Computer Science and Information Technology 5(5): 154-163, 2017 http://www.hrpub.org
DOI: 10.13189/csit.2017.050502

Analysis of Development of Dynamic S-Box Generation

Amandeep Singh[1,2,*], **Praveen Agarwal**[3], **Mehar Chand**[4]

[1]Department of Computer Science, Baba Farid College, Bathinda-151001, India

[2]Department of Computer Science, Singhania University, Pacheri Bari, Jhunjhunu, India

[3]Department of Mathematics, Anand International College of Engineering, Jaipur303012, India

[4]Department of Applied Sciences, Guru Kashi University, Bathinda-1513002, India

Abstract Advanced Encryption Standard is a symmetric block cipher which is widely used in encrypting data by different organizations to make secure their data from being hacked. The only nonlinear part of Advanced Encryption Standard (AES) is S-Box (Substitution Box), which provides confusion in the algorithm. But the main limitation of the S-Box in AES is that it is a static one throughout the algorithm, which is the main center of attraction for the cryptanalyst to analysis the weakness for certain attacks. Since 2000 onwards a number of algebraic attacks on AES have been carried out, which challenged the security of AES. But at the same time till date a number of researches have being carried out for making AES more secure by using dynamic S-Boxes to provide more confusion to the cryptanalyst. In present paper we tried to address dynamic S-Box techniques and provide their analysis on the basis of S-Box properties, which are essential for secure S-Box construction like Non-linearity, XOR profile, Strict Avalanche criterion (SAC) and Bit independence criteria (BIC). Also these techniques are compared with the original AES results.

Keywords Advanced Encryption Standard, S-Box, Dynamic S-Box, Non linearity, SAC, BIC, XOR Profile

1 Introduction

Cryptography is a method or technique of secure communication in the presence of an adversary. In modern age of computers, cryptography is a technique to scramble plain text or ordinary text into ciphertext (by using cryptographic algorithms called encryption process) and converting back into plain text on receiver side (called decryption). The central objective of modern cryptography is to attain data confidentiality, data integrity, authentication and non repudiation.

Cryptography is broadly divided into two major categories. One is symmetric key cryptography and another is asymmetric key cryptography.

Symmetric key encryption is a form of cryptosystem in which encryption and decryption are performed using the same key [1] . It is also known as conventional encryption. Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the ciphertext. Most widely used symmetric ciphers are DES (Data Encryption Standard) and AES (Advanced Encryption Standard).

Asymmetric key encryption is a form of cryptosystem in which encryption and decryption are performed using the different keys one is public key and another is private key [2]. It is also known as public-key encryption. Asymmetric key encryption transforms plaintext into ciphertext using a one of two keys and an encryption algorithm. Using the paired key and a decryption algorithm, the plaintext is recovered from the ciphertext. Asymmetric encryption can be used for confidentiality, authentication or both. The most widely used public-key cryptosystem are RSA, Diffie-Hellman key exchange, ElGamal Cryptosystem, Elliptic Curve cryptography.

2 Advanced Encryption Standard

AES is designed on the principle of combination of both substitution and permutation. AES is a variant of Rijndael which uses fixed input block size of 128 bits, which means data is divided into fixed 128 bit blocks and represented in matrix form, called state matrix and a key size of 128, 192, or 256 bits is used depending upon the variant

of AES used. AES operates on a 4×4 column-major order matrix of bytes. Most AES calculations are done in a special finite field GF(2^8). The key size used for an AES cipher specifies the number of repetitions of transformation rounds (Nr 10, 12, 14) that convert the input, called the plaintext, into the final output, called the ciphertext. The number of cycles of repetition is as follows:

- 10 cycles of repetition for 128-bit keys. [3]
- 12 cycles of repetition for 192-bit keys. [4]
- 14 cycles of repetition for 256-bit keys. [5]

For example consider the follwing input and key:
Input:- 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
Key:- 00 e9 c9 f2 a5 09 d4 e8 a8 bb b7 60 a0 2a ab 08
The number of column in input is used are denoted by N_b and in key is denoted by N_k.

$$Input = \begin{bmatrix} 32 & 88 & 31 & e0 \\ 43 & 5a & 31 & 37 \\ f6 & 30 & 98 & 07 \\ a8 & 8d & a2 & 34 \end{bmatrix}, Key = \begin{bmatrix} 00 & a5 & a8 & a0 \\ e9 & 09 & bb & 2a \\ c9 & d4 & b7 & ab \\ f2 & e8 & 60 & 08 \end{bmatrix}$$

Each round consists of four different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

2.1 Encryption Process

Encryption process follows following steps as shown in fig. no. (2)

- **Add Round Key** The Add Round Key operation is an XOR operation between the State and the Round Key. The State s and round key 'w' is of the same size. By doing XOR operation element by element of both (s and 'w') matrix next state matrix is obtained.

$$s^{'}(i,j) := s(i,j) \oplus w(i,j) \tag{1}$$

$$s = \begin{bmatrix} 32 & 88 & 31 & e0 \\ 43 & 5a & 31 & 37 \\ f6 & 30 & 98 & 07 \\ a8 & 8d & a2 & 34 \end{bmatrix}, w = \begin{bmatrix} 00 & a5 & a8 & a0 \\ e9 & 09 & bb & 2a \\ c9 & d4 & b7 & ab \\ f2 & e8 & 60 & 08 \end{bmatrix}$$

The new state will be s'

$$s' = \begin{bmatrix} 32 & 2d & 99 & 40 \\ aa & 53 & 8a & 1d \\ 3f & e4 & 2f & ac \\ 5a & 65 & c2 & 3c \end{bmatrix}$$

- **The Byte Substitution Transformation:** In AES, S-Box is generated by using GF(2^8) (Galois Field) and irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. AES S-Box is a matrix of (16 x 16 = 256) elements in which rows and columns are having values ranging from 0 to 15 (0 to f in hexadecimal). Each byte of S-Box is mapped to its multiplicative inverse in GF(2^8), where 00 is mapped into itself. Then, an affine transformation (over GF(2)) is computed. An affine cipher is a cipher of the following form: [6]. S-Box of AES is generated by equation (2).

$$Y := Ax \oplus c \, mod \, M \tag{2}$$

where 'A' is represented as affine matrix, 'x' is a vector that is multiplicative inverse of element of state matrix s', 'c' is affine constant i.e. 63 (01100011) and 'M' is irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. The affine matrix used is shown as under

$$\begin{bmatrix} b_0^{'} \\ b_1^{'} \\ b_2^{'} \\ b_3^{'} \\ b_4^{'} \\ b_5^{'} \\ b_6^{'} \\ b_7^{'} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$b_i^{'} = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i \tag{3}$$

The S-Box generated by equation (2) is represented in the Table (1).

Table 1. AES S-Box

	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

Following example shows S-Box elements calculation: **Example:-** To calculate the value of S-Box for the input '3D'. The first step is to calculate multiplicative inverse of

'3D' in GF(2^8). Polynomial representation of '3D' is as under:

$$a = 3D = x^5 + x^4 + x^3 + x^2 + x + 1$$

The multiplicative inverse of 'a' is 'BB_h' in hexadecimal and its binary representation is 10111011. The second step is to calculate affine transformation in GF(2) as following.

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

and the result after affine matrix transfromation is '27' and its binary representation is '00100111'. Similarly state matrix s is computed as s' as follows

$$s = \begin{bmatrix} 32 & 2d & 99 & 40 \\ aa & 53 & 8a & 1d \\ 3f & e4 & 2f & ac \\ 5a & 65 & c2 & 3c \end{bmatrix}, s' = \begin{bmatrix} 23 & d8 & ee & cd \\ ac & ed & 7c & a4 \\ 75 & 69 & 15 & 91 \\ be & 4d & 25 & eb \end{bmatrix}$$

- **The Shift Row Transformation:** It is a linear diffusion process, operating on individual rows in a 4 x 4 state matrix s. Rows of state matrix s are cyclically rotated to the left. In which the first row will remain same (no change), 2nd row will be shifted by one element, 3rd row will be shifted by 2 elements and 4th row of the matrix will be shifted by 3 elements . For example we get s' matrix from state matrix s after shift row in following transformation

$$s = \begin{bmatrix} 23 & d8 & ee & cd \\ ac & ed & 7c & a4 \\ 75 & 69 & 15 & 91 \\ be & 4d & 25 & eb \end{bmatrix}, s' = \begin{bmatrix} 23 & d8 & ee & 09 \\ ed & 7e & a4 & ac \\ 15 & 91 & 75 & 69 \\ eb & be & 4d & 25 \end{bmatrix}$$

- **The Mix Column Transformation:** In mix column column vector is multiplied with a fixed matrix, where the bytes are treated as a polynomials rather than numbers. In the following example state matrix s will be multiplied with the fixed mix column matrix and new s' matrix is obtained after mix column operation.

$$M = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}, s = \begin{bmatrix} 23 & d8 & ee & 09 \\ ed & 7e & a4 & ac \\ 15 & 91 & 75 & 69 \\ eb & be & 4d & 25 \end{bmatrix}$$

The new state after mix column will be $s' = M \times s$

$$s' = \begin{bmatrix} 94 & 06 & 08 & b1 \\ 36 & 32 & 6f & d4 \\ c2 & 46 & 77 & 18 \\ 50 & fb & 62 & 94 \end{bmatrix}$$

- **Key Expansion** In AES round keys are generated from cipher key as shown in fig. no. (1). The need of number of round keys to encrypt 128 bit block length data depends on the key length being used. 10 rounds are needed to encrypt 128 bit block data with 128 bit key length. So there is a need to generate 11 round keys for the same. In 'K' key matrix i-th column is denoted by W_i. The main idea in key expansion is to expand the 'K' matrix and the expended form of 'K' is called 'W'.

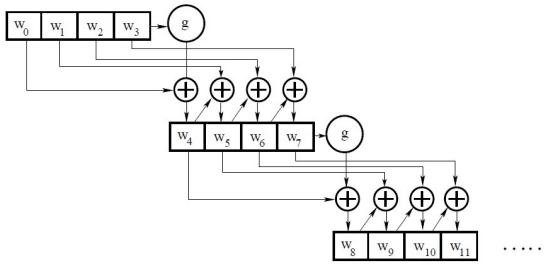

Figure 1. Key Expansion

The key expansion works between two cases where $N_k \leq 6$ and $N_k > 6$. When $N_k \leq 6$ the key expansion is as under:

$$W_i = \begin{cases} W_{i-N_k} \oplus SBox(S(W_{i-1})) \oplus rcon(\frac{i}{N_k}) \\ if\ i\ mod\ N_k = 0 \\ \\ W_{i-N_k} \oplus W_{i-1}\ if\ i\ modN_k \neq 0 \end{cases}, \quad (4)$$

where 'S' is a function that cyclically sifts the the elements of $W_{(i-1)}$. The S-Box function perform byte substitution operation on each element of the vector. $rcon(\frac{i}{N_k})$ is defined as a vector rcon(i)=[x^{i-1}, '00', '00','00'], with x^{i-1} being powers of 'x' in finite field $GF(2^8)$. When $N_k > 6$ the key expression equation will have slight change as under

$$W_i = \begin{cases} W_{i-N_k} \oplus SBox(S(W_{i-1})) \oplus rcon(\frac{i}{N_k}) \\ if\ i\ mod\ N_k = 0 \\ \\ W_{i-N_k} \oplus SBox(W_{i-1}) \oplus\ if\ i\ mod\ N_k = 4 \\ W_{i-N_k} \oplus W_{i-1}\ elsewhere \end{cases} \quad (5)$$

For example in order to obtain round keys from key matrix, it must be expended upto 40 columns to obtain round keys for 10 rounds of AES. Each round has a round key matrix of 4 colunms. In key expansion the elements W_i for $0 \leq i \leq 3$ are simply the i-th columns of Key matrix. To generate all 40 colunms the key schedule is followed shown in fig. no. 1. Following 'W' matrix can be obtained with key matrix.

$$Key = \begin{bmatrix} 00 & a5 & a8 & a0 \\ e9 & 09 & bb & 2a \\ c9 & d4 & b7 & ab \\ f2 & e8 & 60 & 08 \end{bmatrix}$$

'W' is the expended form of Key matrix

$$W = \begin{bmatrix} 00 & a5 & a8 & a0 & e4 & 41 & e9 & 49 & .....ea \\ e9 & 09 & bb & 2a & 8b & 82 & 39 & 13 & .....ef \\ c9 & d4 & b7 & ab & f9 & 2d & 9a & 31 & .....8b \\ f2 & e8 & 60 & 08 & 12 & fa & 9a & 92 & .....1a \end{bmatrix}$$

2.2 Decryption Process

Decryption process follows same steps as encryption process but in reverse order as shown in fig. no. (2)

- **Inverse Byte Substitution:** Inverse byte substitution is similar as byte substitution. In decryption process inverse S-Box is used. Inverse S-Box is shown in table (2)

Table 2. AES Inverse S-Box

	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
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

- **Inverse Shift Row:** The inverse shift row operation is same as shift row operation in encryption but the element shift is on the right instead of left.

- **Inverse Mix Columns:** In decryption inverse mix column matrix is used. The matrix is shown as under

$$\begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix}$$

- **Decryption Key Schedule:** The key schedule for decryption is same as encryption.

AES algorithm is shown in Figure No. 2

3 Essential Properties of S-Box

The strength of block ciphers, which work on substitution and permutation like AES heavily depends on the construction of S-Box (First layer of AES system) which must satisfy some essential properties to develop a secure crypto system. The crypto system that could resist algebraic attacks.

An S-Box S is a $m \times n$ mapping $S : \{0,1\}^n \rightarrow 0,1^m$ it converts input vector $x = [x_{n-1}, x_{n-2}, ..., x_1, x_0]$ to an output vector $y = [y_{m-1}, y_{m-2}, ..., y_1, y_0]$ $y = S(x)$.

Some essential properties of for constructing good S-Boxes are defined as under:

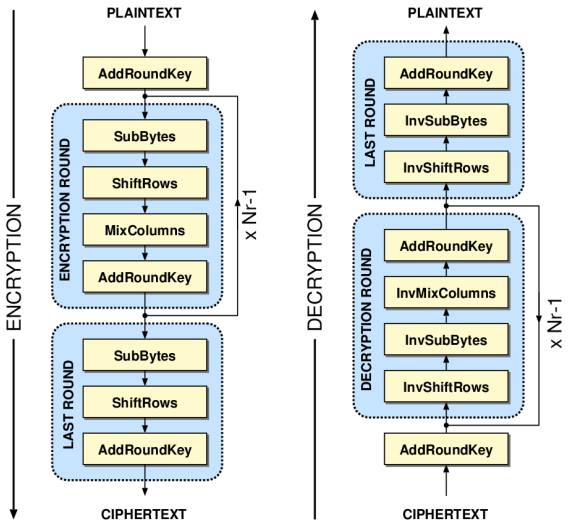

Figure 2. AES Algorithm

- **Hamming weight:** Hamming weight means number of ones contained by a vector [10].

$$hw(x) = \sum_{i=0}^{n-1} x_i. \tag{6}$$

- **Hamming Distance:** Hamming distance is calculated on two vectors which tells the number of bit positions where two vectors differ [10].

$$hd(x,y) = hw(x \oplus y) = \sum_{i=0}^{n-1} (x_i \oplus y_i). \tag{7}$$

- **Completeness:** A Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ is said to be complete if its output is dependent on all input bits [10]. The S-Box $S : \{0,1\}^n \rightarrow \{0,1\}^m$ is said to be complete [6] if the hamming weight for all vectors $x = [x_{n-1}, ..., x_1, x_0] \in \{0,1\}^n$ is 1, $hw(a) = 1$, there exists vectors $y = [y_{n-1}, ..., y_1, y_0] \in \{0,1\}^n$ such that $S(x)$ and $S(x \oplus y)$ differs at least on j bits for all $j \in \{n-1, ..., 1, 0\}$.

- **Balancedness:** A Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ is said to be balanced if output vector has equal number of zeros and ones [10] i.e. 2^{n-1} zeros or ones. S-Box $S : \{0,1\}^n \rightarrow \{0,1\}^m$ is balanced if all columns are balanced

$$\sum_{x \in \sum^n} f_j(x) \oplus f_j(x \oplus \alpha) = 2^{n-1}. \tag{8}$$

$$\forall_{0 \le j \le m-1} \; \forall_{\alpha \in \{0,1\}^n, w(\alpha)=1}$$

- **Nonlinearity:** The nonlinearity is defined as the minimum hamming distance between the function $f : \{0,1\}^n \rightarrow \{0,1\}$ and a set of its all affine functions [9] [10]. Nonlinearity specifies the distance to weak cryptographically affine functions.

$$nl(f) = \min_{l \in A_n} hd(f, l), \tag{9}$$

where A_n is a set of all affine functions over $\{0,1\}^n$ A function is said to be perfectly nonlinear if a non zero vector $x = [x_{n-1}, ..., x_1, x_0] \in \{0,1\}^n$ the values $f(y)$ and $f(y \oplus x)$ are equal to half arguments $y \in \{0,1\}^n$.

- **XOR Profile:** The Differential cryptanalysis was invented by Biham and Shamir [11] [10]. The attacker takes advantage of XOR distribution table and exploits imbalances in XOR distribution table for S-Box to know output XOR from the Input XOR. XOR table consists of 2^n rows of input differences and 2^m columns of output differences. The XOR table entries of S-Box S corresponding to (α, β) are

$$XOR(\alpha, \beta) = \#\{x \in \{0,1\}^n : S(x) \oplus S(x \oplus \alpha) = \beta\}, \tag{10}$$

where $\#$ denotes the cardinality of the set, $\alpha \in \{0,1\}^n$ $0, \beta \in \{0,1\}^m$.
Some of the properties of XOR distribution table are that all the entries of XOR will be zeros or positive even integers, the sum of all the entries in a row will be equal to 2^n, the input difference α may cause output difference β with probability $p = \frac{\delta}{2^n}$ where δ is the eirty of (α, β) in XOR table, if an entry (α, β) in XOR table is zero, then input difference α can not cause output difference β.

- **Strict Avalanche Criterion** The SAC is the concept given by Webster and Tavares in 1985 [7]. A function satisfies SAC if one bit of the plain text or key is complemented then the output is changed with the probability of $\frac{1}{2}$ bits. SAC is calculated by equation 11

$$\alpha := \sum_{x \epsilon Z_2^n} f(x) \oplus f(x \oplus e_i), \tag{11}$$

where x and e are two n bit vectors which differ only in one bit i. The Boolean function $f(x)$ accomplishes SAC criterion if and only if $\alpha = 2^{n-1}$ for all i, $0 \leq i \leq n-1$.

- **Bit Independence Criteria**

The concept of BIC was introduced by Webster and Tavares [7]. According to BIC, with the change of single bit in plain text or a key for a given set of avalanche vectors, all avalanche variables of respective vectors should be pairwise independent. The correlation coefficient of j^{th} and k^{th} components of avalanche vector D^{ei} is calculated to measure the bit independence property of over all input pairs P and P_i, which differ only in bit i ($P_i = P \oplus e_i$).

$$BIC^{e_i}(d_j, d_k) = |corr(d_j^{e_i}, d_k^{e_i})|, \tag{12}$$

Then the overall BIC is defined as:

$$BIC(f) = max_{1 \leq i \leq n, 1 \leq j,k \leq m, j \neq k} BIC^{e_i}(d_j, d_k), \tag{13}$$

BIC(f) is defined in the range 0 and 1. It is ideally equal to zero, and in worst case it is equal to one.

4 AES Security

There are number of attacks designed to break block ciphers like AES. Those are algebraic attacks and side channel attacks. The algebraic attacks analysis the algebraic structures of block cipher while side channel attacks are on physical implementation of block ciphers on hardware level. In this paper algebraic attacks are taken into consideration. Algebraic attacks are linear cryptanalysis, Differential cryptanalysis, Boomerage Attacks, Interpolation attacks, Slide attack, Multiset attacks include round 4, 6, 7, 8 and 9 rounds, XL and XSL attacks.

The linear cryptanalysis attack was invented by Mastsui in 1993 [13]. This is a known plain-text attack. It exploits the linear relationship between input and output of a cipher to discover cipher key bits. This is done by approximating the S-Boxes by linear expressions that have high probability bias larger than $1/2$, then it finds approximation of entire cipher with combinations of plaintext, ciphertext and key bits. It is found that there are no 4 round linear trails with bias above 2^{-75} and no 8 round linear trails with bias above 2^{-150} [12]. This is sufficient to resist against this attack.

Differential cryptanalysis was invented by Biham and Shamir in 1990 [11]. This is a chosen plaintext attack. In this attack attacker analysis the effect of difference of input pair of plaintext on the difference of output pair to discover the key bits. The idea is to find the high probability difference pairs for an S-Box under attack. These input output differences are used to form a differential trail for the entire cipher. It is found that there are no 4 round differential trails with bias above 2^{-150} and no 8 round differential trails with bias above 2^{-300} [12]. This is sufficient to resist against this attack.

In Boomerage attack the attacker propagates highly probable differential patterns from both ends of the cipher to find which differences agree in the middle [14]. This is also called meet in middle attack. But due to low differential probabilities (i.e. 2^{-150} for 4 rounds) and good diffusion properties of mix column and shift row layer of AES this attack is not successful.

In Interpolation attack the plaintext and ciphertext pairs are converted into polynomials [15]. But this attack works

on ciphers which have small degree polynomials. AES polynomials have high algebraic degree i.e. 254 because of its affine transformation in S-Box construction, which provides good diffusion properties. So this attack is not successful on full AES.

Slide attack was invented by Biryukov and Wagner [16]. It exploits the weakness of round function, if the round function is weak then one can easly discover key from one slid pair of two round functions. As the key schedule of AES uses different constant to generate different round keys and provides enough dissimilarity among rounds. These round function dissimilarities makes AES resistant to this attack.

Square attack was introduced for reduced round version of AES and was invented by Lars Knudsen and first time applied to block cipher square [17]. It is a chosen plain text attack. In this attack the attacker uses carefully chosen set of plaintexts and multisets to break substitution permutation network (SPN) of a block cipher, where multisets mean the group of values which appears many times in cipher. Initially this attack was formed to attack 4 rounds of 128 bit version of AES [18] and then extended up to 6 rounds to attack 192 and 256 bit version of AES. An improved 7-round attack was introduced on AES for all key (128, 192 & 256) versions of AES. But the 7-round attack was slower than the exhaustive search on 128 and 192 bit AES [19], which makes it impractical. Further 8-round attack was introduced in [20] on 192 and 256 bit version of AES.

4-round square attack was used to attack 4 round of 128 bit AES, which needs 2^9 plaintext and time complexity is 2^9, 5-round square attack requires huge plaintexts that was 2^{32} and complexity also increased to 2^{40}, 6-round square attack needs 2^{32} plaintexts with complexity of 2^{80} ciphertext executions. But with the improvement in 6-round attack, in which partial sum technique was used the complexity reduced to 2^{40}, which was huge improvement over previous 6-round attack. 7-round square attack was made on 192 and 256 bit AES. This attack attacks the 7th round of AES by guessing all the 128 key bits of last round and the complexity of this for 192 bit AES is 2^{176} and for 256 bit AES is 2^{192}. In first improvement of 7-round attack for 192 bit AES the complexity reduced to 2^{155}, in which partial sum technique was used and in second improvement the complexity reduced to 2^{172} for 256 bit AES in which new technique herd was used (structure of 2^{120} encryptions called herd). 8-round square attack was discovered by improving 7-round attack and to implement this attack the complexity of 2^{204} ciphertext execution time for 256 bit AES was required and 2^{188} ciphertext execution time for 192 bit AES was needed with $2^{128} - 2^{119}$ plaintexts, which was practically infeasible.

XL attack on AES is an algebraic attack which exploits the algebraic structure of AES S-Box. AES S-Box is the only nonlinear part of algorithm. XL attack tries to solve the multivariate polynomial equations (MQ) problem, which proved very inefficient on AES because AES S-Box polynomial structure is based on high algebraic degree i.e. 254 in GF(2^8). N. T. Courtois *(*et. al) [21] invented a method to solve MQ problem (also called NP-hard) in which they represented 128 bit AES with 1600 variables and 8000 quadratic equations and presented that by using eXtended Linearization the complexity of solving these equations is 2^{330}. In the improvement in XL attack if the MQ is sparse then this can be solved by new method called XSL (eXtended Sparse Linearization). This improvement reduced the complexity for AES just 2^{256}, which is not sufficient to break AES.

5 Existing Work

The only nonlinear part of AES algotithm is S-Box which is fixed throughout the algorithm and in previous section we analyze that the cryptanalyst tried to exploit this weakness. So to improve the immunity of AES S-Box against algebraic attacks much research had been carried out by different people to make the S-Box dynamic. We have covered the overview of these in this section.

Krishnamurthy et. al [22] used AES-KDS block cipher which worked on 128 bit key length as well as data length, which used 5 stages instead of 4 stages used in AES. On the encryption side the extra stage that is rotate S-Box added on the top of existing stages which rotates the elements of S-Box on the basis of round key and on decryption side and inverse S-Box is used which nullify the effect of rotate S-Box state. This extra stage which is added on the encryption side makes the S-Box dynamic. This algorithm used four cases to provide different level of security. First case provides moderate level of security in whcih S-Box rotation is based on only one byte of the round key. Second case provides high level of security in which S-Box values are rotated on the bases of the whole round key. Third stage provides very high level of security by creating two subset of the round keys from key expansion algorithm in which one set of keys generated are used to find the value on which the values of the S-Box are rotated and the other set of keys are used to find the key for add round key operation. Stage four provides very high level security in which the S-Box values are dependent on the whole key of the key generated from the set of keys of the set one.

Piotr Mroczkowski [23] presents a general framework for improving the security of the cryptosystem based on the symmetric block cipher. The main idea is based on possibility of chancing substitution boxes (called S-boxes) in encryption/decryption algorithm. In order to make it possible he used pseudorandom sequences to generate identical boxes for encryption and decryption.

Abd-ElGhafar et. al [24] presented another technique in

which RC4 algorithm was used to generate key dependent dynamic S-Boxes.In this algorithm all the values of S-Box are dependent on input key if any byte of input key is changed then different 256 values were generated, like this 256! S-Boxes could be generated.

Kazlauskas et. al [25] proposed an approach to generate the random S-boxes changing for every change of the secret key. The fact that the S-boxes are randomly key-dependent and unknown is the main strength of the new approach, since both linear and differential cryptanalysis require known S-boxes. They analyzed the AES algorithm, substitution S-boxes, linear and differential cryptanalysis, and described a randomly key-dependent S-box and inverse S-box generation algorithm.

Ghada Zaibi et. al [26] presented dynamic S-Boxes based on one-dimensional chaotic maps compared to classic S-Box and evaluated the more suitable one dimensional map to construct a dynamic S-Box used in the AES algorithm.

Jie Cui et. al [27] proposed algorithm to increase the complexity and security of AES S-box by modifying the affine transformation and adding an affine transformation. Performance analysis demonstrates that the improved AES S-box showed improvement in affine transformation period, iterative period and distance to SAC.

Anna Grocholewska-Czurylo [28] presented an algorithm to construct 8×8 S-Boxes on the basis of random irreducible polynomial chosen.

Julia Juremi et. al [29] presented algorithm involved key expansion algorithm together with S-box rotation and that property was used to make the S-box key-dependent to provide a better security to the block cipher.

Razi Hosseinkhani et. al [30] presented dynamic S-Boxes on the basis of cipher key. They used cipher key to dynamically generate S-Boxes.

Oleksandr Kazymyrov et. al [31] proposed an improved gradient descent method for increasing performance of nonlinear vectorial Boolean functions generation with optimal cryptographic properties. Substitutions were generated by proposed method for the most common 8-bits input and output messages have nonlinearity 104, 8 uniformity and algebraic immunity 3.

Mona Dara et. al [32] used Chaotic Logistic Map to generate S-box for AES using its cipher key. Proposed S-box were analyzed and tested for avalanche effect, strict avalanche effect, bit independency criterion, non-linearity, input/output XOR distribution and key sensitivity.

Eman Mohammed Mahmoud et. al [33] used another technique in which PN Sequence generator was used to generate perfect random sequence of bits. This approach used LFSR (Linear Feedback Shift Register) to generate key dependent dynamic S-Boxes.

Sliman Arrag et. al [34] proposed an approach of nonlinear transformation algorithm for AES S-Box to enhance the complexity of the S-Box structure, They made AES stronger by using Dynamic S-box by using look up table S-box and Key expansion schedule was also modified.

Fatma Ahmed et. al [35] modified AES with S-boxes bank to be acted like rotor mechanism and dynamic key MDS matrix (SDK-AES). They tried to make AES key dependent and resist the frequency attack.

Adi Narayana Reddy K et. al [36] presented a dynamic S-Box by adding a secrete value to the static index to shift the substitution to a secrate location. For added security they have also generated variable sub keys by using sequence of pseudo random numbers. They tested this algorithm on the basis of correlation coefficient (BIC) and strict avalanche criteria (SAC).

Kazlauskas et. al [37] modified their key dependent S-Box generation algorithm and presented a fast algorithm to generate key dependent S-Boxes. Author checked the randomness of generated S-Boxes by applying NIST tests. The author claims that new S-Boxes provide algorithm resistance to algebraic attacks and algebraic properties of new S-Boxes are as good as AES S-Boxes.

Balajee Maram et. al [38] proposed a new algorithm to generate S-Boxes based on Pseudo-Random generator. The author claims that this algorithm generates S-Boxes in less time then other existing algorithms and the generated S-Boxes have good liner and differential properties.

Shishir Katiyar et. al [39] proposed a new algorithm which generates S-Boxes based on one-dimensional chaotic map (logistic and PWLCM). The new S-Boxes are checked against the AES S-Boxes and claims that they are as good as AES S-Boxes.

Tianyong Ao et. al [40] proposed an algorithm in which they generated S-Boxes based on key dependent affine transformation. The new generated S-Boxes are tested on the basis of nonlinearity, XOR profile etc. and found that they are as good as AES-SBox.

6 Security Analysis of Different Algorithms

To make a cryptographic algorithm secure against various algebraic attacks it should comply with some standard tests like Non-linearity, Bit independence criteria, XOR profile and strict avalanche criteria. For AES the nonlinearity value is $nl = 112$ which is close to half as mentioned in equation (9). For AES in XOR distribution table the maximum probability of output differences

caused by input differences are $\frac{4}{256}$, which are very low. Strict avalanche criteria for AES as mentioned in equation (11) must be around 50% that means output bits in cipher text should be changed by probability with $\frac{1}{2}$ when single bit of plaintext or key is complemented. The another important test is BIC (bit independence criteria), which shows the correlation between the pair of cipher texts produced by changing one bit of plaintext or key. It should range between -1 and 1 and in worst case it is equal to 0.

Not all the authors analyzed all parameters. They focused on differ parameters.

Strict Avalanche Criteria: As shown in table no. (3) algorithms [22], [23], [24], [26], [27], [29], [32], [33], [35], [36], [37] and [38] worked on SAC. The average SAC values of all algorithms range from 46% to 57% which are around standard value 50%. This means that if a single bit or plain text or key has been changed then the output bits in a vector should change with the probability of one half. So the result is close to the AES standard result and sufficient to resist against algebraic attacks.

Non-Linearity: Non-linearity as shown in table no. (3) algorithms [23], [27], [28], [31] and[40] worked on Non-linearity parameter. The non-linearity for these algorithms are 98, 112, 112, 104 and 112 respectively, which are around standard non-linearity of AES ranges between 112 to 144. This means that in linear approximation table (256×256 matrix) for different number of vectors the non-linearity value is different, but the minimum value is 112. So the results are close to AES and all the algorithms are resistant against linear cryptanalysis.

Bit Independence Criteria: Bit independence criteria as shown in table no. (3) algorithms [24], [25], [32], [36] and[37] worked on BIC parameter. The BIC for these algorithm are 0.4688, 0.4439, 0.4993, -0.0545 and 0.443 respectively, which are between -1 and $+1$ shows that with one bit change in key or plain text the output avalanche vectors are less correlated. This is an essential criteria which shows that with the small change in key or plain text the elements of output vectors should be pair wise independent. So these algorithms meet that criteria.

XOR Profile: XOR criteria as shown in table no. (3) algorithms [26], [27], [28] and [40] worked on Xor profile parameter. The XOR values of these algorithms are $\frac{10}{256}$ for [26] and $\frac{4}{256}$ for [27], [28] and [40]. XOR profile is calculated by constructing difference distribution table (256×256 matrix) in which the effect of input difference on the output difference is observed. The maximum probability of such values in DDT table is $\frac{4}{256}$, which makes it resistant to differential cryptanalysis. To cryptanalysis the algorithm the attacker is interested in higher values in DDT. The result of algorithms [26] and [27] are same as AES so they are resistant to this attack.

Table 3. Analysis of Dynamic S-Box Algorithms

Paper	SAC	NL	BIC	XOR
[22]	48.8%	-	-	-
[23]	46%	98	-	-
[24]	52.34%	-	0.4688	-
[25]	-	-	0.4439	-
[26]	51.25%	-	-	$\frac{10}{256}$
[27]	50.2%	112	-	$\frac{4}{256}$
[28]	-	112	-	$\frac{4}{256}$
[29]	48.03%	-	-	-
[31]	-	104	-	-
[32]	51.7%	-	0.4993	-
[33]	51.31%	-	-	-
[35]	51.43%	-	-	-
[36]	49%	-	-0.0545	-
[37]	48.22%	-	0.443	-
[38]	57%	-	-	-
[40]	-	112	-	$\frac{4}{256}$

So all in all we came to the conclusion that the proposed algorithms have good liner and differential properties to resist various algebraic attacks. The results are summarized in table no. (3)

7 Conclusion

In this paper we give introduction of AES algoritm. The overview of algebaric attacks on AES and different dynamic S-Box algorithms is given. All the techniques discussed in this paper enhance the security of existing AES algorithm by introducing the dynamic S-Box instead of static one used in AES algorithm. The results of all algorithms are comparable and very close to the AES algorithm. These algorithms tried to provide security against the different algebraic attacks by increasing the difficulty for the cryptanalyst by increasing the confusion in the first stage of AES algorithm.

REFERENCES

[1] Forouzan B., *Traditional Symmetric-Key Cipher*, In Introduction to Cryptography and Network Security, 1st ed. New York: McGraw-Hill, (2008):55-96.

[2] Forouzan B., *Traditional Asymmetric-Key Cryptography*, In Introduction to Cryptography and Network Security, 1st ed. New York: McGraw-Hill, (2008):293-335.

[3] Daemen J. and Rijmen V. , *The Design of Rijndael: AES The Advanced Encryption Standard.* Springer-Verlag, 2002.

[4] Daemen J., Rijmen V., *The block cipher Rijindael*, Proceedings of the Third International Conference on smart card Research and Applications, CARDIS98, Lecture Notes in computer Science, 1820(2000):277-284.

[5] Federal Information Processing Standards Publications (FIPS 197), Advanced Encryption Standard (AES) , 26 Nov. 2001.

[6] Kam J., Davida G., *Structured Design of Substitution-Permutation Encryption Networks*, IEEE Transactions on Computers, 28(10)(1979):747-753.

[7] Webster A., Tavares S., *On the Design of S-boxes*, Advances in Cryptology CRYPTO-1985, LNCS 218, Springer-Verlag, 1985

[8] Forre R. *The strict avalanche criterion: spectral properties of booleans functions and an extended definition.* Advances in cryptology, in: S.Goldwasser(Ed), Crypto88, Lecture Notes in Computer Science, 403(1990):450-468

[9] Mister S., Adams C., *Practical S-box design, Workshop on Selected Areas in Cryptography*, SAC 1996, Workshop Record, 1996.

[10] Rodwald P. and Mroczkowski P., *How to create good s-boxes?*, 1st International Conference for Young Researchersin Computer Science, Control, Electrical Engineering and Telecommunications, ICYR, 2006.

[11] Biham E., Shamir A., *Differential cryptanalysis of DES-like cryptosystems*, Advances in Cryptology CRYPTO 1990, Springer-Verlag, 1990

[12] Daemen J., Rijmen V., *The Rijndael Block Cipher - AES Proposal*, available from: http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael-ammended.pdf

[13] Matsui M. *Linear cryptanalysis method for DES cipher*, Eurocrypt, Springer LNCS, 765(1993):386-397.

[14] Wagner D., *The boomerang attack*", Fast Software Encryption, Springer LNCS, 1636(1999):156-170.

[15] Jakobsen T., Knudsen L., *The Interpolation Attack on Block Ciphers*, Fast Software Encryption, Springer LNCS, 1267(1997):28-40.

[16] Biryukov A., Wagner D., *Slide Attacks*, Fast Software Encryption, Springer LNCS, 1636(1999):245-259.

[17] Daemen J., Knudsen L. and Rijmen V., *The block Cipher Square*. Fast Software Encryption 97, Springer-Verlag, (1997):149-165.

[18] Daemen J. and Rijmen V., *AES Proposal: Rijndael, second Version*, AES submission.

[19] Lucks S., *Attacking Seven Rounds of Rijndeal under 192-bit and 256-bit keys*, The third Advanced Encryption Standard Candidate Conference, NIST, (2000):215-29.

[20] Ferguson N., Kelsey J., Lucks S., Schneier B., Stay M., Wagner D. and Whiting D., *Improved Cryptanalysis of Rijndeal*, Fast Software Encryption 2000, Lecture notes in Computer Science, Springer-Verlag, 1978(2001):213-230.

[21] Nicolas T. Courtois and Josef Pieprzyk, *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, ASIACRYPT, (2002):267-287.

[22] Krishnamurthy G N. and Ramaswamy V. *Making AES Stronger: AES with Key Dependent S-Box*, International Journal of Computer Science and Network Security, 9(8)(2008).

[23] Piotr M., *Generating Pseudorandom S-Boxes a Method of Improving the Security of Cryptosystems Based on Block Ciphers*, Journal of Telecommunications and Information Technology, 2009.

[24] ElGhafar A., Rohiem A., Diaa A., Mohammed F., *Generation of AES Key Dependent S-Boxes using RC4 Algorithm*, 13th International Conference on Aerospace Sciences & Aviation Technology, ASAT- 13, May 26-28, 2009.

[25] Kazys K., Jaunius K. *Key-Dependent S-Box Generation in AES Block Cipher System*, INFORMATICA, (2009):23-34

[26] Ghada Z., Abdennaceur K., Fabrice P. and Daniele F., *On Dynamic chaotic S-BOX*, IEEE, 2009

[27] Cui J., Huang L., Zhong H., Chang C. and Yang W. *An Improved AES S-Box and Its Performance Analysis*, International Journal of Innovative Computing, Information and Control, 7(2011).

[28] Anna G., *Cryptographic properties of modified AES-like S-boxes*, Annales UMCS Informatica AI XI, 2(2011):37-48.

[29] Julia Juremi Ramlan Mahmod Salasiah Sulaiman Jazrin Ramli, *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.

[30] Hosseinkhani R. and Haj Seyyed Javadi H. *Using Cipher Key to Generate Dynamic S-Box in AES Cipher System*, International Journal of Computer Science and Security (IJCSS),6(1)(2012).

[31] Kazymyrov O.,Kazymyrova V.,Oliynykov R., *A Method For Generation Of High-Nonlinear S-Boxes Based On Gradient Descent*, IACR Cryptology, 2013

[32] Dara M. and Manochehri K., *A Novel Method for Designing S-Boxes Based on Chaotic Logistic Maps Using Cipher Key*, World Applied Sciences Journal 28(12)(2013):2003-2009.

[33] Mohammed Mahmoud E., Abd El Hafez A., Talaat A. and Zekry A. *Dynamic AES-128 with key-dependent s-box*, International Journal of Engineering Research and Applications, 3(1)(2013):1662-1670.

[34] Arrag S., Hamdoun A., Tragha A. and Eddine Khamlich S. *Implementation Of Stronger AES By Using Dynamic S-Box Dependent Of Master Key*, Journal of Theoretical and Applied Information Technology, 2013.

[35] Ahmed F. and Elkamchouchi D., *Strongest AES with S-Boxes Bank and Dynamic Key MDS Matrix (SDK-AES)*, International Journal of Computer and Communication Engineering, 2013.

[36] Adi Narayana Reddy K. and Vishnuvardhan B., *Secure Linear Transformation Based Cryptosystem using Dynamic Byte Substitution*, International Journal of Security, 3(8)(2014).

[37] Kazys KAZLAUSKAS, Gytis VAICEKAUSKAS, Robertas SMALIUKAS, *An Algorithm for Key-Dependent S-Box Generation in Block Cipher System*, INFORMATICA, 26(1)(2105):51-65.

[38] Balajee Maram K., Gnanasekar J. M., *Evaluation of Key Dependent S-Box Based Data Security Algorithm using Hamming Distance and Balanced Output*, TEM Journal, 5(1)(2016).

[39] Katiyar S., Jeyanthi N., *Pure Dynamic S-box Construction*, International Journal of Computers, 2016.

[40] Tianyong Ao, Jinli Rao, Kui Dai, and Xuecheng Zou, *Construction of High Quality Key-dependent S-Box*, IAENG International Journal of Computer Science, 2017.