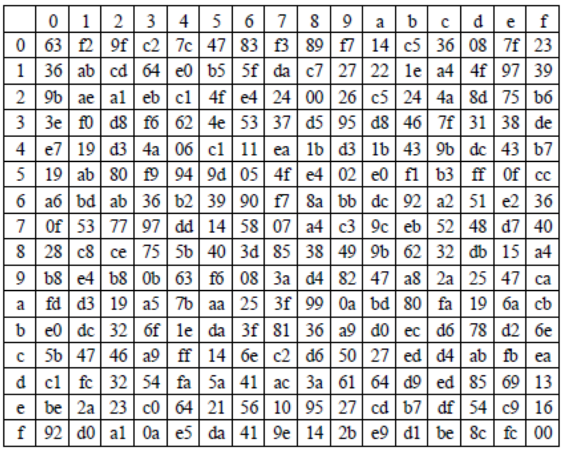
The modified S-Box or Gray S-Box can be given as amalgamation of original AES and Gray code conversion G(x).
Gray S-Box can be represented as function as follow
$G_s(x)=\sum_{0\le i,j<16} a_{ij}\, x^{16i+j}$
a_{ij} = hexadecimal value in ith row and jth column given in Fig.17
Here the highest degree of algebraic expression is 254 and total of 255 terms than original AES which has only 9-terms.

Inverse Gray S-Box function can be given as
$G_s^{-1}(x) = \sum_{0\le i,j<16} b_{ij}\, x^{16i+j}$
b_{ij} = hexadecimal value in ith row and jth column given in Fig.18



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	78	5b	3c	dd	de	52	1f	b1	b3	08	d8	52	a3	20	c8
1	76	05	22	1b	2e	49	99	5c	ee	7d	99	1e	b6	2d	de	75
2	22	c6	11	90	eb	f2	05	72	a0	92	a2	11	52	50	20	ce
3	82	c8	4b	7d	20	fe	d7	2c	f4	41	8b	44	6b	77	95	26
4	57	92	6f	b4	ce	97	1a	7b	6e	e0	b1	ba	df	38	e1	2b
5	62	25	3d	7f	dd	92	a5	61	63	b6	e5	32	26	fe	c9	26
6	b5	7a	b5	98	13	49	15	d4	a5	92	df	a3	46	7e	7b	6b
7	13	a4	91	ac	88	92	c4	13	3d	53	f3	66	e6	5c	be	7c
8	1c	68	d0	f2	5b	e1	bd	f2	2c	af	9e	a4	5b	55	22	19
9	e6	b1	1a	37	8d	25	03	97	a0	2d	a8	92	45	c7	5d	99
a	94	71	c1	4e	33	85	02	6b	86	b1	79	6c	11	fd	54	e9
b	73	5f	4d	89	44	35	55	36	8b	93	37	b4	be	b0	2f	78
c	b5	82	fb	88	76	e7	42	3c	74	23	27	f4	2e	dc	73	f3
d	8d	9b	13	83	88	cd	f4	24	f4	89	14	19	af	bc	76	d9
e	94	16	43	9a	eb	1b	25	42	db	35	eb	0b	2e	06	da	f7
f	b7	d7	73	64	54	98	7d	fe	ff	84	0d	84	f6	ab	fc	00

Figure 17: Gray S-Box



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	f2	9f	c2	7c	47	83	f3	89	f7	14	c5	36	08	7f	23
1	36	ab	cd	64	e0	b5	5f	da	c7	27	22	1e	a4	4f	97	39
2	9b	ae	a1	eb	c1	4f	e4	24	00	26	c5	24	4a	8d	75	b6
3	3e	f0	d8	f6	62	4e	53	37	d5	95	d8	46	7f	31	38	de
4	e7	19	d3	4a	06	c1	11	ea	1b	d3	1b	43	9b	dc	43	b7
5	19	ab	80	f9	94	9d	05	4f	e4	02	e0	f1	b3	ff	0f	cc
6	a6	bd	ab	36	b2	39	90	f7	8a	bb	dc	92	a2	51	e2	36
7	0f	53	77	97	dd	14	58	07	a4	c3	9c	eb	52	48	d7	40
8	28	c8	ce	75	5b	40	3d	85	38	49	9b	62	32	db	15	a4
9	b8	e4	b8	0b	63	f6	08	3a	d4	82	47	a8	2a	25	47	ca
a	fd	d3	19	a5	7b	aa	25	3f	99	0a	bd	80	fa	19	6a	cb
b	e0	dc	32	6f	1e	da	3f	81	36	a9	d0	ec	d6	78	d2	6e
c	5b	47	46	a9	ff	14	6e	c2	d6	50	27	ed	d4	ab	fb	ea
d	c1	fc	32	54	fa	5a	41	ac	3a	61	64	d9	ed	85	69	13
e	be	2a	23	c0	64	21	56	10	95	27	cd	b7	df	54	c9	16
f	92	d0	a1	0a	e5	da	41	9e	14	2b	e9	d1	be	8c	fc	00

Figure 18: Inverse S-Box

Gray S-Box has some properties:
a) Non-Linearity
b) Differential equality
c) Strict avalanche standard

Further it provides security from different algebraic attack and interpolation attack.

4.3 S-Box using Genetic Algorithm and Neural network

In AES cryptosystem, a greater complexity leads to a greater security to resisting any cryptanalytic attacks. But increasing complexity leads to increase in processing time and can get timing attacks. In this situation Genetic Algorithm (GN) and Neural Network (NN) can be better option for reducing the processing time and provides security from timing attacks[16].

As we know that AES is based on Substitution-permutation based encryption system.

In GA and NN system, as a Substitution-permutation Network (SPN) it provides the non-linearity to the system and resistance to the cryptanalytic attacks.

Genetic Algorithm:
Genetic Algorithm is basically used for substitution and some transposition [54]. It creates cipher for block and generates keys[55] and also is used for both text and images. It uses basically three operators

a) Selection: selection of chromosomes that is generated by Pseudo Random Number Generator (PRNG) on initial population.
b) Crossover: to reproduce a new set of values by combining one generated chromosome with another.
c) Mutation: it is used for presenting difference of chromosomes that is newly generated.

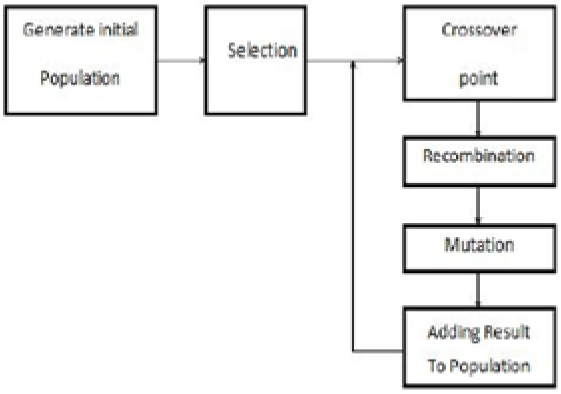


Figure 19: Genetic Algorithm working manner

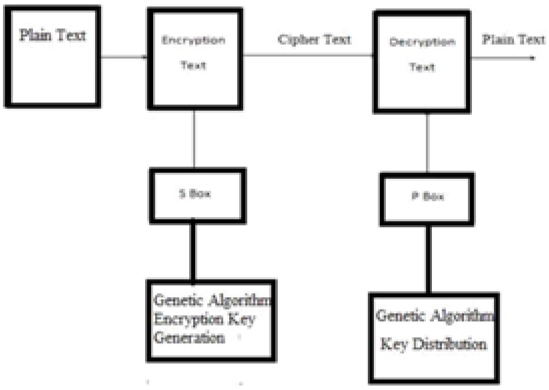

Plain Text
Encryption Text
Cipher Text
Decryption Text
Plain Text
S Box
P Box
Genetic Algorithm Encryption Key Generation
Genetic Algorithm Key Distribution

Figure 20: Functioning of AES with GA

Neural Network:
In Genetic Algorithm, when the computational time get increased in finding the fitness function and encoding, Neural network provides solution for it. Neural network containing knowledge elements is correspond to natural nervous system. It is also used for substitution in the S-boxes and works on 128-bit data block. Neural Network can be seen from the following figure.

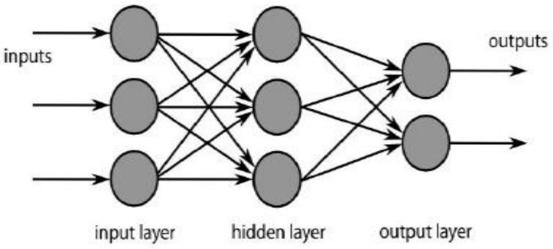


Figure 21: Architecture of Neural Network

DOI: 10.21275/ART20164149

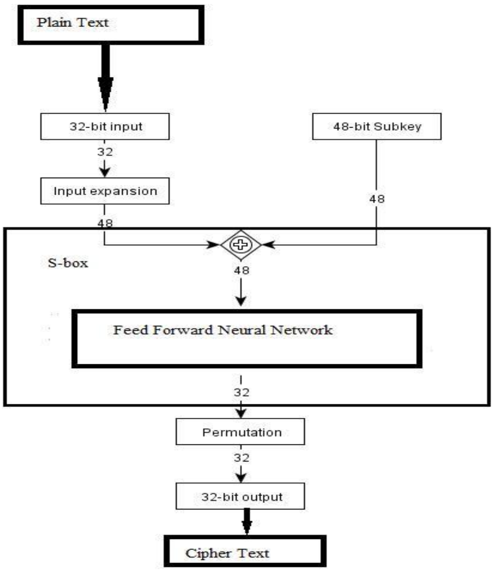


Figure 22: Functioning AES with NN.

4.4 Cellular Automata based S-Box

Cellular Automata’s concept was come in 1940 that was given by von Neumann and Ulam to learn the idea of biological process i.e Self-reproduction[17].

It has the property of parallel processing so implementation speed is high compared with classical AES S-Box.
In CA’s system, space and time are distinct. it’s basically used for creating a pseudo random number.
It is a array of cells and at particular time state of a cell is determined by the current states of neighborhood cells.

In CA array of cell is n dimensional and the identical rule that is enclosed in every cell is a finite state machine and it can be precise in rule table (or transition function)

Rule table contains an entry to every neighborhood relationship of states.

In 1-D CA, a cell contains r local neighbor at either side and to itself also where r is the radius so it make total of (2r+1) neighbors.

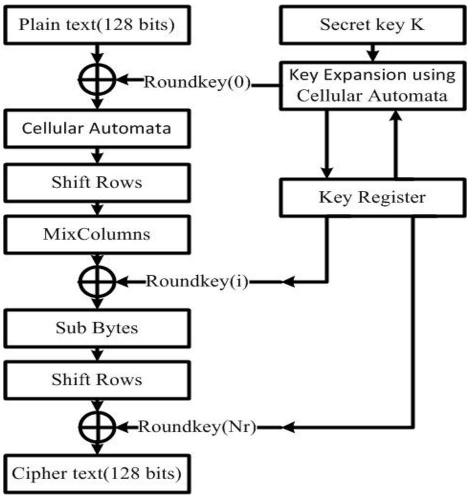


Figure 23: AES process through CA

Cellular Automata has some specific properties:

a) Strict Avalanche Criteria
b) Input/output bit to bit Entropy
c) Non-linearity
d) Correlation Immunity Bias

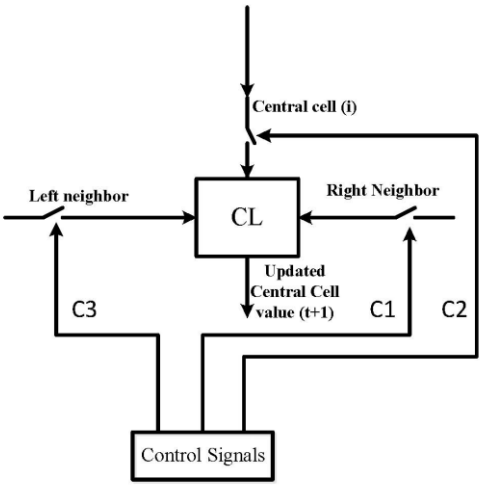


Figure 24: Cellular automata source

5. FPGA Implementation of AES

As we know that Rijndael’s AES encryption/decryption algorithm is selected after the proper implementation on software as well as hardware. For hardware implementation, Field Programmable Gate Array (FPGA) become the more suitable option. It is reprogrammable device that provides agility, physical security and high performance than any software implementation [20].

FPGA specification is given in [21].

FPGA mainly focus on three area:

a) Algorithm integrity
- It ensures the originality of data at any instant of time[22].

b) High throughput
- It ensures the encryption as fast as
- possible [23,24].

c) Low power consumption
- It ensures the lower consumption of
- Power [25,26].

By means of power analysis, normal AES is broken in 1999[56]. To protect the data from differential power analysis (DPA) attacks, a high throughput masked AES is projected[57]. In AES block cipher Power and Sizes can also be reduced by using compact key expansion mechanism. It can be design with mainly three methodologies:

- Implementation of optimized number of AES S-Box [27]
- Dipping down the number of pipeline register
- Sharing input bus

It gives three optimized architecture. Total of 1818 logic element, 122.04mW power dissipation and throughput of 198.77Mbps is obtained in best architecture [28].

6. Design Technique of AES regarding Complexity

6.1 Area efficient and Low- Cost Design of AES

[29]An Area-Efficient design of Advanced Encryption Standard (AES) can be made by eliminating the common sub-expression. As we know that in decryption, four Inverse transformation is performed i.e Inv-SubBytes (ISB), Inv-ShiftRows (ISR), Inv-MixColumns (IMC) and AddRoundKey.

Since decryption process by swapping of ISR and ISB is not affected and IMC can also be processed before ARK as long as the round key in the ARK undergo IMC function.

After performing this swapping, decryption process become equivalent to the encryption process regarding sequence of transformation except for inverse transformation and modified key expansion.

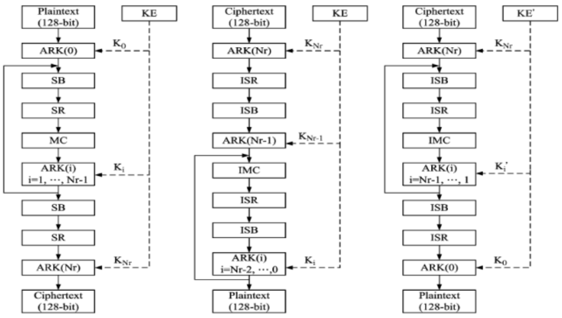


Figure 25: Equivalence of encryption and modified Decryption regarding sequence of transformation.

To design a low complexity AES, a method of using extended instruction technique can be used in which the instructions is reduced from 688 to 340 [30].

6.2 High Regular and Scalable AES Architecture

AES hardware based design can be made as a high regular and scalable. it can present a high throughput of 241Mbits/s on a 0.6µm area of CMOS process[36].

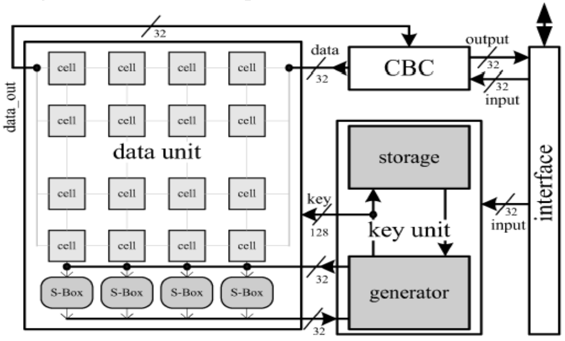


Figure 26: Hardware Module of the AES

This Hardware Module contains four components i.e

Interface
It handles all the communication of this module with its surroundings and it's communication is based on 32-bit data words.

Data Unit
It is used to perform encryption/decryption round using the round key.
Data input is independent of key sizes.

Key Input
This is used for storing the cipher key as well as computation of round key.

Cipher Block Chaining (CBC)
Since Data Unit, Key Unit and Interface perform AES Encryption/Decryption in Electronic Code Book (ECB) mode.
To provide resistance against attacks due to the reordering of blocks, AES is performed in CBS mode in which output of AES Encryption is X-OR with the next 128-bit data input.

6.3 Pipelined Architecture of AES

Two architecture can be used for Encryption/Decryption.
First Architecture can be based on feedback strategy and can reach to throughput value of 259Mbit/s.
Second Architecture is based on Pipeline technique and can reach to throughput value of 3.65Gbit/s.
Second Architecture has two characteristics:
a) Pipeline Technique
b) Key Storage using RAM.

Pipeline Technique can not possible for other cryptographic application but Rijndael's AES can be implemented using pipeline technique. paper[60] can be referenced for more details.

Hardware implementation provides high throughput than software implementation [59].

AES can be implemented on parallel, pipeline and in sequential manner. The Field Programming Gate Array (FPGA) is used for performing parallel operations. Throughput can be increased to three times after using these three implementation of AES [58].

6.4 Dual Core Architecture

A Dual Core architecture is described in paper [31]. In compact Dual-Core Architecture, encryption and decryption is done simultaneously. it is used in real-time dual-duplex wireless communication. It uses 32-bit data path with four clock cycle to implement 128-bit data for one round.

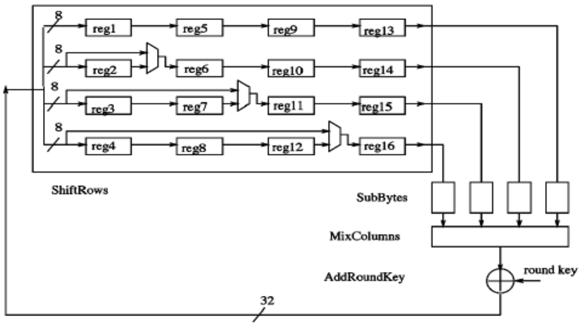


Figure 27: Data path for encryption

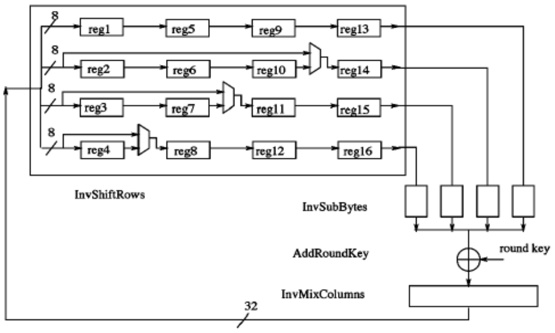


Figure 28: Data path for decryption

6.5 Implementation on Grain on Sand

AES can be implemented on hardware which requires very less resources. It is implemented on 0.35µm CMOS process which requires only 0.25mm^2 area and needed 3400 gates that is corresponding to a size of a grain of sand. This is done on a semiconductor (i.e silicon)
and called silicon implementation. It requires very less amount of hardware resources and the power efficiency is very high. Detailed description is given in paper [32].

7. Attacks and Countermeasure (if possible)

a) In [33], a parity based technique is given for error detection. Parity based concurrent checking method leads to hardware overhead because it adds one additional bit per bytes in 128-bit data. This result in 16 extra bits which modifies (8-bit × 8-bit) S-Box into (9-bit × 9-bit) S-Box.

To overcome the overhead due to the parity based concurrent checking, paper[34] describes the low-cost concurrent checking method.

This method checks for Substitution-Permutation-Network (SPN) and the input parity bit is modified according to the execution step of SPN into the output parity bit and then it is compared with output parity of every round.

b) Some attack is based on fault injection into the cryptographic devices which aim is to incorporating an error to recover the secret keys.
In taking countermeasure, an extension [35] is done on an existing AES architecture which is given by Mangard et al.[36].

A fault injection attack can be done on smart card implementing the AES.

In the countermeasure, paper [37] shows the technique to protect from this attack.

This attack can be a side channel attack known as Differential Fault Analysis and it uses the nonlinear robust error detecting codes to handle this attacks.

A high efficient fault can be induced in the AES round. It changes the mapping relationship of S-Boxes. Using this two fault models can be presented. First model requires only 16 faulty ciphertexts to obtain the secret key of 128bit and second model requires two rounds attack by Differential Fault Analysis to achieve 4-byte round key in 9th round[38].

A Differential Fault Analysis attack can be done on during key expansion [39]. It can be seen in AES-128 with two faults, AES-192 with six and AES-256 with four faults using a new differential attack [40].

B.Baharak [40] proposed a impossible differential attack, which is done on AES-128 upto seven round. It requires $2^{115.5}$ plaintext, 2^{109} bytes memory and 2^{119} seven round encryption. This impossible differential attack take advantage of differences that is impossible at several intermediate state of algorithm. [41]A Fault Round Modification analysis is developed, which can analyze the modification in AES round. This is efficient than Differential Fault Analysis.

c) [42]There can be a collision attack on the 7-round of Rijndael's algorithm
It takes advantage of some collision that exist between the partial function bring by the cipher. Paper [43] describes the collision timing attacks. Since timing attributes on combination circuit depends on the I/O variation of function. This characteristic can be gain by fault sensitivity analysis.

Due to the access time uncertainty and resource sharing in cache memory, there has been a timing attack which can leak secret information.

To find full AES keys, first round takes 300 samples and second round takes 80 samples [44].

[45]To prevent AES algorithm from Differential Power Analysis (DPA) , masking of logical gates is very effective. In spite of this masking, logic circuit used in implementation of AES algorithm, leak the side channel secret information which can be taken advantages in Differential power attacks.

A clock wise collision attack can also be induced in masked AES. It is also known as Fault Rate Analysis. In this attack, clock glitch is inject into the masked

S-Box to recover the secret key[46].

A Low complexity attack can be done on AES upto four round using three known plaintext and it can also increased to six round[47].

8. Application of AES

8.1 Implementation in Smart card

In spite of three times faster than DES, AES is difficult to implement in smart card.

NexCard which is a Chip operating System, design by Microsoft, become the more suitable platform for implementing the AES as a efficient memory usage.

Cipher System on Demand (CSOD) method, utilizes the multi-application capacity of NexCardv2.0 to execute the same AES algorithm on Card [48]. In paper [49], also given integrated design of AES that gives the suitable hardware architecture to implement AES in smart card.

8.2 AES in Radio Frequency Identification(RFID)

AES can be used as a strong authentication in RFID. In this approach 128-bit data block uses 1000 clock cycles and power consumption is less than 9*μA on 0.35μm CMOS[50]*.

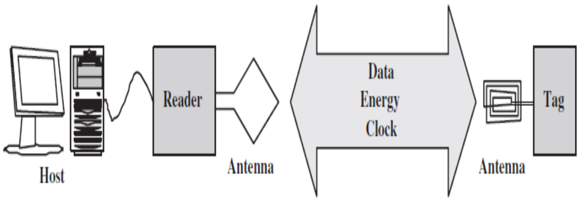


Figure 29: Structure of RFID

8.3 Image Encryption

In today's communication system, it is important to secure text data as well as multimedia data like image, video, audio etc.
A modified AES can be made to fulfill this requirement.

In this version of AES, modification is focused on ShiftRows transformations.

The 1st and 4th row is unchanged when the value of 1st row and 1st column is even and the bytes of 2 and 3 rows is shifted right by different number.

Contrary, if first row and first column is odd,1st and 3rd rows are unchanged and the byte of 2nd and 4th shifted left by different number[51].

AES can also be used to secure the biometric image data[52].

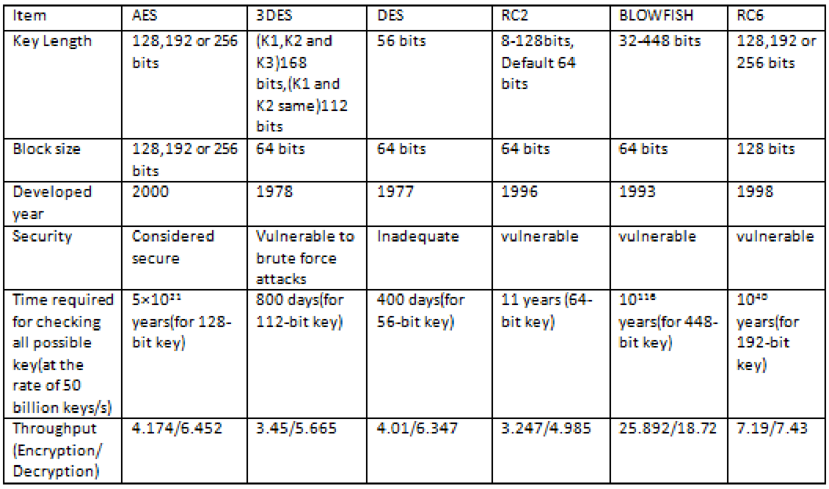
By using Singular Value Decomposition (SVD) and Discrete Wavelet Transform (DWT) as a SVD-DWT watermarking technique, AES can be use to transfer medical image securely[53].

8.4 AES implementation for SANs

To secure the large-scale Storage Area Network, AES can be used as trusted Encryption/Decryption Algorithm.It can be designed based on FPGA and takes advantage of all the resources of recent FPGA i.e Block RAM(BRAM) and Digital Signal Processing(DSP) slices[61].

9. Performance of AES

Now, it can said that AES works properly on software as well as hardware architecture. In hardware, it provides high performance from 8-bit processor to high bit processor. It provides throughput of 11Mb/s for a 200Mhz processor with a 18 clock cycles on a Pentium and a throughput of 60Mb/s on 1.7Ghz Pentium M processor[62].



Item	AES	3DES	DES	RC2	BLOWFISH	RC6
Key Length	128,192 or 256 bits	(K1,K2 and K3)168 bits,(K1 and K2 same)112 bits	56 bits	8-128bits, Default 64 bits	32-448 bits	128,192 or 256 bits
Block size	128,192 or 256 bits	64 bits	64 bits	64 bits	64 bits	128 bits
Developed year	2000	1978	1977	1996	1993	1998
Security	Considered secure	Vulnerable to brute force attacks	Inadequate	vulnerable	vulnerable	vulnerable
Time required for checking all possible key(at the rate of 50 billion keys/s)	5×10^{21} years(for 128-bit key)	800 days(for 112-bit key)	400 days(for 56-bit key)	11 years (64-bit key)	10^{116} years(for 448-bit key)	10^{40} years(for 192-bit key)
Throughput (Encryption/ Decryption)	4.174/6.452	3.45/5.665	4.01/6.347	3.247/4.985	25.892/18.72	7.19/7.43

Figure 30: Comparison of AES with other cryptographic algorithm (source:[63])

10. Conclusion and Future Work

Consequently it can be said that Rijndael's AES Encryption/Decryption algorithm is high efficient and secure in terms of speed, time, throughput to any other symmetric cryptographic algorithm. It is very strong against different type of attacks like differential attacks, interpolation and square attack. This paper's motive is to present all the valuable work that has been done on AES. Since increasing complexity leads to more security.

Hence, in future work our motive will be to test different S-Box in each round of Encryption / Decryption in which S-Boxes can be made by different polynomial of same degree i.e Galois Field (2^8).

References

[1] Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael." (1999).

[2] National Institute of Standards and Technology, Advanced encryption standard (AES). (FIPS 197), 2001. http://csrc.nist.gov/.publications

[3] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard_process

[4] Nechvatal, James, et al. *Report on the development of the Advanced Encryption Standard (AES)*.

[5] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson, Tadayoshi Kohno, Mike Stay (May 2000). "The Twofish Team's Final Comments on AES Selection" (http:/ / www. schneier. com/ paper-twofish-final. pdf).

[6] Rijmen, Vincent, and Joan Daemen. "Advanced encryption standard." *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology* (2001): 19-22.

[7] Selent, Douglas. "Advanced encryption standard." *Rivier Academic Journal*6.2 (2010): 1-14.

[8] Ferguson, Niels, Richard Schroeppel, and Doug Whiting. "A simple algebraic representation of Rijndael." *International Workshop on Selected Areas in Cryptography*. Springer Berlin Heidelberg, 2001.

[9] Jinomeiq, Liu, Wei Baoduui, and Wang Xinmei. "One AES S-box to increase complexity and its cryptanalysis." *Journal of Systems Engineering and Electronics* 18.2 (2007): 427-433.

[10] Mui, Edwin NC, R. Custom, and D. Engineer. "Practical implementation of Rijndael S-box using Combinational logic." *Custom R&D Engineer Texco Enterprise Pvt. Ltd* (2007).

[11] Juremi, Julia, et al. "Enhancing advanced encryption standard S-box generation based on round key." *International Journal of Cyber-Security and Digital Forensics (IJCSDF)* 1.3 (2012): 183-188.

[12] Sinha, Shristi Deva, and Chaman Prakash Arya. "Algebraic Construction and Cryptographic Properties of Rijndael Substitution Box." *Defence Science Journal* 62.1 (2012): 32-37.

[13] Satoh, Akashi, et al. "A compact Rijndael hardware architecture with S-box optimization." *International Conference on the Theory and Application of Cryptology and Information Security*. Springer Berlin Heidelberg, 2001.

[14] Gangadari, Bhoopal Rao, and Shaik Rafi Ahamed. "FPGA implementation of compact S-Box for AES algorithm using composite field arithmetic." *2015 Annual IEEE India Conference (INDICON)*. IEEE, 2015.

[15] Tran, Minh Triet, Doan Khanh Bui, and Anh Duc Duong. "Gray S-box for advanced encryption standard." *Computational Intelligence and Security, 2008. CIS'08. International Conference on*. Vol. 1. IEEE, 2008.

[16] Kalaiselvi, K., and Anand Kumar. "Enhanced AES cryptosystem by using genetic algorithm and neural network in S-box." *Current Trends in Advanced Computing (ICCTAC), IEEE International Conference on*. IEEE, 2016.

[17] Gangadari, Bhoopal Rao, et al. "Design of cryptographically secure AES S-Box using cellular automata." *Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on*. IEEE, 2015.

[18] S. Mangard, E. Oswald, and T. Popp, *PoweAnalysis Attacks*. US:Springer 2007.

[19] Pammu, Ali Akbar, et al. "High Secured Low Power Multiplexer-LUT Based AES S-Box Implementation." *2016 International Conference on Information Systems Engineering (ICISE)*. IEEE, 2016.

[20] Elbirt, Adam J., et al. "An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 9.4 (2001): 545-557.

[21] Van Dyken, Jason, and José G. Delgado-Frias. "FPGA schemes for minimizing the power-throughput trade-off in executing the Advanced Encryption Standard algorithm." *Journal of Systems Architecture* 56.2 (2010): 116-123.

[22] Standaert, François-Xavier, Sıddıka Berna Örs, and Bart Preneel. "Power Analysis of an FPGA." *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2004.

[23] Rodrıguez-Henrıquez, F., N. A. Saqib, and A. Dıaz-Pérez. "4.2 Gbit/s single-chip FPGA implementation of AES algorithm." *Electr. Lett* 39.15 (2003): 1115-1116.

[24] Hodjat, Alireza, and Ingrid Verbauwhede. "A 21.54 Gbits/s fully pipelined AES processor on FPGA." *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on*. IEEE, 2004.

[25] Chodowiec, Paweł, and Kris Gaj. "Very compact FPGA implementation of the AES algorithm." *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2003.

[26] Kaps, Jens-Peter, Gunnar Gaubatz, and Berk Sunar. "Cryptography on a Speck of Dust." *IEEE Computer* 40.2 (2007): 38-44.

[27] M. M. Wong, "VLSI Implementation and Its Optimisation for Digital Cryptosystems," Ph.D. dissertaton, School of Engineering, Computing and Science, Swinburne University of Technology Sarawak Campus,Sarawak, Malaysia, 2012

[28] Tay, J. J., Ming Ming Wong, and I. Hijazin. "Compact and low power aes block cipher using lightweight key expansion mechanism and optimal number of s-boxes." *Intelligent Signal Processing and Communication Systems (ISPACS), 2014 International Symposium on*. IEEE, 2014.

[29] Hsiao, Shen-Fu, Ming-Chih Chen, and Chia-Shin Tu. "Memory-free low-cost designs of advanced encryption standard using common subexpression elimination for subfunctions in transformations." *IEEE Transactions on Circuits and Systems I: Regular Papers* 53.3 (2006): 615-626.

[30] Nadehara, Kouhei, Masao Ikekawa, and Ichiro Kuroda. "Extended instructions for the AES cryptography and their efficient implementation." *Signal Processing Systems, 2004. SIPS 2004. IEEE Workshop on*. IEEE, 2004.

[31] Li, Hua, and Jianzhou Li. "A new compact dual-core architecture for AES encryption and decryption." *Canadian Journal of Electrical and Computer Engineering* 33.3/4 (2008): 209-213.

[32] Feldhofer, Martin, Johannes Wolkerstorfer, and Vincent Rijmen. "AES implementation on a grain of sand." *IEE Proceedings-Information Security*152.1 (2005): 13-20.

[33] Bertoni, Guido, et al. "Error analysis and detection procedures for a hardware implementation of the advanced encryption standard." *IEEE Transactions on Computers* 52.4 (2003): 492-505.

[34] Wu, Kaijie, et al. "Low cost concurrent error detection for the advanced encryption standard." *Test Conference, 2004. Proceedings. ITC 2004. International*. IEEE, 2004.

[35] Breveglieri, Luca, Israel Koren, and Paolo Maistri. "Incorporating error detection and online reconfiguration into a regular architecture for the advanced encryption standard." *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*. IEEE, 2005.

[36] Mangard, Stefan, Manfred Aigner, and Sandra Dominikus. "A highly regular and scalable AES hardware architecture." *IEEE Transactions on Computers*52.4 (2003): 483-491.

[37] Karpovsky, Mark, Konrad J. Kulikowski, and Alexander Taubin. "Robust protection against fault-injection attacks on smart cards implementing the advanced encryption standard." *Dependable Systems and Networks, 2004 International Conference on*. IEEE, 2004.

[38] Liao, Nan, et al. "A high-efficient fault attack on AES S-box." *Information Science and Technology (ICIST), 2016 Sixth International Conference on*. IEEE, 2016.

[39] Floissac, Noémie, and Yann L'Hyver. "From AES-128 to AES-192 and AES-256, how to adapt differential fault analysis attacks on key expansion." *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2011 Workshop on*. IEEE, 2011.

[40] Bahrak, Behnam, and Mohammad Reza Aref. "Impossible differential attack on seven-round AES-128." *IET Information Security* 2.2 (2008): 28-32.

[41] Dutertre, Jean-Max, et al. "Fault round modification analysis of the advanced encryption standard." *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*. IEEE, 2012.

[42] Gilbert, Henri, and Marine Minier. "A collisions attack on the 7-rounds Rijndael." *AES Candidate Conference, Citeseer*. 2000.

[43] Moradi, Amir, Oliver Mischke, and Christof Paar. "One attack to rule them all: collision timing attack versus 42 AES ASIC cores." *IEEE Transactions on Computers* 62.9 (2013): 1786-1798.

[44] Xinjie, Zhao, et al. "Robust first two rounds access driven cache timing attack on AES." *Computer Science and Software Engineering, 2008 International Conference on*. Vol. 3. IEEE, 2008.

[45] Alam, Monjur, et al. "Effect of glitches against masked AES S-box implementation and countermeasure." *IET Information Security* 3.1 (2009): 34-44.

[46] Wang, An, et al. "Fault rate analysis: breaking masked AES hardware implementations efficiently." *IEEE Transactions on Circuits and Systems II: Express Briefs* 60.8 (2013): 517-521.

[47] Bouillaguet, Charles, et al. "Low-data complexity attacks on AES." *IEEE Transactions on Information Theory* 58.11 (2012): 7002-7017.

[48] Lu, Chi-Feng, et al. "Fast implementation of AES cryptographic algorithms in smart cards." *Security Technology, 2003. Proceedings. IEEE 37th Annual 2003 International Carnahan Conference on*. IEEE, 2003.

[49] Lu, Chih-Chung, and Shau-Yin Tseng. "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter." *Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on*. IEEE, 2002.

[50] Feldhofer, Martin, Sandra Dominikus, and Johannes Wolkerstorfer. "Strong authentication for RFID systems using the AES algorithm." *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2004.

[51] Kamali, Seyed Hossein, et al. "A new modified version of advanced encryption standard based algorithm for image encryption." *Electronics and Information Engineering (ICEIE), 2010 International Conference On*. Vol. 1. IEEE, 2010.

[52] Kester, Quist-Aphetsi, et al. "Feature Based Encryption Technique For Securing Forensic Biometric Image Data Using AES and Visual Cryptography." *Artificial Intelligence, Modelling and Simulation (AIMS), 2014 2nd International Conference on*. IEEE, 2014.

[53] Ajili, Sondes, Mohamed Ali Hajjaji, and Abdellatif Mtibaa. "Hybrid SVD-DWT watermarking technique using AES algorithm for medical image safe transfer." *Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2015 16th International Conference on*. IEEE, 2015

[54] W.Stallings“Cryptography and Network Security: Principles and Practice”, Prentice Hall, 3rd Edition, 2007.

[55] Tragha, A., F. Omary, and A. Mouloudi. "ICIGA: Improved cryptography inspired by genetic algorithms." *2006 International Conference on Hybrid Information Technology*. 2006.

[56] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in Proc. CRYPTO, 1999, vol. LNCS 1666, pp. 388 397

[57] Regazzoni, Francesco, Yi Wang, and François-Xavier Standaert. "FPGA implementations of the AES masked against power analysis attacks." *Proceedings of COSADE 2011, International Workshop on Side-Channel Analysis and Secure Design*. 2011.

[58] Deshpande, Pournima U., and Smita A. Bhosale. "AES encryption engines of many core processor arrays on FPGA by using parallel, pipeline and sequential technique." *Energy Systems and Applications, 2015 International Conference on*. IEEE, 2015.

[59] Liu, Bin, and Bevan M. Baas. "Parallel AES encryption engines for many-core processor arrays." *IEEE Transactions on Computers* 62.3 (2013): 536-547.

[60] Hodjat, Alireza, and Ingrid Verbauwhede. "A 21.54 Gbits/s fully pipelined AES processor on FPGA." *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on*. IEEE, 2004.

[61] Wang, Yi, and Yajun Ha. "High throughput and resource efficient AES encryption/decryption for SANs." *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*. IEEE, 2016.

[62] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard#Performance

[63] Mathur, Milind, and Ayush Kesarwani. "Comparison between Des, 3des, Rc2, Rc6, Blowfish And Aes." *Proceedings of National Conference on New Horizons in IT-NCNHIT*. Vol. 3. 2013.