Pure Dynamic S-box Construction

Shishir Katiyar, N. Jeyanthi
School of Information Technology and Engineering,
VIT University, Vellore – 632 014, Tamilnadu, India
shishrshiv007@gmail.com

***Abstract* – The S-box or substitution table used to provide non-linearity to encrypting algorithm. There are two types of S-box design such as Static S-box and Dynamic S-box. This paper reviews and analyses the efficiency of Dynamic S-box over Static S-box. This proposed approach has two fold, first fold is to design a Dynamic S-box based on the one-dimensional chaotic map (logistic and PWLCM) and compare to the classical S-box in terms of on confusion. Second fold is to calculate the one-dimensional map equation to construct Dynamic S-box. Finally analyse the properties of S-box on the constructed key-dependent Dynamic S-box.**

Keywords – S-box, Chaos equation, security analysis, static, dynamic

I. INTRODUCTION

The S-box is a Substitution box that works as the non-linearity to provide confusion, strength and resistance to cryptanalysis. The strength of the algorithms such as DES, AES, Blowfish and Twofish depends on the S-box. Most encryption algorithms use Static S-box, which we have generated in advance. Due to the availability of S-boxes in advance we face many security issues. However, if Dynamic S-box created with the help of key then it could resist more, over the Static S-boxes. Various conventional methods have been proposed to design S-box. This paper focuses on the basic properties of S-boxes and learns from the review of the existing S-boxes about security issues. Proposed methodology adopts key-dependent Dynamic S-box based on the one-dimensional chaotic map equation.

II. STATIC AND DYNAMIC S-BOX PROPERTIES

S-boxes can be constructed in two distinct ways: Static and Dynamic. In Static S-box, input vector values are not changed while in Dynamic S-box input vector value changes. Following Static and Dynamic view:

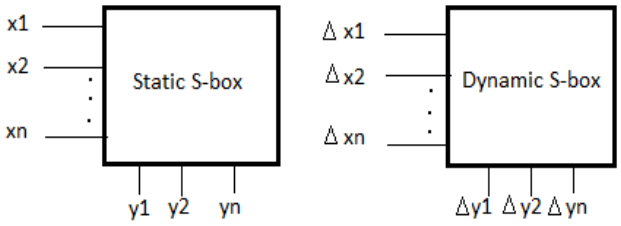


Figure 1: Static and Dynamic S-Box

Properties of Static and dynamic S-box were defined using entropy [1]. A metric to measure the randomness of data is entropy defined by H (Z) for random variable "z" as follows:

$$H(Z) = \sum_{i=1}^{n} P(Z_i)\, log_2(Z_i^{-1}) \qquad (1)$$

High entropy means difficult to guess the values. S-box should satisfy better entropy values.

A. S-BOX PROPERTIES

1) Completeness: A Boolean function $f: \{0, 1\}^n \rightarrow \{0, 1\}$ is complete if its output depends on all bits in the input. [6]

2) Balancedness: A Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ is balanced if its truth table has 2^(n-1) zeros or ones.

3) Non-linearity: It is a very important property for s-box because of which S-boxes have been used in cryptographic algorithms. A Boolean function $f: \{0, 1\}^n \rightarrow \{0, 1\}$ is non-linear, defined as least hamming distance between function and set of all affine function. [8]
Non-linearity of S-box S: $\{0,1\}n \rightarrow \{0,1\}^m$ is defined as the all nonzero linear combinations of m Boolean functions $fi : \{0,1\}^n \rightarrow \{0,1\}$, $i=m-1,\ldots 1,0$.

4) XOR Profile: This property important to overcome the attack based on the imbalances in "XOR distribution table" for S-box; use to predict XOR output when there is XOR input.
XOR distribution table consists of:

- 2^n rows, for input differences and
- 2^m columns, for output differences.

Properties of XOR profiles are:

- All entries are zeros or positive even integers in XOR table.
- The sum of entries in each row equal to 2^n.
- An input difference Z may cause the output difference Y with probability P = U/ (2^n).
  Where U is entry of (Z, Y) in the XOR table.

5) Strict Avalanche Effect criterion: Function satisfies the strict avalanche effect criterion, if each of the output bits change with probability of $1/2$ whenever single bit is complemented.

A Boolean function $f:\{0,1\}n \rightarrow \{0,1\}$ satisfies the SAC if the f(x XOR y) is balanced for every vectors "x" $\in \{0,1\}^n$ and for every vectors y $\in \{0,1\}^n$, which Hamming weight = 1.

6) Bit Independence criterion: The Boolean function $f:\{0,1\}^n \rightarrow \{0,1\}$ satisfies bit independence criterion, if for every input bit $k \in \{0,1,..., m-1\}$ and for every output pair of bits $m, l \in \{0,1,..., p-1\}$, $m \neq l$, the change of bit "k" on the input which effect on the output independent changes of bits (m,l).

III. LITERATURE REVIEW

Many encryption algorithms use the static and dynamic S-box. I take a review of some shown below:

1) Camellia S-Box: Camellia S-Box follows the Avalanche-property. S_1 is defined by the arithmetic operation GF (2) ^8 which is pre-computed and S_2, S_3, S_4 computed using the looked up values of S_1 table.

$$S1: \quad x(8)\ h(g(\ f\ (c5\ x(8)\ ))) 6e \qquad (2)$$

$$S_2 \quad :x_{(8)}\ S_1 \quad (x_{(8)}) <<<1 \qquad (3)$$

$$S_3 \quad :x_{(8)}\ S_1 (x_{(8)}) >>>1 \qquad (4)$$

$$S_4 \quad :x_{(8)}\ S_1 \quad (x_{(8)} <<<1) \qquad (5)$$

Where "x" is the 8-bit element in GF(2)^8.
High degree of the Boolean polynomial of every output bit of the S-box makes it difficult to attack camellia by higher order differential attacks.[2]
But Integral Attack on 10/11-Round Camellia-128 is possible easily.[3]

*2)MD2 S-Box has value of "pi" :*In MD2 for S-box use the value of "Pi" which can pre-calculated and can be attacked or weaknesses found by attacker who has a magnetic tape with millions of digits of "Pi". So, permutation table in MD2 is not secret.[5]
Differential and linear cryptanalysis is very powerful techniques, especially against S-boxes. In the case of Hash functions, check for the pair of inputs for which the output difference is zero, or fir which the output difference is equal to the input differences [4].

3) DES S-Box: Both differential cryptanalysis and the S-Box pairs/triplets attacks are based on the non-uniform distribution of S-Box outputs.
DES is resistant to differential cryptanalysis. Biham and Shamir modified the S-Boxes (changing the order, using random permutation in the rows etc.) that can significantly weakness of DES with respect to differential cryptanalysis. We can calculate S-Box pairs of output by distribution tables for any number in the rounds of DES, using function (calculates the distribution for the XOR of "n" pairs of S-Box).[7]

4) AES S-Box: In AES also we face weakness regarding the S-Box. Static S-Box allows data analyses attack and captures of sub-keys using inverse Sub-Byte knowing inverse S-Box.
Mathematical attack against the AES algorithm is easier, when S-box contain static values for all rounds and for all entries. AES not resistant to Timing attack, Cache-timing attacks.[9]
For attacker it's infeasible as the "n" value increase ("n" = 128, $2^{38} = (10)^{38}$ and $((10)^{38})!$) possible S-Box can be generate which is not easy to use for attack.

5) Blowfish S-Box: This encryption algorithm uses key-dependent Dynamic S-Box. Therefore, Blowfish is not attackable by some methods by differential and linear cryptanalysis. Blowfish seems to be weak on Vandenay Attack (presence of weak keys in the cipher). "Weak keys" resultant because of at least one collision into one the four S-boxes in Blowfish. The condition is:

$$\text{S-box (P)} = \text{S-box (Q)} \qquad (6)$$

And see the relation

$$k = P \text{ xor } Q \qquad (7)$$

Where P, Q are different bytes and "k" is the key (weak).
Consequences we can perform "chosen plaintext" attack due to collision happens in [2^ (-21)!].[11]

6) Twofish S-Box: This encryption algorithm also based on the key-dependent. Avalanche effect used by cryptanalysis to reduce the number of active characteristics in S-Boxes. The changes of 8 bits lead to be conflicts that hard to resolve. However, the use of key-dependent dynamic S-box in Twofish may well be some keys that resolve any potential conflict.[12]

We conclude with above discussed encryption algorithm that Dynamic S-box provide more diffusion over Static S-box as in the Blowfish, Twofish. No doubt, DES static S-boxes are strong, but use dynamic nature can give more non-linearity. Consequence, the flexibility of key-dependent S-boxes can create difficultly for attackers.

IV. DETAILED PROBLEM DEFINITION

Three strong property of S-box that is Non-linearity(NL), Avalanche Effect criterion (AEC) and Bit Independence criterion(BIC) due to which S-box is important in terms of security, can provide diffusion and confusion. In the case, when use the Static S-boxes which is pre-defined may open the door for attackers because of "static behaviour" where values are fixed (Camellia, DES). We have another option that is Dynamic S-box based on "dynamic behaviour" where values are not fixed (key dependent rotation) but create when

require encryption (Blowfish, Twofish). Which give better result over static nature. As known that key is the heart of encryption algorithm, it will be benefited if use the key in the creation of S-Box to provide "dynamic nature".

V. SOLUTION METHODOLOGY

S-Box work as the non-linear element in the encryption used to provide security. Using chaos equation which has the random behaviour can create the "good" S-Boxes by slightly modification. Many chaotic maps equations are present, in this paper one of them is use that called *Logistic map* equation which defined as.

$$X_{n+1} = r^* X_n (1-X_n) \qquad (8)$$

Where X_n is between 0 and 1, r is a positive number between 1 and 4[13]. In this paper proposed the method to create the key-dependent Dynamic S-box based on the Logistic map. Use the key in the equation with some modification. The proposed algorithm given below:

```
for(i= 0; i<= dsize; i++)
 {
   temp = a;
   p=k1*(dsize-i)
      k1=k2*(1-k2)+(k3/p)+(k4/p);
   k2= temp;
   System.out.println(a);
   V = afterDecimal(k1);
   System.out.println(V);
      x = V % MDv;
  System.out.println("  " + x );
 }
```

S-box is strong if it satisfies SAC, BI and NL property. For dynamic S-box it is difficult to achieve because of unpredictable nature of mathematical concept use to provide dynamic behavior for S-box. Furthermore, equation use to create S-box should be carefully chosen and how key bits are distributed among the variables in the equations is important that can affect the properties. Using the proposed algorithm get the S-box named as *Pure Dynamic S-box*. "Pure Dynamic" is used because everything is dynamic size, modulo value, input value use to create S-box to provide security. As we know, number of unknown variables used in encryption algorithm proportional to security.

VI. EXPERIMENTS AND ANALYSIS

Using the Algorithm: 1 creates the S-Box Table 1. This S-box satisfies all three properties completely. The only problems with dynamic S-box that may be possibility of multiple values because we can't predict the nature between the equation used and any key subparts. To give better result, modified the equation as :

$$k_1=k_2*(1-k_2)+(k_3/p)+(k_4/p). \qquad (9)$$

where k_1, k_2, k_3, k_4 all are subpart of key which use in the encryption algorithm size of subparts depends on the encryption algorithm and "p" provide the randomness due to change in every iteration. The value of *MDv* is "modulo-dependent value" and *dsize* is "dependent size" this value also depends on encryption algorithm, *MDv* and *dsize* value are proportional to security. For analysis, take some sample subparts of key of size less than 2^8, MDv and dsize value is 2^8 and 2^6 respectively. Created S-box shown below at phase1 .

TABLE I. S-Box at phase1

179	92	60	229	122	70	150	143
163	202	171	253	180	41	191	83
205	122	223	160	244	198	12	236
40	19	71	60	105	102	143	147
185	195	232	247	27	47	84	109
147	177	218	254	43	85	136	186
244	48	117	190	18	110	216	81
224	138	93	106	217	248	181	97

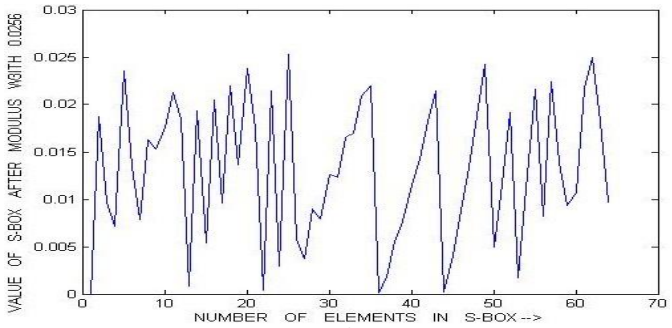
As per analysis, there is the possibility of multiple values in S-box. To resolve this problem assign the different values which are not used in the calculated S-box. For example, In above S-box 122, 60, 244, 143, 147 is repeating therefore assign the value from the set (values not used limit 256) may be 1, 2, and 255 as soon based on the implementation. Here use hash data structure in implementation and replace by the first value that not used is 1, 199, 271, 111, and 41. Final S-box at phase2 is shown below:

TABLE II Final S-Box in phase2

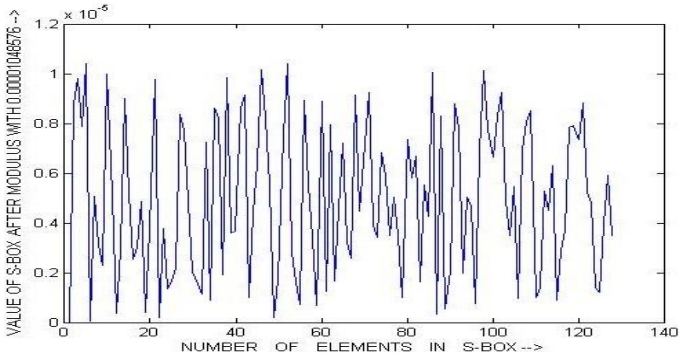
179	92	60	229	122	70	150	143
163	202	171	253	180	41	191	83
205	1	223	160	244	198	12	236
40	19	71	199	105	102	111	147
185	195	232	247	27	47	84	109
41	177	218	254	43	85	136	186
271	48	117	190	18	110	216	81
224	138	93	106	217	248	181	97

Multiple values may not be more than 5% to 10% of S-box size as checked in the experiment. This is only single flaw of *Pure Dynamic S-box* but this flaw cannot create backdoor for attacker because repeated values are replacing by other values. Depends on the values in the S-box create the following

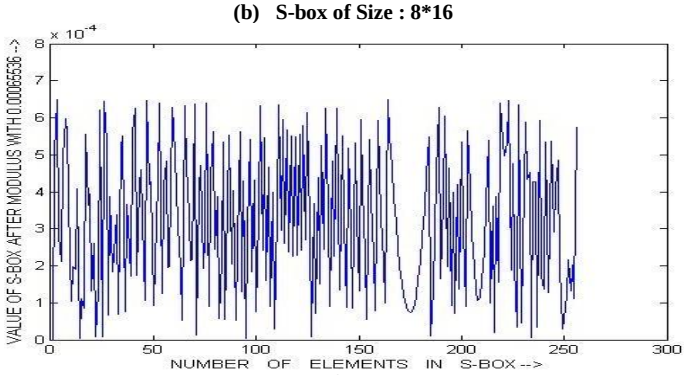
graphs to analysis the Non-Linearity and Strict Avalanche Criterion shown below:



(a) S-box of Size : 8*8



(b) S-box of Size : 8*16



(c) S-box of Size: 16*16

FIGURE 2: Non-linearity of S-box for different size

In Figure 2, that shows the graphs created using Algorithm 1 where use the same key division but *dsize(S-box size)* and *DMv (dependent modulo)* value are differ. Even for different combination of size and *DMv* values *Pure Dynamic S-box* satisfies the properties.

As we can see, *Pure Dynamic S-box* technique security is proportional to DMv value and S-box size. Increase the values of both, give overhead as shown in table:

TABLE III Security depends on MDv value

"MDv" Size \ Value of S-box	2^8	2^{12}	2^{16}	2^{20}	2^{24}
4*4	2^{12}	2^{16}	2^{20}	2^{24}	2^{28}
8*8	2^{14}	2^{18}	2^{22}	2^{26}	2^{30}
8*16	2^{15}	2^{19}	2^{23}	2^{27}	2^{31}
16*16	2^{16}	2^{20}	2^{24}	2^{28}	2^{32}

Coloured value shows the possible Brute force attack values on S-box. From the above table it is clear that how *MDv* value enhance the security. Furthermore, the size of the S-box also depends on the user or encryption algorithm. This property of Pure *dynamic S-box* increase the overhead for attacker. For example, if size is *16*16 and dependent modulo value* is 2^{20} then possible brute force value is 2^{28} increases overhead by 40% in terms of *DMv* value.

To make the inverse of S-box, here use the AVL tree implementation is best take less time and work efficiently. Every node in the AVL tree has two values one is S-box value and corresponding place in S-box. For example, S-box value 180 present on the place (1, 4) that represented as shown below:

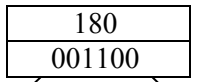


Figure 3: Node of AVL tree

Using "node and position" value pair per node in AVL tree gives the accurate result and take less time [17].

TABLE IV Comparison between Pure Dynamic and others S-box

S-box type In cipher	*Camellia 128 - Block Cipher*	*DES*	*AES*	*Blowfish*	*Pure Dynamic S-box*
Initial S-box type	Static	Static	Static	Static	Dynamic
Subsequent S-box	Static	Static	Static	Dynamic	Dynamic
Key input	No	No	No	Yes	Yes
Avalanche Effect	High	High	Good	Good	Good
Non-Linearity	Good	Fair	High	High	High
Multiple values	No	Yes	No	No	No
Size	16*16	4*16	16*16	16*16	dependent
Attacks Due To S-box	Cache_ timing attack, differential fault attack	DPA_ attack ,Timing Attack	Cache_ timing attack, Side_ Channel_ Attack	Vandenay Attack	Brute Force Attack

Above table [14][15][16] completely represent the advantage of *Pure Dynamic S-box* over existing S-box. Compare to blowfish that use initial S-box as Static may be backdoor for attacker, if initial S-box itself is dynamic which provide more security. This happened in *pure dynamic S-box* due to independent values given as input to it.

VII. CONCLUSIONS

In this paper, a key-dependent dynamic approach for generating *Pure Dynamic S-box* by using Logistic map is proposed. In the experiment have shown that how the key-dependent S-box that satisfies all the cryptographic properties of good S-box can enhance the security due to dynamic nature. Equation used to create S-box is Logistic map make more easily to cover the properties that provide high entropy. Total three unknown require to create the Pure *Dynamic S-box* that are key subparts, size of S-box and modulo value. More unknown require at encryption time produce more overhead to attacker to decryption. Since, number of *Pure Dynamic S-box* possible using proposed method, it is suitable to use in block cipher based dynamic S-box.

REFERENCES

[1] Przemysław Rodwald and Piotr Mroczkowski, "*How to create good s-boxes?*," *Mobile Robots, ICYR*, September 2006.

[2] Yuanqing Deng Hui Shi Jing Gong Tao Xie, "*Research on the Avalanche Property of the Camellia S-box"*, IEEE 1996, Workshop Record, 1996.

[3] Yanjun Li,Wenling Wu, Lei Zhang, and Liting Zhang, " *Integral Attacks on Feistel-SP Structure Block Cipher*",The authors - Published by Atlantis Press, 2013.

[4] Bart Preneel, "*Design principles for dedicated hash functions*", springer 1994.

[5] J.Kam,G. Davida, "*Structured Design of Substitution Permutation Encryption Networks",* IEEE Transactions on Computers, Vol 28, No. 10, 747, 1979.

[6] Fauzan Mirza, "*Linear and S-box pairs cryptanalysis of the Data Encryption Standard"*, Department of Computer Science Egham, Surrey TW20 0EX, England.

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

[8] Behnam Rahnama, Yunus Kıran, Raz Dara, "*Countering AES Static S-Box Attack*" ACM 2013.

[9] Razi Hosseinkhani, H. Haj Seyyed Javadi, "*Using Cipher Key to Generate Dynamic S-Box in AES Cipher System*", IJCSS, Volume (6): Issue (1):2012

[10]Evilcry, "*Blowfish Study n' Reverse*", RET ,22 July 2006.

[11] Sean Murphy1, M.J.B. Robshaw, "*Differential Cryptanalysis, Key-dependent S-boxes, and Twofish*", NIST, 15 May 2000.

[12]Ghada Zaibi, Abdennaceur Kachouri, "On Dynamic chaotic S-BOX", IEEE , 2009.

[13] ZHAO Xin-jie, WANG Tao, ZHENG Yuan-yuan, "*Cache Timing Attacks on Camellia Block Cipher*", IARC, 2009.

[14] Kazumaro, Tetsuya, Masayuki, Mitsuru, Shiho, Junjo, Toshio, "*Specification of Camellia – a 128-bit Block*", NTTC, MEC, July 12 2000.

[15] ZHAO Xin-jie, WANG Tao, "*An Improved Differential Fault Attack on Camellia*", IARC, 2009.

[16] Burkhard, Walt, *"AVL Dictionary Data Type Implementation"*, Advanced Data Structures. La Jolla: A.S. Soft Reserves, UC San Diego. p. 103, Spring 2012.