IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol. 8, Issue.10, October 2020 E-ISSN: 2347-2693

# Based on ABC Optimization effective Substitution-Boxes deployed using Chaotic mapping

**Manpreet Kaur[1*], Sarabjeet Kaur[2]**

[1]Maharaja Ranjit Singh Punjab Technical University Bathinda, Panjab India
[2]Adesh Institute of Engineering and Technology Faridkot, Panjab, India

**Corresponding Author: mpmanpreet71@gmail.com*

DOI: https://doi.org/10.26438/ijcse/v8i10.133140 | Available online at: www.ijcseonline.org

Received: 12/Oct/2020, Accepted: 20/Oct/2020, Published: 31/Oct/2020

***Abstract*** **-** Most block ciphers contain primitive substitution boxes to add required nonlinearity. S-boxes maintain high confusion and resistance to linear and differential attacks. The protection of those ciphers depends on the force of the S-boxes used during the replacement stage. It is difficult to create encrypted, strong S-boxes which fulfill various characteristics like high non-linearity, good avalanche effect, bit-independent requirements, low differential uniformity and linear probability, etc. We proposed in this paper to create an S-box based on optimization of artificial colony bee and chaotic diagram. An initial S-Box is built to customize the algorithm to meet several features. The results of the simulation and comparison with recent proposals suggest that the proposed ABC optimization algorithm performs fairly easily and creates an S-box with a higher degree of security.

***Keywords***- ABC optimization Substitution-box Chaotic Logistic map Block ciphers Security

## I. INTRODUCTION

Recent developments in the field of telecommunications have led to a shorter stream of huge data across open networks. Contemporary security systems that can enable secure information transmission have always been an open question. In order to secure the sharing of secluded information between trusted parties, the cryptographic primitive elements are used [1]. Modern cryptogram block cyphers have been developed to fulfil the widespread demand for safe communications in today's technologically well equipped era (3). The technology of block chiphery has played an important role for secure storage and transmission for many years now. A powerful chip complies with the Shannon confusion and diffusion requirement as well as its intrinsic capacity to mitigate traditional and other forms of cryptographic attacks [3]. In order to guarantee uncertainty, many modern ciphers such as Concept, DES, AES, Blowfish, GOST etc. are using substitution boxes. They are only block chip components that cause nonlinearity and enhance cipher security [4]. The construction of strong, vibrant S-boxes has therefore become an important research topic for security experts [5]. The n X m replacement box is mathematically a nonlinear mapping of S from GF(2n) to GF(2 m), where n C m. Its fundamental role is to transform an n-bit input into an m-bit output in a nonlinear fashion. The Boolean multi-output S: fm(x) fm-1(x) ..... f1(x) is a Boolean part of each Fi(x) that has been specified from GF(2n) to GF(2) as a boolean scale multi-output function (n-variable). The S-boxes are typically displayed and implemented as search tables. Cryptographers were attracted by the development of efficient S-boxes, notably in size 8 X 8, due to the success of AES cipher blocks, that used 8 X 8 S-box [6 , 7].

Many design techniques for mainly 8 X 8 S boxes with strong cryptographically characteristics have been studied and used. Techniques include: group theory, chaos (random), cellular automatic design, optimization dependent (heuristics). The most basic application and investigation is made for extremely small numbers of techniques of optimization such as TLBO[8], ACO[9], GA[10–12] and simulated annealing [13].

The key contribution of the algorithm proposed is constructing an 8 9 8 replacement box with the use of optimization of artificial colonies and chaotic maps. To create an initial S-box, the chaotic logistic map is used. In order to obtain a significant S-box configuration with several upright cryptographic properties, the proposed algorithm optimizes the first S-box. The entire building process for the S-box is secretly regulated. The detailed comparison of the suggested S-box with the newly examined S-boxes indicates a superlative success in cryptography. The optimized S-box will act as a happy nonlinear feature to reinforce the block ciphers.

The rest of this correspondence is as follows. The 1D chaotic map model is shown in Sect. 2. In Sect. In Sect. 3 briefly presented the quest technique for optimizing the artificial bee colony. The S-box construction algorithm proposed for optimization is explained and presented in Sect. 4. Performance analysis and comparison with the new S-Box construction system based on optimization and chaos was performed in Sect. 5, accompanied by work performed in Sect. 6.

 

## II. LOGISTIC MAPS

The 1D chaotic Logistic maps is simplest and most studied and applied map. It is a polynomial mapping of degree 2 which govern as

$$x_{n+1} = F(x_n, \lambda) = x_n \times \lambda \times (1 - x_n) \quad (1)$$

where $x_{n?1}$ is a state variable which is bounded in range (0, 1) for all n C 0, n is the number of iterations, and k is system parameter for function F(x, k). It is a simple mapping from (0, 1) to (0, 1) but exhibiting complicated dynamics as introduced by R. M. May [17]. The dynamics of map is extremely sensitive to its seed value $x_0$ and parameter k. Theresearch have ascertained that the Logistic map shows chaotic phenomenon only when 3.57 \ k B 4. The values of k extremely close to 4 have shown excellent chaotic dynamics. A random sequence of floating point values can be obtained by sampling chaotic variable x in each iteration. The iteration of F(x, k) in 15-digit precision of computation is performed to get good random sequence. We engaged the chaotic Logistic map as a key-dependent and deterministic source to yield initial 8 X 8 random S-box.

## III. ABC OPTIMIZATION

Artificial bee colony optimization, introduced by Karaboga [18] and then modeled by Tereshko [19], is swarm intelligence based meta-heuristic approach which mimics the collective intelligent foraging behavior of honey bees.

Due to fewer control parameters, ABC optimization found to exhibit better performance to most of the other population-based techniques [20]. The ABC optimization has been applied to solve a number of problems such as transportation problems, weighted satisfiability problem, data mining problems, routing problem, parameter identification problems, clustering, information hidings, etc., [21, 22].

The model developed by Tereshko has three main components: employed bees, unemployed bees and food sources. Firstly, a number of food sources positions are randomly decided and their nectar amount is evaluated. Each processing step in ABC optimization employs the onlooker bees to their food sources and finds fitness amount of food sources, selection of food sources to generate possible new solution that is judged against the constraints for its suitability and selection as near optimal solution. The mathematical modeling of ABC based optimization is as follows:

Initialize the population size, maximum iterations, randomly generated initial solution candidate and food sources positions. A bee measures the suitability of a food source depending on the probability value $P_i$ as per the expression:

$$P_k \leftarrow \frac{fit_k}{\sum_{k=1}^{SN} fit_k} \quad (2)$$

where $fit_k$ (1 B k B SN) is the fitness value which is proportional to food source at position k and SN is the number of food sources which is equal to the employed and onlooker bees. ABC generates new candidate food source positions from old ones as

$$v_{ij} \leftarrow x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (3)$$

where, 1 B i, j B D, i, j denotes the indexes of dimension of problem. / is random number and controls the (perturbation) neighbor food sources around position x. The parameters must be within acceptable range before and after any operation pertinent to the problem to be solved. If solution is not improvised on several consecutive iterations, then bees should then find the positions through the expression (4) given as:

$$x \leftarrow x_{\min} + randgen(0, 1)(x_{\max} - x_{\min}) \quad (4)$$

In our case, the $x_{min}$ = 0 and $x_{max}$ = 255, where randgen(0, 1) routine is meant to generate any random number in interval (0, 1). The bees pick the food source according to probability value and search the vicinity to generate new solution. The acceptability of new solution is decided through greedy selection and accordingly the solution is saved if it is found better than before under specified conditions. The whole process of ABC optimization is continued for

## IV. PROPOSED OPTIMAL S-BOX CONSTRUCTION

Firstly, the initial candidate 8 X 8 substitution-box is generated through random process via chaotic Logistic map. The routine Initial_Sbox_Gen () is suggested to yield an initial S-box identified as IBox for optimization using ABC meta-heuristic approach. In this routine, the argument pr_num stands for a large prime number.The aim of the proposed optimization algorithm is to fetch a configuration of 8 X 8 S-box that holds strong cryptographic lineaments.

The entire initial IBox is considered as environment of bees with the 128th node chosen as the bees hive, so that the hive is located in the centre of the environment. The S-box values deemed as food sources for bees. The colony size for bees is 255. The number of food sources is set to SN (1 B SN B 255). Assignment of small value to SN is avoided to receive handful amount of changes in the configuration of current IBox during each optimization cycle. This much number of unique random indexes each lying between 0 and 255 (excluding 128) are generated for random food source positions.

 

These nodes or positions are stored in array X. For each node in X, the value of fitness function is determined and saved in array Fit. The probability values corresponding to each food source computed and recorded in P. The $P_{hive}$ holds the probability value for bees hive and calculated from fitness function value $Fit_{hive}$ at hive position. The onlooker bees make use of probability values to choose the food source. A loop runs for a total of SN times and compares the deviations of probability function values P[k] and P [k+1]. The Swap () function exchange the food sources via their positions. It helps the onlooker bees for searching better solution. The suitability of new configuration of IBox, after swapping, is tested and selected as local best pBox if found better than current IBox. Further, the current local best pBox is further compared with global best gBox and the gBox is updated if required. The employed bees compute the new food sources positions. The process repeats until termination condition is satisfied.

The dominance of an S-box over the other is set depending upon the values of their average nonlinearities, maximum differential uniformities, and maximum linear probabilities. The S-box is aimed to be optimized against these cryptographic properties of substitution-boxes. The reason being, they are mainly responsible for strong confusion, nonlinear transformation and potential to mitigate the cryptanalyses such as differential and linear assaults. To set the selection criteria for new S-box, consider the following:

$nl_{old}$ = mean(nonlinearities($S_{old}$))
$du_{old}$ = max(differential_uniformity($S_{old}$))
$lp_{old}$ = max(linear_probability($S_{old}$))
$nl_{new}$ = mean (nonlinearities($S_{new}$))
$du_{new}$ = max(differential_uniformity($S_{new}$))
$lp_{new}$ = max(linear_probability($S_{new}$))
condition C1 is true if $nl_{new}$C $nl_{old}$
condition C2 is true if $du_{new}$B $du_{old}$
condition C3 is true if $lp_{new}$B $lp_{old}$

The new S-box $S_{new}$ is preferred over the previous one if selection criteria C: (C1 && C2C3) is true, else previous S-box is retained for next cycle of optimization process. Where, the symbol && denotes the logical AND operation. The equality sign is considered so as to belittle the restriction which results in high number of hits. Means, $S_{new}$ dominates over $S_{old}$ if and only if $S_{new}$ is no worse than $S_{old}$ on the grounds of nonlinearity and differential uniformity. The proposed S-box optimization algorithm is presented as Optimization (). The components of secret key K of proposed algorithm include $x_0$, k, $n_0$, prenum, SN, and N.

---

lBox= intial-Sbox-Gen(x,w,n,pr-num)

1. Take an empty array of size 256
2. Iterate chaotic map n▫ times to rule out transient effect.
3. x← further iterate chaotic map once.
4. w← [ cell(x* pr-num]mod(256)
5. If (w is not member of l Box) store w to l Box
6. Repeat step 3 to 5 until lBox contains all 256 distinct w in[0,255].

---

gBox= Optimization(lBox,SN,N)

Read number of food source position as SN, iteration as N and set gBox←lBox, pBox←lBox.

1. Take an empty array X, Fit and P each size SN

2. Repeat for k←1 to SN
   x← further iterate chaotic map once
   if(η==128)
   η← generate random number in[0,255] from current chaotic variable x
   discard η and set← k-1 for regeneration
   else
   set X[k]←η
3. Repeat for k←1 to SN
   Fit[k]← compute fitness function for food source at node X[k]
4. Repeat for k←1 to SN
   P[k]←compute probability value using fitness function Fit[k] for each node
5. Repeat for k←1 to SN-1
   Compare adjacent probability deviation as:$\Delta_{dev}$←($\delta_k$ - $\delta_{k-1}$)
   If($\Delta_{dev}$>0)
   tBox←lBox
6. swap(lBox[X[k],lBox[X[k+1])
   if(lBox is dominant over pBox)
   pBox←lBox
   else

 

lBox←tBox
7. if(pBox is dominant over gBox)
update gBox as: gBox←pBox

8. computer new food source positions
9. repeat from step 4 for N number of iterations

## V. PERFORMANCE ANALYSIS OF PROPOSED S-BOX

The method proposed previous section is implemented in matlab. The initial S-box is optimized for maximizing nonlinearity, minimizing differential uniformity and minimizing linear probability. The final configuration of the proposed 8 X 8 S-box is provided in Table 1. The cryptographic strengths of S-boxes are assessed using well-accepted benchmarks among researchers such as: bijectivity, nonlinearity, strict avalanche criteria, bit-independent criteria, differential uniformity, and linear approximation probability [22, 23]. The results of performance evaluation of S-box in Table 1 is compared with most recent S-boxes, where the core designs are based on techniques of optimization, cellular automata, group theory, chaos, etc.

### 5.1 Bijectivity

An S-box is said to be bijective if all the component Boolean functions are balanced. A Boolean function $f_i$ : GF($2^8$) ? GF(2) is said to be balanced if it outputs has equal distribution of 0's and 1's (= 128). The imbalance represents weakness in terms of linear attacks [24]. The bijectivity of an 8 X 8 S-box is affirmed if its look-up has unique values in [0, $2^8$ - 1]. It is evident from the S-box LUT depicted in Table 1 that the proposed S-box is bijective as the Table has unique values in the specified range. The bijectivity of the proposed S-box is verified mathematically through the expression [11]:

$$hwt\left(\sum_{i=1}^{8} a_i f_i\right) = 2^7 \tag{5}$$

where, hwt() denotes the hamming weight, $a_i$2 f0 ; 1g and ($a_1$, $a_2$, …., $a_8$) = (0, 0, …., 0).

Table 1 The proposed 8 9 8 substitution-box

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

### 5.2 Nonlinearity

The score of nonlinearity is directly related to the strong confusion and immunity of block ciphers to mitigate linear cryptanalysis. All component Boolean functions should have as maximum nonlinearity as possible. Nonlinearity of a Boolean function $f_i$ in n-variable is measured by minimum of hamming distance between the set of all non-constant linear combinations:

$$nl_{f_i} = \min_{l \in A_n} \left\{ distance_{hamming}(f_i, l) \right\} \tag{6}$$

where $A_n$ is the set of all affine functions corresponding to function $f_i$. The nonlinearity can also be computed through Walsh spectrum as given in Ref. [26]. The best affine and linear approximation attacks [27, 28] both have reflected the significance of constructing S-boxes with high nonlinearity scores. The nonlinearity scores of all eight Boolean functions in proposed 8 X 8 S-box are computed and found as $nl_{f1}$ = 108, $nl_{f2}$ = 110, $nl_{f3}$ = 112, $nl_{f4}$=108, $nl_{f5}$ = 108, $nl_{f6}$ = 108, $nl_{f7}$ = 110, $nl_{f8}$ = 110, providing excellent $nl_{min}$ = 108, $nl_{max}$ = 112 and $nl_{avg}$ = 109.25.

 

All eight nonlinearity values are quite high and larger than or equal to 108, clearly evincing reasonably well nonlinearity performance of proposed S-box. Thus, the proposed S-box has the ability to provide high nonlinearity, strong confusion, and good resistance to linear and affine approximation attacks.

5.3 Strict Avalanche Criterion

Any cryptographic system should exhibit good avalanche effect. In 1986, Websters and Tavares introduced the concept of strict avalanche criteria for the design of good S-boxes [29]. For S-boxes, to satisfy SAC, the flipping of any single bit of input vector should leads to 50% change in output vector. An avalanche of 50% is significant to diminish any correlation among I/O combination and fails to leak information. Any value closer to 0.5 is always deemed as good. The SAC can be evaluated through an 8 X 8 dependency matrix by following the procedure reported in [29]. The average of this matrix is referred to as the SAC value. It can be seen that all values are more or less close to 0.5. The average of dependency matrix is SAC = 0.4985 which is fairly close to ideal SAC. Means, the proposed S-box tends to exhibit good avalanche and satisfy the stated criteria.

5.4 Bits Independent Criterion

Bits independent is another equally significant design criterion for strong S-boxes. Adams and Tavares suggested a method to test BIC in [30]. Consider that $f_1$, $f_2$,…, $f_8$ be the component Boolean functions of an 8 X 8 S-box. The S-box is said to fulfill BIC if the function f = $f_i$ $f_j$ (i = j, 1 B i, j B 8) is highly nonlinear and also satisfy the SAC. Based on this method, BIC for proposed S-box is verified by computing the nonlinearity and SAC of $f_i$ $f_j$. The result of BIC for nonlinearity is provided in Table 2 and that of BIC for SAC is depicted graphically in Fig. 2. The average of BIC-nonlinearity matrix is 104.29, which is a commendable score and the average of BIC-SAC matrix in Fig. 2 is 0.4992, which is very close to 0.5. The results indicate that proposed S-box is competent enough to satisfy the output bits independent criteria.

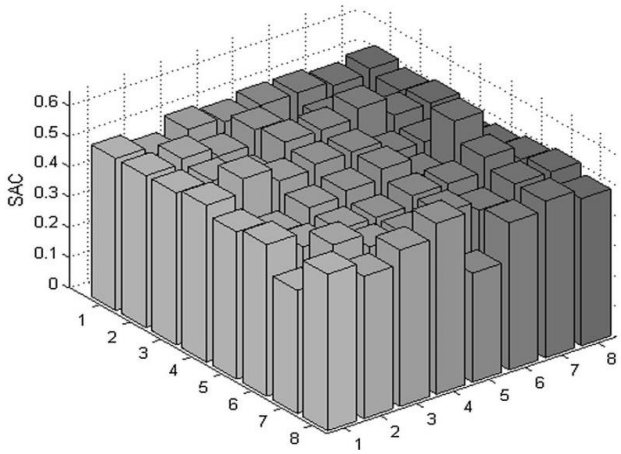


Fig. 1 Dependency matrix for SAC in graphical form

Table 2 BIC results for nonlinearity

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|---|---|---|---|---|---|---|---|---|
| $f_1$ | – | 108 | 102 | 100 | 106 | 104 | 106 | 104 |
| $f_2$ | 108 | – | 102 | 104 | 104 | 104 | 104 | 104 |
| $f_3$ | 102 | 102 | – | 98 | 102 | 106 | 106 | 104 |
| $f_4$ | 100 | 104 | 98 | – | 102 | 104 | 106 | 106 |
| $f_5$ | 106 | 104 | 102 | 102 | – | 106 | 106 | 106 |
| $f_6$ | 104 | 104 | 106 | 104 | 106 | – | 104 | 106 |
| $f_7$ | 106 | 104 | 106 | 106 | 106 | 104 | – | 106 |
| $f_8$ | 104 | 104 | 104 | 106 | 106 | 106 | 106 | – |

5.5 Differential Uniformity

The measure of differential uniformity is accounted to find the S-box capability to resist potential differential cryptanalysis. This analysis was devised by Biham and Shamir in 1991 to attack block ciphers [31]. It exploits certain occurrences of I/O differences. Dif-ferential uniformity represents the maximum likelihood of generating an output differential Dy = $Y_i$ $Y_j$ when the input differential is Dx = $X_i$ $X_j$. In this method, the XOR distribution between the inputs and outputs of S-box is determined. Mathematically, it is expressed as:

$$DU = \max_{\Delta x \neq 0, \Delta y} (\#\{x \in X | S(x) \oplus S(x \oplus \Delta x) = \Delta y\}) \quad (7)$$

where, # denotes cardinality and X is set of all inputs x. The output XOR score should have equal likelihood for corresponding input score. As a general S-box design guideline, the maximum differential uniformity has to be kept as low as possible to resist differential assaults. Following the approach adopted in [31], an input/output XOR distribution matrix of size 16 X 16 is calculated for proposed S-box and displayed graphically in Fig3.

 

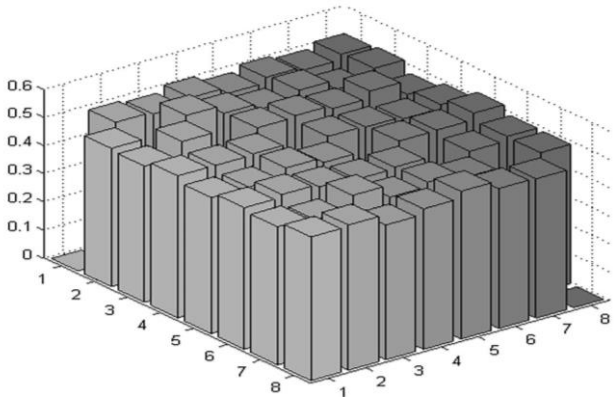

Fig. 2 BIC results for SAC

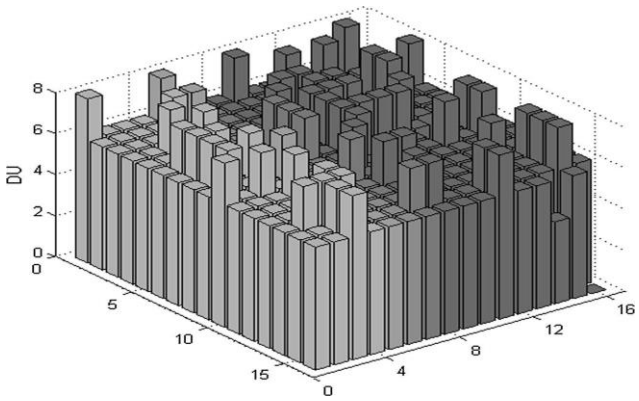


Fig. 3 Differential uniformity matrix



Table 3 Comparison of nonlinearity, SAC and BIC scores of recent 8 9 8 S-boxes

| S-box | Technique | Nonlinearity | | | SAC | BIC-NL | BIC-SAC |
|---|---|---|---|---|---|---|---|
| | | $^{nl}min$ | $^{nl}max$ | $^{nl}avg$ | | | |
| Proposed | ABC | 108 | 112 | 109.25 | 0.4985 | 104.29 | 0.4992 |
| Farah's [8] | TLBO | 104 | 110 | 106.5 | 0.4995 | 104.57 | 0.4983 |
| Ahmad's [9] | ACO | 106 | 110 | 107 | 0.5014 | 104.21 | 0.5016 |
| Guesmi's [10] | GA | 106 | 110 | 107.5 | 0.4971 | 103.86 | 0.5034 |
| Wang's [11] | GA | 108 | 108 | 108 | 0.5068 | 103.36 | 0.5017 |
| Wang's [12] | GA | 108 | 110 | 109 | 0.4988 | 104.79 | 0.5026 |
| Ahmad's [33] | TSP | 106 | 110 | 107.5 | 0.5036 | 103.93 | 0.5040 |
| Bhatacharya's [34] | CA | 102 | 108 | 105 | 0.4888 | 103.29 | 0.5011 |
| Jamal's [35] | Group | 104 | 110 | 106.75 | 0.4988 | 106.28 | 0.5010 |
| Khan's [36] | Chaos ? Group | 96 | 106 | 103.25 | 0.5151 | 103.07 | 0.4864 |
| Khan's [37] | Chaos | 84 | 106 | 100 | 0.4812 | 101.93 | 0.4967 |
| Khan's [38] | Chaos ? Group | 98 | 110 | 105.5 | 0.4937 | 105.7 | 0.5013 |
| Islam's [39] | Chaos | 102 | 108 | 106 | 0.5002 | 104.4 | 0.5013 |
| Belazi's [40] | Chaos | 100 | 110 | 105.5 | 0.5000 | 103.78 | 0.4970 |
| Ozkaynak's [41] | Chaos | 100 | 108 | 104.7 | 0.4982 | 103.1 | 0.4942 |
| Lambic's [42] | Chaos | 106 | 108 | 106.75 | 0.5034 | 103.78 | 0.5015 |
| Cavusoglu's [43] | Chaos | 104 | 110 | 106 | 0.5039 | 103.38 | 0.5058 |
| Anees's [44] | Chaos | 100 | 106 | 103 | 0.5020 | 102.93 | 0.4998 |
| Kazlauskas's [45] | Pseudo-random | 102 | 106 | 104 | 0.5009 | 103.36 | 0.5061 |
| Khan's [46] | Chaos | 100 | 108 | 104.75 | 0.4978 | 103.6 | 0.5009 |
| Liu's [47] | Chaos | 104 | 108 | 105.80 | 0.4976 | 104.5 | 0.5032 |
| Hongjun's [48] | Chaos | 102 | 106 | 104 | 0.5018 | 103.5 | 0.5019 |
| Gondal's [49] | Chaos | 98 | 106 | 103 | 0.4961 | 104.14 | 0.5043 |

The highest value of this table is referred as differential uniformity of S-box. The differential uniformity for our S-box is found as 8 which is the highest value of DU in Fig. 3.

5.6 Linear Approximation Probability

The well studied linear cryptanalysis was introduced by Matsui in 1993 [28] to break popular DES block cipher. It is a known-plaintext assault which approximates relationship between inputs, outputs and key. The magnitude of linear approximation probability should be as low as possible to resist this attack [24]. According to Matsui, LAP is the highest value of event that the parity of input bit selected by mask $x_x$ is same as the parity of output bits selected by mask $x_y$. LAP is quantified as:

$$LAP = \max_{\omega_x,\omega_y \neq 0} \left| \frac{\#\{x \in X | x.\omega_x = S(x).\omega_y\}}{2^n} - \frac{1}{2} \right| \quad (8)$$

where, xx and xy denoted the mask values of inputs and outputs. X is the set of all possible inputs x whose cardinality is 2n. An S-box having lower LAP score has better resistance to linear cryptanalysis. The maximum value of LAP for proposed S-box comes out as 0.1250 which is fairly low compared to many recently investigated S-boxes.

 

5.7 Performance Comparison

The capabilities of a designed S-box should be such that it has (1) adequate power to thwart the chosen-plaintext attack (differential cryptanalysis) practiced by Biham and Shamir in [31], and the known-plaintext attack (linear cryptanalysis) regulated by Matsui in [28], (2) the ability to provide strong confusion in ciphers, [32] and (3) ability to diminish any correlation among I/O bit patterns so as not to leak information [24]. The performance metrics are quantified to assess these inherent capabilities of S-boxes. As design guidelinesfor cryptographically strong S-boxes, the maximization of nonlinearity is targeted, the largest value of DU and LAP should be as low as possible, the offset of SAC (difference from ideal SAC of 0.5) should be as close to zero as possible. Similarly, the maximization of nonlinearity and satisfaction of SAC during testing bits independent criteria.

To make a clean comparison, we picked most recent state-of-the-art algorithms [8–12, 33–49] where the techniques of optimization, cellular automata, group theory, and chaos are applied to generate efficient 8 9 8 S-boxes. In all the selected 22 S-box proposals, the authors have also made a comparative analysis to show that their designed S-box has better performance over many previous S-boxes investigated in the literature. The comparison of our proposed S-box is done in Tables 3 and 4 based on the metrics discussed and evaluated in previous subsections. The comparison made in tables show that the proposed S-box outperforms over the S-boxes. The average nonlinearity of 109.25 is highest (best) and the maximum value of differential uniformity of 8 is lowest (best) among all other S-boxes. On the aspect of other metrics such as SAC, BIC-NL, BIC-SAC and LAP, the proposed S-box has better scores over most of the other S-boxes in table. The success rate of proposed S-box over other S-boxes for each metric is counted from two tables and accumulated in Table 5. It is found that the proposed S-box has a success rate of 90.9% on min nonlinearity, 100% on max and average nonlinearity, 68.2% on SAC, 72.7% on BIC-NL, 95.5% on BIC-SAC, 100% on DU, and 66.6% on LAP over other competing S-boxes. The high success rates for each metric unarguably narrate the reasonably out-standing security performance of proposed S-box.

Table 4 Comparison of maxi-mum DU and LAP scores of most recent 8 9 8 S-boxes

| S-box | Technique | DU | LAP |
|---|---|---|---|
| Proposed | ABC | 8 | 0.1250 |
| Farah's [8] | TLBO | 10 | 0.1172 |
| Ahmad's [9] | ACO | 10 | 0.1484 |
| Guesmi's [10] | GA | 10 | 0.1250 |
| Wang's [11] | GA | 10 | 0.1406 |
| Wang's [12] | GA | 10 | 0.1406 |
| Ahmad's [33] | TSP | 10 | 0.1484 |
| Bhatacharya's [34] | CA | 10 | 0.1250 |
| Jamal's [35] | Group | 30 | 0.1250 |
| Khan's [36] | Chaos ? Group | 44 | 0.1562 |
| Khan's [37] | Chaos | 16 | 01796 |
| Khan's [38] | Chaos ? Group | 32 | 0.1172 |
| Islam's [39] | Chaos | 10 | 0.1484 |
| Belazi's [40] | Chaos | 12 | 0.1250 |
| Ozkaynak's [41] | Chaos | 10 | 0.1406 |
| Lambic's [42] | Chaos | 10 | 0.1328 |
| Cavusoglu's [43] | Chaos | 10 | 0.1406 |
| Anees's [44] | Chaos | 10 | 0.1406 |
| Kazlauskas's [45] | Pseudo-random | 12 | 0.1328 |
| Khan's [46] | Chaos | 12 | 0.1406 |
| Liu's [47] | Chaos | 10 | 0.1250 |
| Hongjun's [48] | Chaos | 10 | NA |
| Gondal's [49] | Chaos | 12 | 0.1484 |

Table 5 Success rate of pro-posed S-box when compared with most recent S-boxes on aspects of various criteria

| Performance criteria | Success rate |
|---|---|
| $^{NL}$min | 20/22 |
| $^{NL}$max | 22/22 |
| $^{Nl}$avg | 22/22 |
| SAC | 15/22 |
| BIC-NL | 16/22 |
| BIC-SAC | 21/22 |
| DU | 22/22 |
| LAP | 14/21 |

 

## VI. CONCLUSION

An 8 X 8 replacement box construction method based on the technique of optimization of the artificial bee colony is suggested, in which an initial candidate for the S-box is created by a chaotic logistic map. The initial S-box is configured according to three criteria: the maximisation of nonlinearity, minimum uniformity and linear likelihood of approximation. The process of optimised S-box generation depends on the key. By making minor changes to some of the key components, a range of dynamical S- boxes can be produced. The safety evaluation of the S-box proposed and a careful comparison with the new S-boxes defended the outstanding performance of the system. You can use the proposed S-box to build crypto-strong block systems.

## REFERENCES


[1]. Menezes, A. J., Oorschot, P. C. V., & Vanstone, S. A. Handbook of applied cryptography. Boca Raton: CRC Press. **1997**

[2]. Stinson, D. R. Cryptography: Theory and practice. Boca Raton: CRC Press. **2005**

[3]. Schneier, B. Applied cryptography: Protocols algorithms and source code in C. New York: Wiley. **1996**

[4]. Knudsen, L. R., & Robshaw, M. The block cipher companion. Berlin: Springer. **2011**

**[5].** Ozkaynak, F., & Sirma, Y. Designing chaotic S-boxes based on time-delay chaotic system. Nonlinear Dynamics, **74(3), 551–557. 2013**

[6]. Cui, L., & Cao, Y. A new S-box structure named Affine-Power-Affine. International Journal of Innovative Computing, Information and Control, **3(3), 751–759. 2007**

[7]. Hussain, I., & Shah, T. (2013). Literature survey on nonlinear components and chaotic nonlinear components of block ciphers. Nonlinear Dynamics, 74(4), 869–904.

[8]. Farah, T., Rhouma, R., & Belghith, S. (2017). A novel method for designing S-box based on chaotic map and teaching–learning-based optimization. Nonlinear Dynamics, 88(2), 1059–1074.

[9]. Ahmad, M., Bhatia, D., & Hassan, Y. (2015). A novel ant colony optimization based scheme for substitution box design. Procedia Computer Science, 57, 572–580.

[10]. Guesmi, R., Farah, M. A. B., Kachouri, A., &Samet, M. (2014). A novel design of Chaos based S-boxes using genetic algorithm techniques. In IEEE/ACS 11th international conference on computer systems and applications (AICCSA) (pp. 678–684).

[11]. Wang, Y., Wong, K. W., Li, C., & Li, Y. (2012). A novel method to design S-box based on chaotic map and genetic algorithm. Physics Letters A, 376(6), 827–833.

[12]. Yong, W., & Peng, L. (2012). An Improved method to obtaining S-box based on chaos and genetic algorithm. HKIE Transactions, 19(4), 53–58.

[13]. Clark, J. A., Jacob, J. L., &Stepney, S. (2005). The design of S-boxes by simulated annealing. New Generation Computing, 23(3), 219–231.

[14]. Millan, W. (1998). How to improve the nonlinearity of bijective S-boxes. In Australasian conference on information security and privacy, lecture notes in computer science (Vol. 1438, pp. 181–192).

[15]. Fuller, J., Millan, W., & Dawson, E. (2005). Multi-objective optimisation of bijective S-boxes. New Generation Computing, 23(3), 201–218.

[16]. Laskari, E. C., Meletiou, G. C., &Vrahatis, M. N. (2006). Utilizing evolutionary computation methods for the design of S-boxes.In International conference on computational intelligence and security (pp. 1299–1302).

[17]. May, R. M. (1976). Simple mathematical models with very complicated dynamics. Nature, 261(5560), 459–467.

[18]. Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200). Technical report-tr06, Erciyes University, Faculty of Engineering, Department of Computer Engineering.

[19]. Tereshko, V. (2000). Reaction–diffusion model of a honeybee colony's foraging behaviour. In M. Schoenauer (Ed.), Parallel problem solving from nature VI (Vol. 1917, pp. 807–816)., Lecture notes in computer science Berlin: Springer.

[20]. Karaboga, D., &Akay, B. (2009). A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation, 214(1), 108–132.

[21]. Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. Artificial Intelligence Review, 42(1), 21–57.

[22]. Dawson, M. H., & Tavares, S. E. (1991). An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks. Advances in Cryptology, Lecture Notes in Computer Science, 547, 352–367.

[23]. Braeken, A. (2006). Cryptographic properties of Boolean functions and S-boxes.Ph.D. thesis available at http://homes.esat.kuleuven.be/abraeken/thesisAn.pdf.Accessed 21 May 2017.

[24]. Burnett, L. (2005). Heuristic optimization of boolean functions and substitution boxes for cryptography.Doctoral dissertation, Queensland University of Technology.

[25]. Isa, H., Jamil, N., &Zaba, M. R. (2015). Improved S-box construction from binomial power functions. Malaysian Journal of Mathematical Sciences, 9(S), 21–35.

[26]. Cusick, T. W., &Stanica, P. (2009). Cryptographic Boolean functions and applications. Amsterdam: Elsevier.

[27]. Ding, C., Xiao, G., & Shan, W. (1991). The stability theory of stream ciphers (Vol. 561)., LNCS Berlin: Springer.

[28]. Matsui, M. (1994). Linear cryptanalysis method for DES cipher. In Proceedings of EUROCRYPT'93, lecture notes in computer science (Vol. 765, pp. 386–397).

[29]. Webster, A. F., & Tavares, S. E. (1986). On the design of S-boxes. Advances in Cryptology, Lecture Notes in Computer Science, 218, 523–534.

[30]. Adams, C., & Tavares, S. (1990). The structured design of cryptographically good S-boxes. Journal of Cryptology, 3(1), 27–41.

[31]. Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4(1), 3–72.

[32]. Sarfraz, M., Hussain, I., & Ali, F. (2016). Construction of S-Box based on Mobius transformation and increasing its confusion creating ability through invertible function.International Journal of Computer Science and Information Security, 14(2), 187–18x.

[33]. Ahmad, M., Mittal, N., Garg, P., & Khan, M. M. (2016). Efficient cryptographic substitution box design using travelling salesman problem and chaos.Perspectives in Science, 8, 465–46.


 