A New S-Box Design by Applying Bat Algorithm Based Technique

Maiya Din *
*Corresponding author, Scientist, M.Sc., DRDO HQrs, New Delhi, India. E-mail: anuragimd@gmail.com

Saibal K. Pal
Scientist, PhD., DRDO, Delhi, India. E-mail: skptech@yahoo.com

S.K. Muttoo
Professor (Retd.), PhD., 505, Patel Society, Dwarka, New Delhi, India. E-mail: drskmuttoo@gmail.com

Sushila Madan
Professor, PhD., LSR College, University of Delhi, New Delhi, India. E-mail: sushila.madan@gmail.com

Abstract

Substitution-boxes (S-boxes) are very important nonlinear components used for achieving strong confusion for enhancing cryptographic security in most of the block ciphers. Designing cryptographically strong S-boxes has been a major research domain for the designers of symmetric crypto systems. In the proposed research work, Bat Algorithm based swarm technique is proposed to design strong S-boxes. Cryptographic strong S-boxes are obtained by the developed swarm technique. Authors analyze cryptographic strength of the obtained S-box by evaluating properties like Bijectivity, Nonlinearity, Bit-Independence Criterion, Linear Probability and Differential Uniformity. The obtained performance parameters for the designed new S-box by the swarm technique are compared with some recently reported S-boxes in the literature. The designed S-box has good cryptographic strength. The designed S-box has good cryptographic strength like nonlinearity = 110.75 and average Strict Avalanche Criterion (SAC) value = 0.506. For the constructed S-box, most of the Differential uniformity components are 4 and shows uniform distribution approximately. The proposed new S-box is also free from the fixed points.

Keywords: Cryptography; Block Cipher; S-box; Nonlinearity; Bat Algorithm.
Introduction

Cryptography is a branch of cryptology concerned with design of cryptosystems (Stallings, 2012).

With the advancement of communication technologies, design of encryption techniques for secure communication of confidential information over insecure channel, have attracted major attention of research community. Encryption algorithms are mainly categorized as stream and block ciphers. The principles of confusion and diffusion are ensured during design phase of block ciphers (e.g., DES, IDEA, PRESENT, RC6 and AES). An S-box is a vital and most common crypto primitive in block ciphers. The S-box employed within a block cipher imparts confusion and makes the system resilient to defend against cryptanalytic attacks applied by the cryptanalyst for breaking the crypts. In most of the block ciphers, S-box is an important nonlinear element due to fulfilling crypto characteristics like Nonlinearity, Bit-Independence Criterion, Linear Probability, Differential Uniformity and Bijectivity. The strength of the employed S-box contributes significantly in over all cryptographic security of the cryptosystem.

The cryptographic strength of block ciphers mainly depends on applied S-boxes. There are three types of S-box designing process, known as, random search-based techniques, algebraic techniques, and heuristic based techniques. Finite Field inversion scheme is used in algebraic techniques for constructing S-box satisfying required cryptographic properties. The ‘Random Search’ based techniques are easy to implement and produce large number of S-boxes. However, quality of generated S-boxes using random search are far from the S-boxes produced by algebraic techniques. The third type of designing process apply various heuristics to design S-boxes satisfying most of the cryptographic strength parameters. There is a performance gap between algebraic and heuristics-based techniques. Therefore, meta-heuristics based iterative swarm techniques are applied for constructing S-boxes with enhanced cryptographic strength.

Review of Related Work

S-box is defined as a set of Boolean functions. An n × m S-box is a function S: \{0, 1\}^n → \{0, 1\}^m , here n and m are positive integers indicating size of input and output string respectively.
This maps an input string to an output string nonlinearly. In many ciphers, deployed S-boxes have same size of i/p and o/p string. S-boxes of size $(4\times4)$ and $(8\times8)$ are utilized in many block ciphers. The S-box $(8\times8)$ are used in highly secure ciphers, like AES and S-box $(4\times4)$ is generally used in light weight cryptosystems like PRESENT Cipher (Stinson, 2013).

The correlation between the input-output strings is also determined for satisfying the security criteria of the S-box. Nonlinearity is a very important cryptographic property that imparts confusion for resisting against linear cryptanalysis of block ciphers (Hussain et al., 2012) (Özkaynak & Yavuz, 2013). After development of linear and differential cryptanalytic attack on S-boxes of DES cipher, a need was felt for the design of cryptographically strong S-boxes. Consequently, several S-box design schemes were evaluated which were based on optimization techniques. Tang et al proposed a novel technique using discrete Baker map for construction of S-box (Tang et al., 2005). An S-box construction method based on chaotic maps was presented by Jakimoski (Jakimoski & Kocarev, 2001). Wang et al presented a design technique based on genetic algorithm and chaos to compute improvised S-box in the proposed work (Wang et al., 2012). Ahmed and Bhatia proposed a nature-inspired swarm-based technique by applying Ant Colony optimization (ACO) for generating strong S-boxes (Ahmad et al., 2015b). Farah et al presented a technique for achieving S-box by applying “Teaching-Learning-Based Optimization (TBLO)” swarm algorithm (Farah et al., 2017).

The firefly algorithm is an iterative, nature-inspired swarm technique developed by Yang. The algorithm has been applied by researchers to solve real world optimization problems. Many researchers have developed several variants of the algorithm in their research work (Fister et al., 2013) (Gandomi et al., 2013). Since the proposed FA, many modifications have been made in the schemes which provided many variants to solve real-world continuous and constrained optimization cases (Yang, 2015).

(Yang, 2009). Many variants of the algorithm also developed in last decades due to performance comparison between FA and other well known swarm techniques.

The Bat swarm algorithm has shown its capability in addressing various issues of optimization, however its search behavior usually depends on initial positions and controlling parameters. Performance of the algorithm can be improved further by initializing it from a suitable initial point. In meta-heuristic-based optimization techniques, generally random initialization is done in every execution of the program of the algorithm (Parpinelli & Lopes, 2011) (Tsai et al., 2012) (Yang et al., 2013).

Swarm intelligence-based techniques are also applied in cryptanalysis of several ciphers (Laskari et al., 2007) (Din et al., 2019b) (Din et al., 2019a). In last two decades, some methods have been developed to design S-boxes by employing chaotic maps due to their nonlinear
chaotic behavior (Webster & Tavares, 1986) (Asim & Jeoti, 2008) (Özkaynak, 2019) (Lambić, 2014) (Ahmad et al., 2015a) (Ahmad et al., 2018). According to above reported motivating research on S-box design the authors propose a swarm technique by employing nature-inspired Bat Algorithm (BA).

In this research paper, related work on design of S-boxes by applying swarm-based techniques is reviewed. The next Section describes Bat Algorithm and BA based computational intelligence technique. The performance results of achieved S-box are discussed in the last Section.

**Bat Algorithm**

A nature-inspired Bat Swarm Algorithm was proposed by Yang X.S. in 2010 (Yang, 2010a).

BA is a population based meta-heuristic technique for solving optimization problems. It was inspired by the echolocation behavior of microbats, with varying pulse rates of emission and loudness. Each microbat of the swarm searches an optimal solution of the considered problem.

This algorithm is naturally inspired form the social behavior of bats. The capability of the echolocations of these bats composed a great competent manner to detect prey, avoid obstacles, and to locate their rooster crevices in the dark based on sensing distances. In order to formalize the bat algorithm optimally, the following bat’s echolocation characteristics should be idealized:

(i) Object distances are always perfectly sensed by the echolocation system on bats. This makes the ability to differentiate between different objects even in darkness.

(ii) Bats are flying randomly with velocity \( v \), fixed frequency \( f_{\text{min}} \) at position \( x_i \) and fluctuating wavelength \( \lambda \) and loudness form large loudness \( A_0 \) to minimum loudness \( A_{\text{min}} \) to search for its prey. Wavelength or frequency can be changed instantly by adjusting the pulse emission rate \( r \in [0,1] \) based on the closeness of the bat objective.

(iii) Variation of the loudness parameter takes values between \( A_0 \) and \( A_{\text{min}} \).

The pseudo code of the bat algorithm is mentioned in figure 1. It starts with the initialization of all the echolocation system variables. Initial location of all bats swarms should be initialized as initial solutions.

\[
 f_i = f_{\text{min}} + (f_{\text{max}} - f_{\text{min}}) \beta 
\]

\[
 v_i^t = v_i^{t-1} + (x_i^t - x^*) f_i 
\]

(1) (2)
\[ x_i^t = x_i^{t-1} + v_i^t \]  

(3)

Here;

\( \beta \): Random vector drawn from uniform distribution \( \beta \in [0, 1] \).

\( x^* \): Current global best location (solution) which is located after comparing all the solutions among all the bats.

\( f_i \): Frequency which is drawn uniformly from \([f_{\text{min}}, f_{\text{max}}]\)

A random walk with direct exploitation is used for the local search that modifies the current best solution according to the equation:

\[ x_{\text{new}} = x_{\text{old}} + \varepsilon A^t \]  

(4)

Here;

\( \varepsilon \): is a random number \( \in [-1, 1] \).

At: is the average loudness of all the best at this time step.

\( r_i \): is the rate of pulse emission

For each bat, as soon as the prey found, the bat loudness decrease and the pulse emission rate increase. Both loudness and pulse emission expressed mathematically as follows:

\[ A_i^{t+1} = \alpha A_i^t \]  

(5)

\[ r_i^{t+1} = r_i^0 \left[ 1 - \exp(-\gamma t) \right] \]  

(6)

\[ A_i^t \to 0 \quad \text{and} \quad r_i^t \to r_i^0 \quad \text{as} \quad t \to \infty \]  

(7)

Here; \( \alpha \): is constant \( 0 < \alpha < 1 \) and \( \gamma \): is constant \( \gamma > 0 \)
The Pandemic Benefits Reaped by Online Teaching Platforms: A Case study…

Figure 1. Bat Algorithm

**Proposed Bat Algorithm based Technique**

The BA is based on automated subdivision and have multimodality handling capacity. According to the subdivision criteria, the fireflies try to discover optimized solution applying parallel optimizations, particularly when population size is sufficiently higher compared to the modes. The proposed technique based on BA for designing $8 \times 8$ Substitution-boxes is as follows:

**Initialization of Swarm Bats**

1. Initialize $n$: No. of bats
2. Initialize each bat with a random permutation vector (P), having values in $[0, 255]$ without repetition.
3. Array P is reshaped into a 16 x 16 two-dimensional matrix representing initial S-box, which shows initial position of the
ith bat \((f_i)\).

(4) Steps 2 to 3 are repeated for generating all \(n\) bats such that each bat represents one initial S-box.

**Fitness function**

Fitness value of each bat is calculated on the basis of nonlinearity of each S-box as follows:

4.2.1 Nonlinearity of S-box is calculated using following equation:

\[
N_f = 2^{n-1}(1 - 2^{-n} \max_{w \in GF(2^n)} |WS < f > (w)|)
\]

(8)

Here, the Walsh spectrum of \(f_x\) is determined as per equation follows:

\[
WS(f)(w) = \sum_{x \in GF(2^n)} (-1)^{f(x) \oplus x \cdot w}
\]

(9)

for the Eq. (4) \(w \in GF(2^n)\) and the dot product \((x \cdot w)\) is computed as follows:

\[
x \cdot w = x_1 \cdot w_1 \oplus \ldots \oplus x_n \cdot w_n
\]

The fitness value of \(i\)th bat \((fi)\) is determined using following equation.

\[
fitness = 112 - NL(f_i)
\]

(10)

Here, 112 is an optimal value of S-Box nonlinearity mentioned in the Block Cipher AES (Stinson, 2013), and NL function determines nonlinearity of \(i\)th bat corresponding to S-box. In the current iteration, the bat with lowest fitness value provides the best S-box.

\[
r_{ij} = \sqrt{\sum_{k=1}^{d} (P_{i,k} - P_{j,k})^2}
\]

(11)

Where, \(P_{i,k}\) represents \(k\)th element of the S-box corresponding to \(i\)th bat and \(d\) is set to 255.

After every movement of bats, the obtained values are checked for keeping within the specified threshold values: 0 and 255. This is ensured by satisfying Bijective property of the S-Box corresponding to the bat.

**Minor Adjustment**

An S-box \((8 \times 8)\) is represented as a vector of 256 elements. As per Bijective property of an S-box, each element occurs only once in the vector. Sometime this property is not fully satisfied
Performance Analysis of the Proposed S-box

The proposed Bat algorithm-based technique is implemented using MATLAB software (Ver. R2017a). The developed software is run on the 3.0 GHz computing machine. Various trials are conducted for different values of the control parameters of the algorithm to compute cryptographic strong S-boxes.

The parameters of proposed technique are taken as: $\alpha = 0.25$, $\beta = 0.15$ to 0.35, $\gamma = 1.0$, iterations = 750, swarm size (N) = 20. The obtained S-box is shown in the following Table:

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

Many performance measures for S-box are defined by expert designers for examining cryptographic security strengths of an S-box. Biham and Shamir proposed the differential uniformity criteria through its differential cryptanalysis of DES block cipher based on differential probability, calculated using I/O XOR distribution (Menezes et al.,2018) (Biham & Shamir, 1991). Matsui proposed linear cryptanalysis based on linear approximation probability (Matsui, 1993). Dawson et.al applied information theoretic concepts for designing and evaluation of cryptographic strong S-boxes (Dawson & Tavares, 1991). Therefore, cryptographic security of designed S-boxes is ensured by nonlinearity, bijectivity, strict avalanche criterion (SAC), bit-independence criterion (BIC), differential uniformity and linear-approximation probability (LP) (Wang et al., 2009).
Bijective Property

The bijectivity implies that each item of the S-box occurs only once.

\[ wt(\sum_{i=1}^{n} a_i f_i) = 2^{n-1} \]  

(12)

Where \( a \in [0,1] \) and \( wt(\cdot) \) is the Hamming weight. As per equation (13), bijectivity is fulfilled when all Hamming weights are equal to 128 for the designed S-box.

Nonlinearity

Nonlinearity of S-box is calculated using equation (3). The non-linearity is connected to immunity of block cipher and plain text confusion. This is computed using Walsh spectrum as mentioned in equation (4). The computed nonlinearity corresponding to the Boolean functions of the obtained S-box is shown in following table and depicted in Figure 2.

<table>
<thead>
<tr>
<th>S-box</th>
<th>L_1</th>
<th>L_2</th>
<th>L_3</th>
<th>L_4</th>
<th>L_5</th>
<th>L_6</th>
<th>L_7</th>
<th>L_8</th>
<th>Min.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>110</td>
<td>112</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td>112</td>
<td>110</td>
<td>112</td>
<td>110</td>
</tr>
</tbody>
</table>

Table 2. Nonlinearity of S-box

![Nonlinearity](image)

Figure 2. Nonlinearity of 8 Boolean Functions of S-box

Strict Avalanche Criteria

Webster and Tavares (Webster & Tavares, 1986) defined SAC by ensuring that there is a probability of \( \frac{1}{2} \) for a change in all the output bits when single input bit is flipped. SAC is ensured through dependence matrix and offset values of SAC are computed using following equation:

\[ S(f) = \frac{1}{n^2} \sum_{1 \leq i < n} \sum_{1 \leq j \leq n} \left| 0.5 - p_{ij}(f) \right| \]  

(13)
Bit Independence Criteria (BIC)

This stipulates the pair-wise independence of all avalanche variables for any avalanche vector generated by the complementing of a single plaintext. The criteria was defined by Webster and Tavares (Webster & Tavares, 1986).

I/O XOR distribution

For an ideal S-box, the obtained XOR distribution table should be balanced since imbalance distribution table helps in applying differential cryptanalytic attack. A cryptographic strong S-box should be differential uniform. For a given S-box, differential uniformity is measured using equation (14).

\[ DP(f) = \max_{a \neq 0} \left| \frac{\#\{x \in X | f(x) \oplus f(x \oplus \Delta x) = \Delta y \}}{2^n} \right| \]  \hspace{1cm} (14)

Here, X contains all the input values and total number of elements is shown by 2n. Table 3 shows the I/O XOR distribution for the obtained S-box.

| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |

Linear approximation probability

The LP probability is determined as the maximum imbalance value. Thus, mask a selects input bits that have parity equal to the output bits selected by mask b. The equation (15) is applied for computing LP.

\[ LP = \max_{a, b \neq 0} \left| \frac{\#\{x \in X | a = f(x), b \}}{2^n} \right| - 0.5 \]  \hspace{1cm} (15)

For a given S-box, LP should be as small as possible. An S-box with lower LP has more resistance against linear cryptanalysis than other S-box with higher LP.
Table 4. The Dependence Matrix for the Computed S-Box

<table>
<thead>
<tr>
<th></th>
<th>0.469</th>
<th>0.492</th>
<th>0.500</th>
<th>0.484</th>
<th>0.461</th>
<th>0.469</th>
<th>0.539</th>
<th>0.469</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.445</td>
<td>0.508</td>
<td>0.516</td>
<td>0.516</td>
<td>0.508</td>
<td>0.484</td>
<td>0.461</td>
<td>0.563</td>
<td></td>
</tr>
<tr>
<td>0.378</td>
<td>0.531</td>
<td>0.398</td>
<td>0.516</td>
<td>0.453</td>
<td>0.414</td>
<td>0.531</td>
<td>0.477</td>
<td></td>
</tr>
<tr>
<td>0.453</td>
<td>0.516</td>
<td>0.602</td>
<td>0.383</td>
<td>0.555</td>
<td>0.523</td>
<td>0.438</td>
<td>0.477</td>
<td></td>
</tr>
<tr>
<td>0.563</td>
<td>0.461</td>
<td>0.531</td>
<td>0.539</td>
<td>0.641</td>
<td>0.500</td>
<td>0.453</td>
<td>0.563</td>
<td></td>
</tr>
<tr>
<td>0.492</td>
<td>0.477</td>
<td>0.531</td>
<td>0.461</td>
<td>0.633</td>
<td>0.500</td>
<td>0.484</td>
<td>0.508</td>
<td></td>
</tr>
<tr>
<td>0.398</td>
<td>0.555</td>
<td>0.484</td>
<td>0.461</td>
<td>0.523</td>
<td>0.625</td>
<td>0.469</td>
<td>0.539</td>
<td></td>
</tr>
<tr>
<td>0.492</td>
<td>0.563</td>
<td>0.570</td>
<td>0.523</td>
<td>0.508</td>
<td>0.531</td>
<td>0.453</td>
<td>0.453</td>
<td></td>
</tr>
</tbody>
</table>

A swarm technique applying Bat Algorithm is proposed for designing strong S-boxes. The above-mentioned results of the S-box obtained by the swarm technique have shown that most of the criteria of a good S-box are fulfilled. The obtained S-box has high immunity against differential cryptanalytic attacks. The computed S-box is free from the fixed points. The maximum nonlinearity of the S-box is 112 and minimum is 110. The average nonlinearity stands at 110.75. The average SAC value for the S-box is 0.506.

According to the Table of Input / Output XOR distribution, DP values are approximately uniformly distributed and most of the values are 4.

Table 5. Performance Comparison of the New S-box

<table>
<thead>
<tr>
<th>S-box design</th>
<th>Nonlinearity</th>
<th>SAC</th>
<th>I/O XOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technique</td>
<td>Min.</td>
<td>Max.</td>
<td>Avg.</td>
</tr>
<tr>
<td>(Hussain et al., 2012)</td>
<td>102</td>
<td>108</td>
<td>104.7</td>
</tr>
<tr>
<td>(Özkaynak &amp; Yavuz, 2013)</td>
<td>103</td>
<td>109</td>
<td>105.1</td>
</tr>
<tr>
<td>(Ahmad et al., 2015b)</td>
<td>106</td>
<td>110</td>
<td>107.0</td>
</tr>
<tr>
<td>(Farah et al., 2017)</td>
<td>104</td>
<td>108</td>
<td>106.5</td>
</tr>
<tr>
<td>(Özkaynak, 2019)</td>
<td>106</td>
<td>108</td>
<td>106.7</td>
</tr>
<tr>
<td>(Ahmed et al., 2019)</td>
<td>106</td>
<td>108</td>
<td>107.5</td>
</tr>
<tr>
<td>(Alhadawi et al., 2021)</td>
<td>106</td>
<td>110</td>
<td>108.5</td>
</tr>
<tr>
<td>(Lambič, 2017)</td>
<td>106</td>
<td>108</td>
<td>106.7</td>
</tr>
<tr>
<td>New S-box by Proposed Technique</td>
<td>110</td>
<td>112</td>
<td>110.75</td>
</tr>
</tbody>
</table>

According to the Table 5, the performance indicators for the constructed S-box are compared with some recently reported S-boxes. The maximum nonlinearity of the S-box is 112 and minimum is 108. The computed average nonlinearity is 110.75. The average SAC value for the S-box is 0.506. The obtained maximum DP value in the I/O XOR distribution table is 6. The proposed new S-box is also free from the fixed points.

Authors addressed the issues of designing cryptographic strong S-boxes. An efficient Bat algorithm-based technique is proposed to generate S-boxes. The computed S-box has better performance as compared with some of the existing S-boxes. The achieved S-Box meets most
of the cryptographic requirements. Authors plan to focus on designing strong S-boxes having different sizes of inputs and outputs by developing efficient swarm techniques based on other meta-heuristics and swarm intelligence.

Acknowledgements

My sincere thanks to the co-authors for their valuable technical guidance and motivation in carrying out the proposed research work and supporting me in research paper submission.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References


Yang X-S (2015). Analysis of firefly algorithm and automatic parameter tuning. In emerging research on swarm intelligence and algorithm optimization, IGI Global, 36-49


---

**Bibliographic information of this paper for citing:**


---

Copyright © 2023, Maiya Din, Saibal K. Pal, S.K. Muttoo and Sushila Madan