Searching for S-boxes with better Diffusion using Evolutionary Algorithm

Rahul Mishra[1], Bhupendra Singh[2], and Radhakrishnan Delhibabu[3]

[1] School of Computer Science and Engg.,Vellore Institute of Technology, Vellore
rahulmishrajs@gmail.com
[2] Centre for Artificial Intelligence and Robotics, DRDO, Bengaluru
bhupendra@cair.drdo.in
[3] School of Computer Science and Engg.,Vellore Institute of Technology, Vellore
r.delhibabu@vit.ac.in

Abstract. Over the years, a large number of attacks have been proposed against substitution boxes used in symmetric ciphers such as differential attacks, linear attacks, algebraic attacks, etc. In the Advanced Encryption Standard (AES) Block cipher, the substitution box is the only nonlinear component and thus it holds the weight of the cipher. This basically means that if an attacker is able to mount a successful attack on the substitution box of AES, the cipher is compromised. This research work aims to provide a solution for increasing cryptographic immunity of S-boxes against such attacks. A genetic algorithm based approach has been proposed to search for 8×8 balanced and bijective S-boxes that exhibit values of differential branch number, non-linearity, differential uniformity, count and length of cycles present and distance from strict avalanche criterion that are similar to or better than the AES S-box. An S-Box evaluation tool is also implemented to evaluate any S-boxes generated. S-box of AES is resistant to the crypt-analytic attacks. S-boxes constructed by the proposed algorithm have better cryptographic properties so they are also resistant to the crypt-analytic attacks. The strict avalanche criterion[11], which is based on completeness[22] and diffusion[5], is an essential property for any 8×8 S-box. Good diffusion means that a small change in the plaintext may influence the complete block after a small number of rounds. Therefore, a lower DSAC value is desirable to prevent vulnerabilities to attacks such as differential attacks. The DSAC is therefore used as the primary fitness criterion in this research work to search for S-boxes with better diffusion.

Keywords: Affine transformation · Confusion · Coefficient matrix · Diffusion · Distance from strict avalanche criterion · Evolutionary algorithm · Field characteristic · Galois Field.

1 Introduction

The substitution box is a vectorial Boolean function $S : GF(2^m) \mapsto GF(2^n)$. It is a basic component of symmetric ciphers. The primary purpose served by a Substitution box is to obscure the relationship between the key and the ciphertext in the block ciphers (Shannon's property of confusion)[5]. In general, an S-box takes some input bits, m, and transforms them into some number of output bits, n, where n is not necessarily equal to m.

The S-box proposed by Kaisa Nyberg[10] and used in the Rijndael cipher[7] is what the Advanced Encryption Standard is based on.The AES S-box maps an 8-bit input, x, to an 8-bit output, s = S(x). Both the input and output are interpreted as polynomials over GF(2). In the first step the input is mapped to its multiplicative inverse in $GF(2^8) = GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$, Rijndael's finite field. Zero, which has no inverse, is mapped to zero. The multiplicative inverse is then transformed using the affine transformation shown in equation 1.

$$\begin{bmatrix} s_7 \\ s_6 \\ s_5 \\ s_4 \\ s_3 \\ s_2 \\ s_1 \\ s_0 \end{bmatrix} = \begin{bmatrix} 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \\ 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \end{bmatrix} \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{1}$$

where $[s_7, ..., s_0]$ is the S-box output and $[a_7, ..., a_0]$ is the multiplicative inverse as a vector.

Many researchers have proposed methods to find S-boxes with better cryptographic properties than the S-box used in AES. One approach that is followed, is to make changes in the elementary steps of the substitution box used in Rijndael to arrive at new S-box. Liu[2], Cui[1], Gray[9], Hussain[12] have used this approach. Another approach is an exhaustive search using a heuristic based technique as in the case of Millan[17,16], Burnett[18], Clark[19] and Ivanov[20].

In this paper a Genetic Algorithm based approach has been proposed to generate new invertible coefficient matrix used in affine transformation step (as shown in Equation 1) of subtituion box that is used in Rijndael cipher, thus arriving at new S-box having better cryptographic properties and hence better security.

In the next section of the paper we describe the cryptographic properties of S-boxes that are used to evaluate its cryptographic strength. Section 3 describes the proposed genetic algorithm based approach. In section 4, results of the experiments are discussed. Finally section 5 concludes the paper.

2 Cryptographic properties of S-boxes

There are many properties that can be used to determine the strength of the S-box. Shannon[5] introduced the concepts of confusion and diffusion and proposed that every good cipher must hold these properties.

Confusion aims to obscure the relationship between the ciphertext and the key by ensuring that each binary digit (bit) of the ciphertext depends on multiple parts of the key.

Diffusion aims to increase the consequence of a single bit change of the plaintext in the ciphertext and that of a single bit change of ciphertext in the plaintext. Statistically, if one bit of the plaintext is changed it should result in half of the bits in the ciphertext getting changed and the same should hold true for the converse.

Following properties are considered to evaluate the strength of the S-box in this paper. An S-box evaluation tool has been implemented in C language. The tool can be used to check these cryptographic properties for any S-box:-

2.1 Bijectivity and Balancedness

These are the most important properties for a cryptographically good S-box and are checked in all the discovered S-boxes.

It is essential for the S-box to be bijective. The arithmetic of the 8 × 8 S-box occurs in the finite field of $GF(2^8)$. If the S-box is not bijective, it has poor cyclic structure and is vulnerable to attacks.

It is also necessary for the S-box to be constituted of a set of balanced Boolean functions, i.e., Boolean functions with equal number of zeroes and ones as output. The AES S-box and all the S-boxes proposed in this paper are bijective and balanced.

2.2 Strict Avalanche Criterion (SAC)

A cryptographic function satisfies the strict avalanche criterion if each output bit changes with the probability of half whenever a single input bit is complemented. The precise definition of Strict Avalanche Criterion as given in[11] is as follows: "consider X and X_i, two n-bit, binary plaintext vectors, such that X and X_i differ only in bit i, $1 \leq i \leq n$. Let

$$V_i = Y \oplus Y_i \tag{2}$$

where $Y = f(x)$ and $Y_i = f(x_i)$ and f is the cryptographic transformation under consideration. If f is to meet the strict avalanche criterion, the probability that each bit in V_i is equal to 1 should be one half over the set of all possible plaintext vectors X and X_i. This should be true for all values of i." Here, $'\oplus'$ is the Exclusive OR operation, or addition Modulo 2.

The distance from strict avalanche criterion (DSAC)[11] for a cryptographic function is calculated as the absolute difference of (i) the probability of inversion of an output bit for all input bits where X assumes all possible input values to the function and (ii) 2^{n-1}.

Therefore, the DSAC is calculated from the strict avalanche criterion table as

$$X_i = 2^i, \; for \; i = \{0, 1, ..., n-1\}, Y = \{y_0, y_1, \dots, y_{n-1}\}$$

$$DSAC = \sum_{i=0}^{n-1} |2^{n-1} - SAC[X_i][Y]| \quad (3)$$

The closer the values are to 2^{n-1} (which is 128 for an 8×8 S-box) the better the S-box satisfies the property of diffusion. The ideal DSAC value is 0 [25]. For the AES S-box, the SAC value for each bit (y_i, where $i = \{0, 1, 2, 3, 4, 5, 6, 7\}$ and corresponding X_i values are $\{1, 2, 4, 8, 16, 32, 64, 128\}$) is as follows:

Table 1. SAC Table for AES S-box

SAC	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_0	132	132	116	144	116	124	116	128
x_1	120	124	144	128	124	116	128	136
x_2	132	132	128	120	144	128	136	128
x_3	136	136	120	116	128	136	128	140
x_4	116	128	116	132	128	128	140	136
x_5	116	132	132	120	120	140	136	136
x_6	136	136	120	132	120	136	136	124
x_7	132	144	132	136	124	136	124	132

The DSAC value for AES S-box, as calculated from Equation 3, by using the values for $SAC[X_i][Y]$ from Table 1, comes out to be 432.

The results of our experiments show that the proposed genetic algorithm based approach arrives at many S-boxes having DSAC value lower than 432. The strict avalanche criterion[11], which is based on completeness[22] and diffusion[5], is an essential property for any 8×8 S-box. A good value of DSAC is desirable for good diffusion and avalanche effect. It means that for an S-box with good diffusion, a small change in the plaintext may influence the complete block after a small number of rounds thereby helping prevent vulnerabilities to attacks such as differential attacks.

2.3 Differential Cryptanalysis properties

The differential cryptanalysis properties are calculated from the difference distribution table T[14,23,24]. Each entry $T(\alpha, \beta)$ shows the number of times following equation was satisfied on varying x from 0 to 255.

$$S(x) \oplus S(x \oplus \alpha) = \beta, \tag{4}$$

where $S : GF(2^8) \mapsto GF(2^8)$ and $x, \alpha, \beta \in GF(2)^8$.

Non-zero linear structure If for some value of $\alpha \in GF(2)^8$, $S(x) \oplus S(x \oplus \alpha)$ is always same for all inputs $x \in GF(2)^8$, then α is termed as the linear structure[1] of the S-box. An S-box should never have a non-zero linear structure. The AES S-box has no non-zero linear structure.

Differential uniformity Assume $F(x) = (f_0(x), ..., f_{n-1}(x))$ from $GF(2)^8$ to $GF(2)^8$ is a multiple output Boolean function, the differential uniformity[1] is denoted by $\delta(F)$ and is defined to be

$$\delta(F) = Max|\{x|F(x) \oplus F(x \oplus \alpha) = \beta\}|(\alpha, \beta \in GF(2)^8, \alpha \neq 0) \tag{5}$$

The differential uniformity value for AES comes out to be 4. For all S-boxes obtained as a result of our genetic algorithm based approach the value for $\delta(F)$ comes out to be 4.

Differential branch number The differential branch number[13] has a value of 2 for AES. It is calculated as:

$$\beta_d(S) := \forall x, x' \in F_{2^8}, x \neq x', Min\{Wt(x \oplus x') + Wt(S(x) + S(x'))\} \tag{6}$$

Most of the proposed S-box design algorithms and optimization algorithms use the differential branch number as the property for the fitness function. The higher the branch number, the higher is its resistance to differential attacks. However the probability of finding S-boxes with high branch number reduces drastically as the number of input and output bits increase. According to Rui-sancchez [15], the probability of finding an 8×8 S-box with branch number greater than or equal to 3 is $6.7 * 10^{-15}$. All results obtained have a branch number value 2, which is the same as that of the S-box used in AES.

2.4 Linear cryptanalysis properties

The linear cryptanalysis properties revolve around how often:

$$\alpha.x = \beta.S(x), \forall x, \alpha, \beta \in GF(2)^8 \tag{7}$$

where $S : GF(2^8) \mapsto GF(2^8)$ is a multiple output Boolean function and '.' is the scalar product of the two vectors. A good S-box satisfies the above equation for nearly half the values of x.

Nonlinearity The linearity is denoted by L(F) and is defined to be:

$$L(S) = Max\{|LAT(a,b)|\} \tag{8}$$

where $a, b \in GF(2)^8$ and $LAT(a,b)$ is the linear approximation table[8,14]. The Nonlinearity[3,4] value for an S-box is calculated as:

$$NL(S) = 2^{n-1} - |L(S)/2| \tag{9}$$

The Nonlinearity for all the results of this research work have a value of 112 which is the same as that of the AES S-box.

Linear propagation The Linear propagation[21] is calculated as:

$$LP_{max}(f) = \forall a, b \neq 0, Max(2 * Pr[X.a = S(X).b] - 1)^2 \tag{10}$$

where $a, b, X \in GF(2)^8$ and The linear propagation value for AES comes out to be $2^{-6} = 0.015625$. The properties concerning immunity to linear attacks for AES have fairly optimal values. In fact, it would be necessary to have nearly 2^{62} known plain-texts and cipher-texts to be able to mount a successful Linear Cryptanalytic attack on AES.
In all the S-boxes generated using the modified genetic algorithm this value is successfully achieved.

2.5 Fixed and Inverse fixed points

An input vector x, for which $x \oplus S(x) = (00)_{16}$ is termed as a fixed point[6] for the S-box. If, $x \oplus S(x) = (FF)_{16}$, then x is termed as an inverse fixed point[6]. The AES S-box has no fixed points or inverse fixed points. An S-box should have very few, if any, fixed or inverse fixed points. The discovered S-boxes have either zero or very few fixed or inverse fixed points. If an 8×8 S-box has 2^8 , i.e., all its points as fixed points (Identity S-box), then the key can be retrieved using only one known plaintext-ciphertext pair.

2.6 Cycles in the S-box

The cyclic structure contained within the S-box increases its predictability. They are calculated recursively as $(x, S(x), S(S(x)), ...S(S(...S(x)), x)$ where $S : GF(2^8) \mapsto GF(2^8)$ is the S-box function. The cryptographically good 8×8 S-box should have 1 cycle of length 2^8 where n is the number of output bits.

3 The Proposed Genetic Algorithm approach for finding good S-boxes

The System flow diagram of the proposed Genetic Algorithm is shown in Figure 1

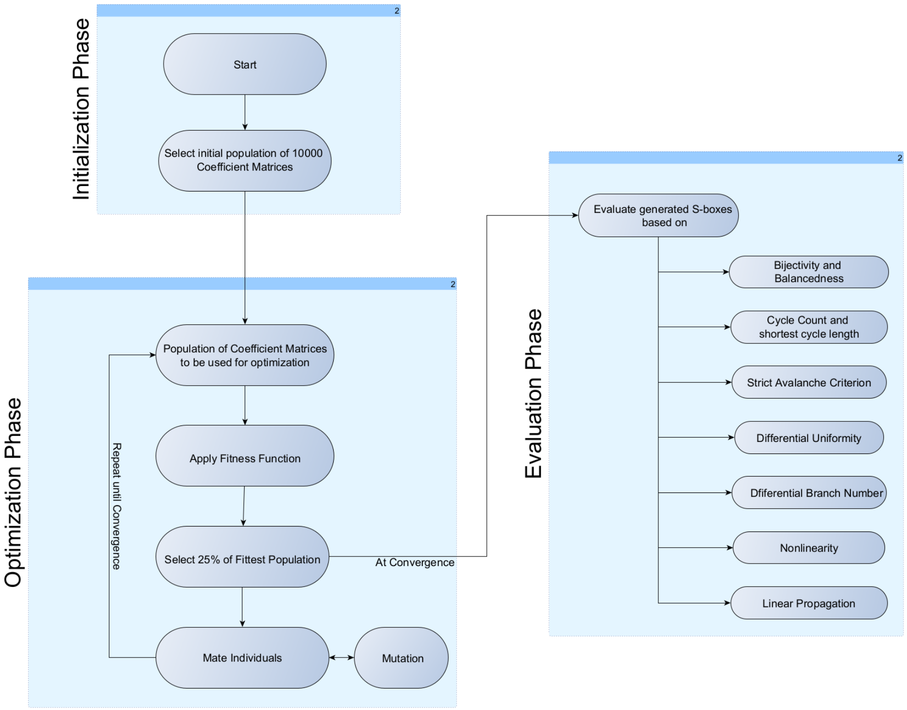
Initialization Phase
2
Start
Select initial population of 10000 Coefficient Matrices
Optimization Phase
2
Population of Coefficient Matrices to be used for optimization
Repeat until Convergence
Apply Fitness Function
Select 25% of Fittest Population
At Convergence
Mate Individuals
Mutation
Evaluation Phase
2
Evaluate generated S-boxes based on
Bijectivity and Balancedness
Cycle Count and shortest cycle length
Strict Avalanche Criterion
Differential Uniformity
Dfiferential Branch Number
Nonlinearity
Linear Propagation

Fig. 1. System flow Diagram for the proposed Genetic algorithm approach

The experiment has been conducted for 8×8 S-boxes. The key elements of the proposed Genetic algorithm are:

Gene: A gene (smallest variable units which combine to form a possible solution to the optimization problem otherwise called a chromosome) is one row of the coefficient matrix in the affine transformation step of AES (Equation 1). It can assume a value in $GF(2)^8$.

Chromosome: A chromosome (combination of genes) is a complete 8×8 invertible coefficient matrix (Equation 1) that can be used to generate an S-box permutation.

Fitness Function: The fitness function uses minimum length of cycles, number of cycles, branch number, differential uniformity, nonlinearity and DSAC to calculate the fitness score. To ensure that S-boxes generated have properties as good as the AES S-box, we first ascertain that generated S-boxes have values of branch number, differential uniformity, nonlinearity as at least 2, 4 and 112 respectively. Then we compare the obtained DSAC value with the target DSAC value which is taken as 384. This target value was calculated by keeping maximum allowed deviation from optimum SAC value as 6 (i.e. $|2^{n-1} - SAC[X_i][Y]| \leq 6$ in Equation 3). The equation for the fitness function that is used is described below:

```
Initialize Fitness = 0
if (Sbox.isBijective() and fixed points < 2
    and inverse fixed points < 2){
        if (Sbox.differential uniformity >= 4 and
        Sbox.nonlinearity >= 112 and Sbox.branch number >= 2){
                Fitness += Sbox.DSAC - Target DSAC
        }
}
```

(11)

If fitness is less than zero for an individual in any generation, the algorithm converges. The S-box generated with an individual with fitness less than zero found using the above equation will be:
a) Bijective
b) Have very few or no fixed or inverse fixed points
c) Have differential uniformity value as 4 or higher
d) Have differential branch number as 2 or higher
e) Have nonlinearity as 112 or higher
f) Have DSAC definitely lower than target value 384.

The proposed Genetic algorithm is given below:

Algorithm

Step 1: Choose initial population: The population size is fixed at 10000. The Individuals in the initial population are a set of coefficient matrices in which each element of the matrix is randomly selected from $GF(2)^8$. Calculate fitness score of all individuals using equation 11.

Step 2: Selection:

2a. Select top 25% of the population. Discarding the bottom 75% allows us to search through the search space quicker.

2b. Directly carry forward top 5% of the population based upon its fitness score to the next generation (Elitism). This allows the S-box output permutations with highest fitness to proceed to the next generation without modification.

2c. To ensure that the algorithm quickly converges, generate rest of the population from the remaining top 20% population as follows:

(i) Out of this top 20%, two parents are selected at random and mated as per step 3 to generate a new individual.

(ii) Repeat (i) till the remaining 95% of the population is generated. The reason for selecting small percentage of individuals with highest fitness and then randomly choosing a parent is to avoid local optima.

Step 3: Crossover & Mutation: Each row (gene) of the offspring is decided based on the following rule:

3a. Generate a random number between 0 and 100 for each row i (gene i).

3b. Offspring inherits row i from parent 1 (parent 2) if random number is between 0 and 40 (41 to 80) otherwise row i is obtained using mutation (random value in $GF(2)^8$).

This ensures that the algorithm does not get stuck at a point of no solution even if there is one.

Step 4: Convergence: Calculate fitness score of the offsprings. Stop if calculated fitness score (Sbox.DSAC - TargetDSAC) for an offspring is less than zero else repeat step 2 and 3.

The detailed diagram for the proposed approach is given below:

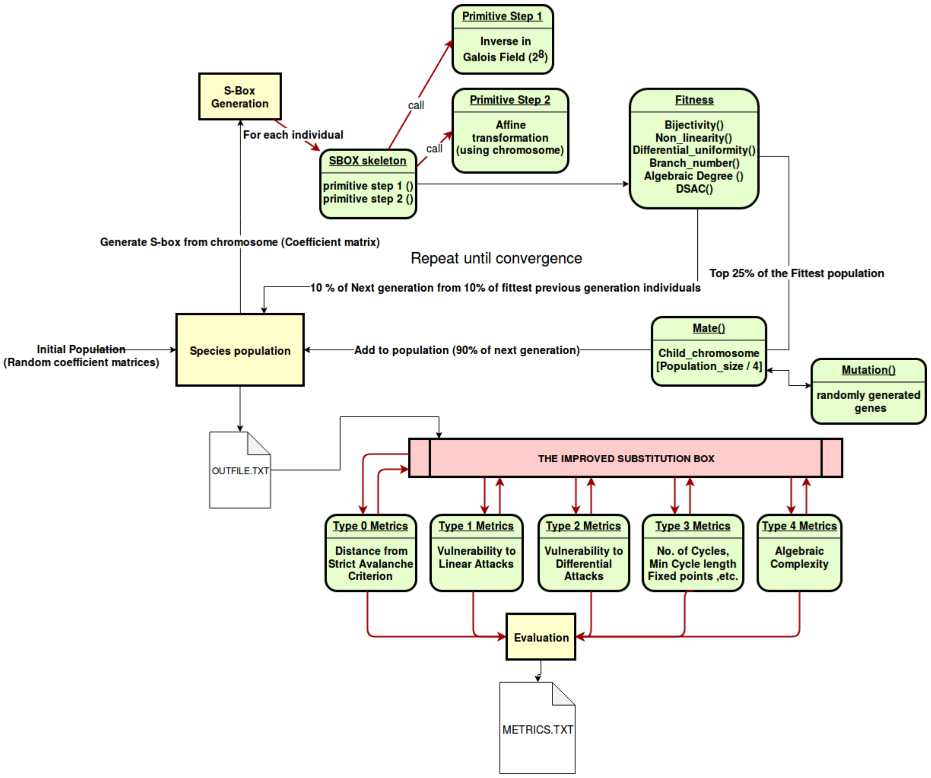
Primitive Step 1
Inverse in Galois Field (2^8)
S-Box Generation
call
Primitive Step 2
Affine transformation (using chromosome)
Fitness
Bijectivity()
Non_linearity()
Differential_uniformity()
Branch_number()
Algebraic Degree ()
DSAC()
For each individual
call
SBOX skeleton
primitive step 1 ()
primitive step 2 ()
Generate S-box from chromosome (Coefficient matrix)
Repeat until convergence
Top 25% of the Fittest population
10 % of Next generation from 10% of fittest previous generation individuals
Mate()
Child_chromosome [Population_size / 4]
Initial Population (Random coefficient matrices)
Species population
Add to population (90% of next generation)
Mutation()
randomly generated genes
OUTFILE.TXT
THE IMPROVED SUBSTITUTION BOX
Type 0 Metrics
Distance from Strict Avalanche Criterion
Type 1 Metrics
Vulnerability to Linear Attacks
Type 2 Metrics
Vulnerability to Differential Attacks
Type 3 Metrics
No. of Cycles, Min Cycle length Fixed points ,etc.
Type 4 Metrics
Algebraic Complexity
Evaluation
METRICS.TXT

Fig. 2. Detailed Diagram for the proposed Genetic algorithm approach

4 Performance Analysis of the S-boxes generated

The S-boxes with improved DSAC generated by the proposed GA based algorithm are denoted as I1, I2 and I3.

These are compared with:

a) The AES S-box [6]
b) The S-box proposed by Wang [27]
c) The S-box proposed by Khan [26]
d) The S-box proposed by Liu [2]
e) The Affine-Power-Affine S-box [1]
f) The S-box proposed by Alhadawi [28]

Table shows the comparison of these S-boxes.

Table 2. Comparison of performance of the obtained S-boxes with AES S-box, Khan [26], Wang [27], Liu[2] ,APA[1] and Alhadawi [28]

Cryptographic property	**Performance**								
	AES S-box	**Liu**	**APA S-box**	**Khan**	**Wang**	**Alhadawi**	**I1**	**I2**	**I3**
Bijective	yes	yes	yes	yes	yes	yes	yes	yes	yes
No. of Cycles	5	5	1	2	6	5	7	3	1
Minimum Cycle length	2	2	256	68	5	1	4	5	256
Distance to SAC	432	408	372	428	380	536	352	364	380
Non zero linear structure	none	none	none	none	none	none	none	none	none
Differential uniformity	4	4	4	6	10	10	4	4	4
Branch number	2	2	2	2	2	2	2	2	2
Non-linearity	112	112	112	108	94	94	112	112	112
Linear propagation	0.015625	0.015625	0.015625	0.024414	0.070557	0.070557	0.015625	0.015625	0.015625

From the table it is clear that all the S-boxes found in this research work have exactly same values for differential uniformity, branch number, Nonlinearity and linear propogation as that of the AES S-box. In addition, all S-boxes obtained are bijective, balanced, have no non-zero linear structure and have very few or no fixed and inverse fixed points. The property of DSAC, however, shows an improvement when compared to the AES S-box. All S-boxes have DSAC lower than the target value of 384 as described in Section 3. These values are significantly lower than the DSAC of AES which is 432. The DSAC values obtained in this research work are 352 (I1), 364 (I2) and 380(I3). A lower DSAC results in better diffusion and therefore increases cryptographic strength.

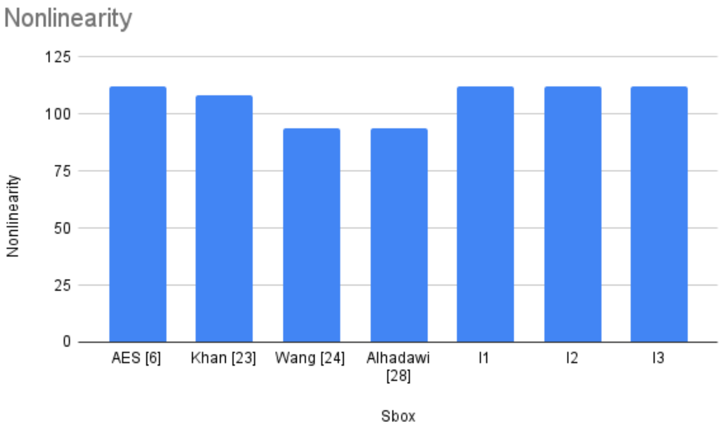

Fig. 3. Comparison of the Nonlinearity of AES[6], Khan[26], Wang[27], Alhadawi[28], I1, I2 and I3

When compared to other recently proposed S-boxes like the one by Khan [26] which is constructed using Gaussian Distribution or the one by Wang [27] which also uses Genetic Algorithm approach, it is clear that the S-box proposed in our paper shows better value of Nonlinearity (As shown in figure 3). The Nonlinearity value for the S-box proposed by Alhadawi [28] which is constructed using discrete chaotic maps and cuckoo search algorithm also falls short of the value that is achieved using our proposed method. The Nonlinearity value achieved through our proposed approach is same as that of AES S-box.

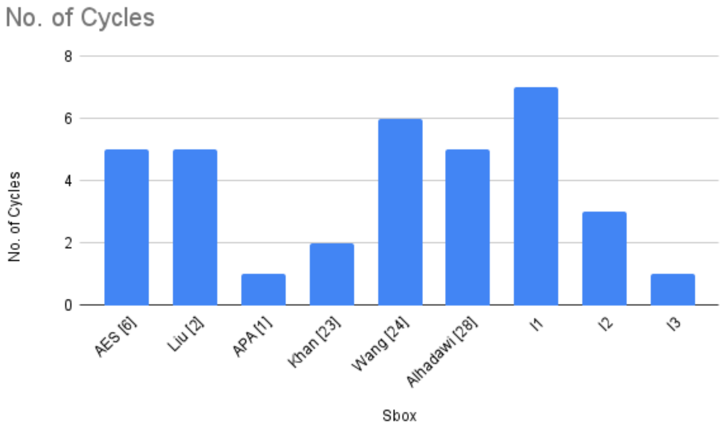

Fig. 4. Comparison of the number of cycles of AES[6], Liu[2], APA[1], Khan[26], Wang[27], Alhadawi[28], I1, I2 and I3

The figure 4 shows how I2 and I3 have very few cycles. Additionally, the proposed S-boxes also have greater length for smallest cycle when compared with AES. For an optimal S-box there should be one cycle of length 2^8. This was achieved by Cui [1] in his APA S-box. All other S-boxes used for comparison in this paper have more than one cycle. Out of the three S-boxes proposed in this paper, one S-box I3 is able to achieve the optimum value for cycles (1 cycle with length 256) and the other two perform extremely well with very few number of cycles.

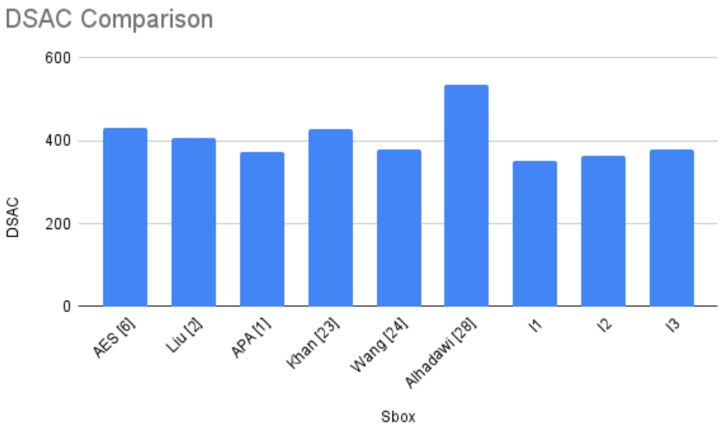

Fig. 5. Comparison of the DSAC values of AES[6], Liu[2], APA[1], Khan[26], Wang[27], Alhadawi[28], I1, I2 and I3

The DSAC value for the improved S-boxes comes out to be significantly lower than the AES S-box. The DSAC values of all proposed S-boxes are lower than that of the one proposed in Khan[26] and Alhadawi[28]. The proposed S-boxes, I1 and I2, also outperform the S-box proposed by Wang[27]. This reduces the possibility of known plain-text attacks and differential attacks. The approach taken in this paper differs from the existing literature in one primary aspect - The individual in the evolutionary process is not an S-box permutation (as in the case of [27]) but rather the coefficient matrix for Affine transformation step (Equation 1) for constructing S-boxes. The algorithm converged within 6 hours in the case of each of the three S-boxes discovered. This can be attributed to the modification in the genetic algorithm (i.e. the use of coefficient matrix instead of S-box permutation) which allows the algorithm to take larger leaps through the Search space and arrive at a solution quicker.

5 Conclusions and Scope of Future Work

Rijndael cipher accepted as the AES in 2001 is a well proven and an efficient cipher. The S-box forms the backbone of this cipher. It exhibits near perfect resistance to Linear and Differential attacks. This research work studies the construction of the AES S-box and explores the possibility of finding S-boxes with similar or better cryptographic properties using a genetic algorithm based approach applied to one of its elementary steps - The affine transformation. The discovered S-boxes have also been analyzed. The tool for measuring and optimizing the S-box properties has been successfully tested on multiple S-boxes with varying sizes and complexities. The discovered improved S-boxes exhibit

lower value for DSAC when compared with the existing AES substitution box. A DSAC lower than that of AES S-box implies that the discovered S-boxes show better diffusion and are therefore less prone to known plaintext and differential attacks.
Future research work can include:
a) Increasing the initial population size from 10000
b) Selecting a lower percentage of individuals for elitism and mutation step
c) Decreasing the maximum permissible deviation from optimum DSAC value (which is kept as ’6’ for this experiment)
and analyzing the effects of varying these parameters on the performance of the algorithm by assessing the properties of the S-boxes received at convergence and the time taken until convergence.

6 Declarations

Conflict of Interest: The authors declare that they have no confict of interest. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Declaration of Data Availability: The algorithm and its data inputs are detailed in section 3. No additional datasets were used to obtain the resultant S-boxes.

References

1. Cui, J., Huang, L., Zhong, H., Chang, C., and Yang, W. (2011). An improved AES S-Box and its performance analysis. International Journal of Innovative Computing, Information and Control , 7 (5), 2291-2302.v
2. Liu, J., Wei, B., Cheng, X., and Wang, X. (2005, March). An AES S-box to increase complexity and cryptographic analysis. In Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on (Vol. 1, pp.724-728). IEEE
3. Rodinko, M., Oliynykov, R., and Gorbenko, Y. (2016, October). Improvement of the high nonlinear S-boxes generation method. In 2016 Third International Scientific-Practical Conference Problems of Info communications Science and Technology (PIC S and T) (pp. 63-66). IEEE.
4. Kazymyrov, O., Kazymyrova, V., and Oliynykov, R. (2013). A Method For Generation Of High-Nonlinear S-Boxes Based On Gradient Descent. IACR Cryptology ePrint Archive , 2013 ,578.
5. Shannon, C. E. (1949). Communication theory of secrecy systems. Bell system technical journal, 28(4), 656-715.
6. Vincent Rijmen and Joan Daemen, The Design of Rijndael, ISBN 978-3-662-04722-4, Springer
7. Daemen, J., and Rijmen, V. (1999, March). The Rijndael block cipher: AES proposal. In First candidate conference (AeS1) (pp. 343-348).

8. Y. Crama, and P. Hammer (2010). Vectorial Boolean Functions for Cryptography. In Boolean Models and Methods in Mathematics, Computer Science, and Engineering,Y. Crama, and P. Hammer, Ed.Cambridge: Cambridge University Press, 2010, pp. 398–469
9. Tran, M. T., Bui, D. K., and Duong, A. D. (2008, December). Gray S-box for advanced encryption standard. In 2008 International Conference on Computational Intelligence and Security (Vol. 1, pp. 253-258). IEEE.
10. Nyberg, K. (1991, April). Perfect nonlinear S-boxes. In Workshop on the Theory and Application of of Cryptographic Techniques (pp. 378-386). Springer, Berlin, Heidelberg
11. Webster, A. F., and Tavares, S. E. (1985, August). On the design of S-boxes. In Conference on the theory and application of cryptographic techniques (pp. 523-534). Springer, Berlin, Heidelberg.
12. Hussain, I., Shah, T., Gondal, M. A., and Khan, W. A. (2011). Construction of cryptographically strong 8x8 S-boxes. World Applied Sciences Journal , 13 (11), 2389-2395.
13. Sarkar, S., and Syed, H. (2018, July). Bounds on Differential and Linear Branch Number of Permutations. In Australasian Conference on Information Security and Privacy (pp. 207-224). Springer, Cham.
14. Anne Canteaut(2015), Cryptographic S-Boxes [PowerPoint slides]. Retrieved from Cryptographic S-boxes.
15. Ruisanchez, C. P. (2015). A new algorithm to construct S-Boxes with high diffusion. International Journal of Soft Computing, Mathematics and Control (IJSCMC) , 4 (3).
16. Millan, W., Burnett, L., Carter, G., Clark, A., and Dawson, E. (1999, November). Evolutionary heuristics for finding cryptographically strong S-boxes. In International Conference on Information and Communications Security (pp. 263-274).Springer, Berlin, Heidelberg.
17. Millan, W., Clark, A., and Dawson, E. (1998, May). Heuristic design of cryptographically strong balanced Boolean functions. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 489-499). Springer, Berlin, Heidelberg.
18. Burnett, L. D. (2005). Heuristic optimization of Boolean functions and substitution boxes for cryptography (Doctoral dissertation, Queensland University of Technology).
19. Clark, J. A., Jacob, J. L., and Stepney, S. (2005). The design of S-boxes by simulated annealing. New Generation Computing ,23 (3), 219-231.
20. Ivanov, G., Nikolov, N., and Nikova, S. (2015, September). Cryptographically strong S-boxes generated by modified immune algorithm. In International Conference on Cryptography and Information Security in the Balkans (pp.31-42). Springer, Cham.
21. Jacques Stern and Serge Vaudenay (1998), CS-Cipher, Fast softwaare Encryption - FSE'98, LNCS 1372, (pp. 189-204) 1998, Springer-Verlag, Berlin, Heide lberg
22. Feistel, H. (1973). Cryptography and computer privacy. Scientific american, 228(5), 15-23.
23. Attaullah, Jamal, S.S. & Shah (2018). A Novel Algebraic Technique for the Construction of Strong Substitution Box. Wireless Pers Commun 99, 213–226 .
24. Nitaj, A., Susilo, W., & Tonien, J. (2020, November). A New Improved AES S-box with Enhanced Properties. In Australasian Conference on Information Security and Privacy (pp. 125-141). Springer, Cham.
25. Sokolov, A. V. (2013). Constructive method for the synthesis of nonlinear S-boxes satisfying the strict avalanche criterion. Radioelectronics and Communications Systems, 56(8), 415-423.

26. Khan, M. F., Ahmed, A., & Saleem, K. (2019). A novel cryptographic substitution box design using Gaussian distribution. IEEE Access, 7, 15999-16007.

27. Wang, Y., Zhang, Z., Zhang, L. Y., Feng, J., Gao, J., & Lei, P. (2020). A genetic algorithm for constructing bijective substitution boxes with high nonlinearity. Information Sciences, 523, 152-166.

28. Alhadawi, Hussam S., et al. "A novel method of S-box design based on discrete chaotic maps and cuckoo search algorithm." Multimedia Tools and Applications 80.5 (2021): 7333-7350.

Appendix 1.A

Here we present in hex notations, the bijctive S-boxes generated by the proposed modified genetic algorithm which are described in Table 1 and have significantly lower value of DSAC and thus better diffusion.

I1

$$
\begin{array}{cccccccccccccccc}
9c & 32 & 36 & fd & fe & f7 & 57 & bc & 67 & 89 & 3c & 69 & 35 & ba & 62 & 55 \\
18 & ed & 77 & 51 & 95 & 76 & bb & f2 & ce & d5 & 6a & c2 & 20 & c6 & f5 & 7f \\
a3 & 5a & 84 & c1 & cb & c3 & 3d & b4 & c7 & a5 & 3b & 91 & 9a & 1e & 04 & 23 \\
4a & b0 & 02 & 10 & 8b & 47 & 29 & 8c & 25 & ec & e1 & f4 & fc & a2 & 60 & 3a \\
e2 & 8e & a6 & 87 & e4 & 34 & 19 & 66 & 72 & 63 & d9 & 03 & 65 & 97 & c9 & 41 \\
11 & 16 & ea & 50 & 6d & 39 & 0f & 7a & 94 & 7b & ab & 6f & 13 & bf & 15 & db \\
75 & 5c & 01 & f6 & 1b & dc & 08 & 22 & b7 & fa & e6 & 5d & 7e & ac & 4f & 0d \\
1d & 09 & 74 & 45 & e7 & 43 & 0c & 31 & a7 & c0 & 12 & 0e & e0 & 56 & 9d & 79 \\
d7 & 21 & 8f & 33 & da & 24 & d4 & 2f & df & a9 & 3e & cf & 53 & d6 & e8 & 96 \\
f3 & 82 & d0 & be & 61 & 0a & eb & 8a & d8 & 3f & 07 & 40 & b2 & b9 & ee & 71 \\
f8 & 6b & 00 & 8d & 7c & 46 & cd & b5 & 73 & 1a & ad & ff & 88 & 28 & 58 & f1 \\
37 & 14 & a8 & 5b & 49 & b6 & b3 & 6e & 38 & d2 & 52 & 9f & 30 & 9e & a1 & 81 \\
0b & 92 & ae & aa & 2b & ca & cc & d3 & 93 & dd & 59 & 44 & 70 & c4 & e5 & 86 \\
f9 & a0 & af & 5f & 1f & 85 & 5e & 2d & 90 & a4 & b1 & 64 & 4d & 1c & 48 & 2c \\
9b & 99 & 80 & 83 & fb & 7d & 26 & 4b & ef & 27 & 2e & f0 & b8 & bd & 06 & d1 \\
2a & 05 & e9 & 42 & c8 & 54 & 78 & 98 & 17 & e3 & c5 & 4e & 6c & de & 68 & 4c
\end{array}
\tag{12}
$$

I2

$$\begin{array}{cccccccccccccccc}
63 & be & b0 & 77 & 40 & 3e & a4 & f8 & 1f & 66 & f9 & 58 & 87 & ea & 3a & b8 \\
6c & 57 & 3f & b6 & 1d & 14 & c1 & 1b & fb & b9 & 6f & a0 & 82 & ae & 22 & 6a \\
b4 & d4 & 16 & 97 & de & 8b & d2 & ad & 85 & a6 & c0 & 13 & 71 & 7e & 17 & b5 \\
f4 & a3 & 05 & 39 & 7a & 84 & fc & 43 & a7 & 7c & 0c & 09 & 5c & 9f & 26 & eb \\
3b & 5f & 91 & 21 & 29 & ac & 47 & 34 & 1a & 11 & e2 & 2e & 03 & 01 & c2 & 96 \\
12 & 2b & 6e & 9d & 56 & dc & 75 & 4f & 36 & 64 & e1 & 4a & 0e & cf & 1c & fe \\
23 & c6 & 32 & 15 & 5b & c7 & 4c & 9e & 9a & 4e & 35 & ed & 41 & d8 & d1 & 69 \\
49 & 67 & 08 & 98 & 1e & 8a & 42 & 89 & ba & bc & 25 & 5e & 27 & 8f & 48 & 78 \\
a5 & a9 & 74 & 95 & d5 & 8c & 92 & ee & f0 & fd & e5 & d0 & aa & 8e & 72 & 2a \\
30 & 04 & 9c & e4 & 0d & 50 & 45 & 51 & c9 & ce & 20 & bd & bf & dd & 60 & 2d \\
52 & 44 & 19 & 68 & 5d & af & cc & 86 & 31 & 70 & f3 & 6b & 4d & d7 & c8 & 2c \\
9b & 37 & d6 & ff & c3 & b1 & 94 & 61 & f7 & 80 & 81 & 54 & a2 & 7f & a8 & 33 \\
7b & 24 & c4 & ca & e0 & f5 & e7 & ab & 0f & ec & e3 & b3 & 06 & b2 & 02 & 0a \\
79 & 83 & ef & f1 & 55 & 3d & da & f2 & 38 & 8d & 88 & 28 & cd & 62 & e8 & d9 \\
5a & 46 & 18 & 2f & 65 & 76 & 90 & df & 4b & bb & c5 & 07 & f6 & d3 & 0b & b7 \\
cb & 3c & 59 & a1 & e9 & 93 & 53 & 6d & 00 & 10 & 99 & fa & 7d & db & 73 & e6
\end{array} \tag{13}$$

I3

$$\begin{array}{cccccccccccccccc}
63 & b3 & ca & 36 & 00 & d3 & 9f & 33 & c2 & 02 & 2f & 5e & fc & 69 & ae & bc \\
06 & 3b & cf & c5 & b0 & da & 7c & bf & 5d & a0 & 68 & e2 & 9d & 9b & f0 & 09 \\
b7 & 35 & a8 & d4 & 31 & f7 & 3a & f5 & 8e & ed & 60 & c9 & 39 & 5c & b4 & ab \\
38 & 8c & ee & c0 & 21 & 92 & 4e & 6e & f1 & 2e & 84 & e5 & 23 & a2 & 8d & 75 \\
b2 & 4d & db & 9e & e8 & e9 & 13 & d7 & a3 & bb & 1f & fb & e1 & 93 & 12 & c8 \\
d5 & 9a & 74 & d0 & 27 & 43 & 44 & 65 & a5 & 70 & 71 & 04 & f6 & 05 & ac & 3c \\
ec & 6f & d8 & c6 & 30 & 73 & 0b & be & c3 & 79 & cb & 7a & 1c & 3e & 54 & 67 \\
6a & 1e & f9 & b1 & de & eb & 72 & 85 & ce & c1 & e3 & 51 & 91 & 8a & 76 & 53 \\
83 & 88 & 58 & a6 & 29 & e4 & b5 & 14 & 45 & 52 & 0c & 48 & e6 & 96 & 57 & 86 \\
aa & f2 & cc & 10 & 98 & 28 & 61 & 34 & 0a & 19 & 82 & dd & af & 5f & 0d & 95 \\
5a & 7d & cd & 7b & 3f & 87 & 6b & e0 & b6 & 25 & 2b & 15 & 17 & 5b & 16 & 89 \\
df & b9 & 47 & 20 & 0e & d6 & ba & 11 & 56 & ef & f3 & 55 & 90 & 40 & 94 & c4 \\
3d & ff & 1d & 64 & 6d & 24 & 7e & fa & ea & 66 & 03 & a4 & 80 & b8 & fd & 8b \\
4f & 81 & 08 & 59 & 49 & bd & 4c & 37 & dc & f8 & 99 & f4 & 77 & 7f & 1b & 22 \\
2c & 0f & d1 & e7 & 6c & 2a & c7 & 2d & 18 & d2 & 01 & 9c & 4a & 26 & 97 & d9 \\
78 & a1 & 42 & fe & 07 & a9 & 46 & 1a & 8f & a7 & ad & 41 & 32 & 50 & 4b & 62
\end{array} \tag{14}$$