# A Novel Design of Chaos Based S-Boxes Using Genetic Algorithm Techniques

Ramzi Guesmi, Mohamed Amine Ben Farah, Abdennaceur Kachouri and Mounir Samet
Laboratory of Electronics and Information Technology,
National Engineering School of Sfax, Sfax University, B.P.W. 3038 Sfax, Tunisia.
Email: ramzi.guesmi@gmail.com, med.farah@yahoo.fr, abdennaceur.kachouri@enis.rnu.tn and mounir.samet@enis.rnu.tn

***Abstract*—In this manuscript we present a novel method to design strong substitution Boxes based on chaos function and genetic algorithm techniques. Furthermore, we analyse the strength of the proposed S-Boxes. The proposed methodology is analyzed and tested for the following criteria: bijective property, nonlinearity, strict avalanche criterion, output bits independence criterion and equiprobable input/output XOR distribution. Numerical simulation and security analysis demonstrate that the scheme is practical in image encryption.**

***Keywords*—S-Box, Chaos, Cryptography, Nonlinearity, Genetic algorithm**

## I. INTRODUCTION

The substitution box (S-box) is an important component in block encryption algorithms. As one of the core components of cryptography, S-boxes have been widely used in cryptographic algorithms such as DES, AES and IDEA. In recent years, some methodologies based on chaos functions have been designed to generate S-boxes. The main reason for using chaos is to take advantage of its nonlinear property. In [1], Jakimoski and Kocarev proposed a four-step method for designing S-boxes based on chaos. Tang et al. proposed an algorithm for generating S-box based on a 2D discretized chaotic Baker map [2]. Chen et al. designed another method to obtain 8 x 8 S-box by employing a Chebychev map and a 3D Baker map [3]. Ozkaynak, proposed an S-box based on continuous-time Lorenz system [4]. Hussain, proposed a method based on NCA (nonlinear chaotic algorithm) chaotic map in [5]. Zaibi, designed a chaotic S-box suitable for implementation on wireless sensor nodes [6].

The use of S-boxes has become popular in image encryption algorithms as a main component to strengthen substitution [7], [8], [9], [10], [11]. In modern cryptography, an S-box is a basic nonlinear component of symmetric key algorithms due to their properties such as nonlinearity, bijective property, differential uniformity, and strict avalanche criterion.

Recently, some other methodologies based on chaos and genetic algorithm are proposed. The main goal of using genetic algorithm is to design S-boxes with high nonlinearity. In [12], Wang designed an S-box based on the chaotic logistic map and tent map and the Traveling Salesman Problem.
In this paper, seven strong S-boxes are obtained by using an algorithm which is divided into two phases. In the first phase, we generate the initial S-box by iterating the chaotic map. In the second phase, we generate a finite number of S-boxes by using genetic algorithm techniques (Crossover and Mutation) and we select the S-boxes having the maximum value of nonlinearity.

The rest of the paper is organized as follows. In the next section we present some important properties of S-boxes briefly. Then, we describe the proposed algorithm in section 3. In Section 4, we analyse the performance of the generated S-boxes and we present a comparison with other S-boxes. Finally, Section 5 encloses the summary and main conclusions of our algorithm.

## II. CRYPTOGRAPHIC PROPERTIES OF S-BOXES

In this section, we present several important properties that can be used to guarantee the cryptographic strength of S-boxes, such as nonlinearity, bijection, the strict avalanche criterion, the output bits independence criterion (BIC)and the equiprobable input/output XOR distribution.

### A. Nonlinearity

Before defining nonlinearity, Hamming weight and Hamming distance need to be identified.

*Definition 1:* The Hamming weight ($H_w$) of a binary vector $v$ is is the number of 1's in $v$.

*Definition 2:* The Hamming distance ($H_d$) between two binary vectors of equal length is the number of places for which the corresponding entries are different.

*Example 1:* The Hamming distance between the two binary vectors $x_1 = (1, 1, 0, 0)$ and $x_2 = (1, 0, 1, 0)$ is 2 since $x_1$ and $x_2$ differ in the second and third positions.
The relationship between Hamming weight and Hamming distance is $H_d(a, b) = H_w(a \oplus b)$.

*Definition 3:* The nonlinearity of a function in the set $B_n$ is defined as the minimum Hamming distance between that function and every linear function in the set.

In general, the nonlinearity of a function $f \in B_n$ is upper bounded by $2^{n-1} - 2^{\frac{n}{2}-1}$. If the S-box is constructed with maximum nonlinearity, it will give a bad approximation by linear functions thus making a cryptosystem difficult to break.

### B. Bijective property

An $n \times n$ S-box is bijective if it has all different output values from interval $[0, 2^n - 1]$.

978-1-4799-7100-8/14/$31.00 ©2014 IEEE

### C. Strict Avalanche Criterion

The Strict Avalanche Criterion (SAC) is presented by Webster and Tavares [13]. The SAC occurs if one input bit $i$ is changed, each output bit will change with probability of one half. SAC requires that if there are any slight changes in the input vector, there will be a significant change in the output vector. The dependence matrix is used to describe the SAC of an S-box. If each element and the mean value of the matrix are both close to the ideal value $0.5$, the S-box is considered as nearly fulfilling the SAC.

### D. Output Bits Independence Criterion

BIC requires that output bits act independently from each other. In other words, there should not be any statistical pattern or statistical dependencies between output bits from the output vectors. An S-box satisfying BIC criteria must be pair-wise independent for a given set of avalanche vectors generated by complementing a single plaintext bit. Let $f_1, f_2, ..., f_n$ denote the boolean functions in the S-box. If the S-box satisfies BIC, $f_j \oplus f_k (j \neq k, 1 \leq j, k \leq n)$ should be highly nonlinear and satisfy the avalanche criterion of $f_j \oplus f_k$. In order to measure the degree of independence between a pair of avalanche variables, we can calculate their correlation coefficient. For two variables $A$ and $B$,

$$\rho(A, B) = \frac{cov(A, B)}{\sigma(A)\sigma(B)} \tag{1}$$

where $\rho(A, B)$ is the correlation coefficient of $A$ and $B$, $cov(A, B)$ is the covariance of $A$ and $B$.

### E. Equiprobable Input/Output XOR Distribution

Biham and Shamir introduced differential cryptanalysis [14], which is based on the use of the imbalances in the input/output XOR distribution table. If an S-box can be close to the equiprobable input/output XOR distribution, it could be immune to the differential attack. The differential approximation probability DP of a given S-box is a measure of differential uniformity. It is defined as:

$$DP(\Delta x \to \Delta y) = \frac{\#\{x \in X \mid S(x) \oplus S(x \oplus \Delta x) = \Delta y\}}{2^m} \tag{2}$$

where $X$ is the set of all possible input values, and $2^m$ is the number of its elements.

## III. The Algorithm

Our proposed algorithm is based on chaotic map and genetic algorithm techniques.

### A. The chaotic maps

The algorithm requires two chaotic maps. The chaotic logistic map to generate the initial S-box and the chaotic Lorenz system to generate the crossover and mutation points.

*1) Chaotic logistic map:* The chaotic logistic map is used to generate a random sequence which will be used to produce the initial S-box (initial population).

$$x_{i+1} = \mu x_i (1 - x_i) \tag{3}$$

where $x_i$, $i = (0, 1, ...)$ is the state value of the chaotic logistic map and $\mu$ is a parameter. We set $\mu = 3.99999$ in order to have good chaotic properties.

*2) Chaotic Lorenz system:* In our algorithm, the Lorenz system will be used to generate the crossover and mutation points. Lorenz system is the first numerical study on chaos. It was developed by Edward Lorenz towards the end of 1950s in order to model the air flow in the atmosphere. Equations of system dynamics are given in (4)

$$\begin{aligned} x_{i+1} &= a(y_i - x_i) \\ y_{i+1} &= bx_i - y_i - x_i z_i \\ z_{i+1} &= x_i y_i - cz_i \end{aligned} \tag{4}$$

States of the system lie in the following intervals: $20 \leq x \leq 20$, $50 \leq y \leq 50$, $50 \leq z \leq 50$. The system has periodic behavior for parameter values $a = 10$, $b = 21$ and $c = 8/3$ and chaotic behavior for parameter values $a = 10$, $b = 28$ and $c = 8/3$.

### B. Genetic algorithm

Genetic algorithms were invented by Holland [15], [16] to mimic some of the processes of natural evolution and selection. They represent an efficient global method for nonlinear optimization problems. The first step is to represent a legal solution to the problem by a string of genes that can take on some value from a specified finite range or alphabet. This string of genes, which represents a solution, is known as a chromosome. Then an initial population of legal chromosomes is constructed at random. At each generation, the fitness of each chromosome in the population is measured. The fitter chromosomes are then selected to produce an offspring for the next generation, which inherits the best characteristics of both parents. After many generations of selection for the fitter chromosomes, the result is hopefully a population that is substantially fitter than the original. All genetic algorithms consist of the following main components:Chromosomal Representation, Initial Population, Fitness Evaluation, Selection, Crossover and Mutation. Algorithm 1 describes the process: In our algorithm, we use the two-point crossover operator as permutation process. The crossover is an operator that mates the two parents (chromosomes) to produce two offsprings, which is illustrated in the following example.

**Algorithm 1** Genetic Algorithm()

```
Generate random population
repeat
  Evaluate fitness of current population
  Select chromosomes, based on fitness, for reproduction
  Perform crossover and mutation to give new improved
  population
until finished
```

*Example 2:* Suppose that Parent1= 011 <u>101</u> 0101, Parent2= 100 <u>111</u> 0111, the first crossover point is 3 and the second crossover point is 6. After the crossover process: Child1=011 <u>111</u> 0101 and Child2= 100 <u>101</u> 0111.

The two crossover points are defined by using the chaotic Lorenz sequences.

### C. Proposed S-box generation method

The proposed method is performed in 3 steps. These are as follows:

*a) **Step 1 (Generating the initial S-box):**:* This step is explained in detail below. Algorithm 2 describes how to generate the initial S-box.

1) Define $S$ as a sequence, which is empty at the beginning.
2) Given the initial value $x_0$, iterate Eq. (3) for 100 times to get rid of the transient effect.
3) Continue to iterate, one time, the Eq. (3) , and denote the current state value as $x'$. Then an integer value $X$ is obtained as below:
$$X = floor(256 \times x') \tag{5}$$
where floor(X) rounds the elements of X to the nearest integers towards minus infinity.
4) If $X$ is not in sequence $S$, append it to $S$. Otherwise, go to item 3.
5) If the number of elements in $S$ is not bigger than 256, go to item 3. Otherwise, output $S$.
6) Construct the initial $(8 \times 8)$ S-box from the sequence $S$ using the algorithm 3. This S-box is used as the initial population.

**Algorithm 2** $Gen\_init\_SBox(x_0)$

```
Input: x_0
Output: SBox
  i ← 1
  while i ≤ 100 do
    iterate_logistic_map
  end while
  nb ← 0
  while nb < 256 do
    x' ← iterate_logistic_map
    X ← floor(256 * x')
    if X ∉ S then
      S(nb) ← X
      nb ← nb + 1
    end if
  end while
  SBox ← S2Sbox(S)
```

**Algorithm 3** $S2Sbox(S)$

```
Input: S
Output: SBox
  i ← 1
  k ← 1
  while i ≤ 16 do
    j ← 1
    while j ≤ 16 do
      SBox(i, j) ← S(k)
      k ← k + 1
      j ← j + 1
    end while
    i ← i + 1
  end while
```

*b) **Step 2 (generate the control parameters):**:* The chaotic Lorenz system defined in Eq. (4) is used to generate the control parameters of the genetic algorithm. This step is explained in detail below.

1) Given the initial values $x_0, y_0$ and $z_0$, iterate Eq. (4) for 100 times to get rid of the transient effect.
2) Continue to iterate it 16 times to generate three sequences, $S1_i = x_i$, $S2_i = y_i$ and $S3_i = z_i$, $i = 1, 2, ..., 16$. The first sequence is to get two crossover points $(prow_1, prow_2)$ for shuffling rows, the second, is to get two crossover points $(pcol_1, pcol_2)$ for shuffling columns and the third, is to get two mutation points $(pmut_1, pmut_2)$ for permutating values in the S-box. The six points are generated by the following.

$$prow1_i = mod(floor(S1_i * 10^{14}), 4) + 2 \tag{6}$$
$$prow2_i = mod(floor(S1_i * 10^{14}), 4) + 8 \tag{7}$$

$$pcol1_i = mod(floor(S2_i * 10^{14}), 4) + 2 \tag{8}$$
$$pcol2_i = mod(floor(S2_i * 10^{14}), 4) + 8 \tag{9}$$

$$pmut1_i = mod(floor(S3_i * 10^{14}), 4) + 2 \tag{10}$$
$$pmut2_i = mod(floor(S3_i * 10^{14}), 4) + 8 \tag{11}$$

*c) **Step 3 (Applying genetic algorithm techniques to generate a maximal S-box nonlinearity)**:*

1) calculate the nonlinearity, $nl_0$ of the initial S-box (S-box$x_0$)
2) set $nl_{max} = nl_0$, $i = 1$ and $j = 0$
3) crossover and mutate the S-box as follows.
   - using the two points $prow1_i \in [2...5]$ and $prow2_i \in [8...11]$, crossover the row number $i$ and the row number $(16 - i + 1)$, $i = 1, 2, ..., 16$.
   - using the two points $pcol1_i \in [2...5]$ and $pcol2_i \in [8...11]$, crossover the column number $i$ and the column number $(16 - i + 1)$, $i = 1, 2, ..., 16$.
   - permute the two points $pmut1_i \in [2...5]$ and $pmut2_i \in [8...11]$ in all lines of the S-box.
   - the new S-box is named S-box$x'_i$, $i = 1, 2, 3...n$, $n$ is the number of iterations.
4) calculate the nonlinearity $nl_i$ of S-box$x'_i$.
5) if $nl_i > nl_{max}$ then set: $nl_{max} = nl_i$, $j = j + 1$ and S-box$x_j$=S-box$x'_i$ . If $i \leq n$ then set $i = i + 1$ and go to 3 else go to 6.
6) after iterating 3, $n$ times, we get $j$ S-boxes having a maximal nonlinearity.

### *D. Generated S-boxes*

Set the initial value of the logistic map $x_0 = 0.2$ and the initial values of the Lorenz system $x_0 = 10.1$, $y_0 = 6.21$ and $z_0 = 20.38$. After generating $10^5$ S-boxes according to the above-mentioned method, we have obtained seven S-boxes (Tables I-II-III-IV-V-VI-VII) with mean values of nonlinearities greater than or equal to 107 which is a high value of nonlinearity. The comparison of performance with other S-boxes is shown in Table X. As demonstrated, the generated S-boxes exhibit a higher nonlinearity value compared with other S-boxes. As for the SAC, the optimum value is 0.5. We note that the average values of SAC of the generated S-boxes are very close to 0.5.

TABLE I. S-BOX1 IN THE 594TH GENERATION

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

TABLE II. S-BOX2 IN THE 4166TH GENERATION

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

TABLE III. S-BOX3 IN THE 4748TH GENERATION

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

TABLE IV. S-BOX4 IN THE 7107TH GENERATION

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

TABLE V. S-BOX5 IN THE 49230TH GENERATION

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

TABLE VI. S-BOX6 IN THE 49465TH GENERATION

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

TABLE VII. S-BOX7 IN THE 53072ND GENERATION

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

## IV. PERFORMANCE ANALYSIS OF THE GENERATED S-BOXES

We choose five properties that are necessary for general cryptographically strong S-boxes. They include the bijective property, the nonlinearity property, the strict avalanche criterion (SAC), the output bits independence criterion (BIC), and the equiprobable input/output XOR distribution. The comparison of performance with other S-boxes is shown in Table X.

### A. Bijectivity of the generated S-boxes

All generated S-boxes have different output values from interval [0;255], so they satisfy the requirement of bijectivity.

### B. Nonlinearities of the generated S-boxes

The nonlinearities of the generated S-boxes are shown in Table VIII. It is noticed that all the average values are greater than or equal to 107.

### C. SAC of generated S-boxes

The dependence matrix is used to describe the SAC of an S-box. The matrix of the generated S-box7 can be found in Table IX. The mean value of the dependence matrix of S-box7 is 0.4971 which is very close to the ideal value 0.5. The mean values of the dependence matrices of the generated S-boxes are listed in Table VIII.

### D. BIC of the generated S-box

Here, we limit the analysis of this criterion to S-box7. The other 6 S-boxes have the same values. The mean value of nonlinearities of S-box7 (Table XI) is 103.8571, and the mean value of dependence matrix (Table XII) is 0.5034, which

TABLE VIII. NONLINEARITY AND SAC OF THE GENERATED S-BOXES

| S-Box | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Avg. Nonlinearity | Avg. SAC |
|---|---|---|---|---|---|---|---|---|---|---|
| S-Box1 | 106 | 108 | 108 | 108 | 108 | 106 | 104 | 110 | 107.25 | 0.5046 |
| S-Box2 | 108 | 106 | 108 | 106 | 108 | 108 | 106 | 106 | 107 | 0.4993 |
| S-Box3 | 106 | 108 | 106 | 106 | 108 | 108 | 108 | 106 | 107 | 0.5078 |
| S-Box4 | 108 | 104 | 108 | 108 | 106 | 108 | 108 | 108 | 107.25 | 0.5100 |
| S-Box5 | 106 | 108 | 108 | 106 | 106 | 108 | 108 | 106 | 107 | 0.5032 |
| S-Box6 | 106 | 106 | 108 | 108 | 106 | 108 | 106 | 108 | 107 | 0.4951 |
| S-Box7 | 106 | 108 | 110 | 106 | 110 | 106 | 106 | 108 | 107.5 | 0.4971 |

TABLE IX. THE DEPENDENCE MATRIX OF S-BOX7

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.46875 | 0.5 | 0.4375 | 0.46875 | 0.48438 | 0.48438 | 0.48438 | 0.48438 |
| 0.46875 | 0.57813 | 0.46875 | 0.53125 | 0.53125 | 0.48438 | 0.51563 | 0.51563 |
| 0.51563 | 0.53125 | 0.51563 | 0.4375 | 0.46875 | 0.51563 | 0.48438 | 0.54688 |
| 0.48438 | 0.53125 | 0.57813 | 0.53125 | 0.48438 | 0.53125 | 0.5 | 0.48438 |
| 0.54688 | 0.48438 | 0.5625 | 0.5 | 0.59375 | 0.45313 | 0.46875 | 0.46875 |
| 0.51563 | 0.48438 | 0.5 | 0.54688 | 0.46875 | 0.5 | 0.5 | 0.46875 |
| 0.40625 | 0.48438 | 0.5 | 0.45313 | 0.53125 | 0.51563 | 0.51563 | 0.46875 |
| 0.48438 | 0.40625 | 0.51563 | 0.48438 | 0.53125 | 0.48438 | 0.46875 | 0.45313 |

TABLE X. COMPARISON OF PERFORMANCE

| S-Box | Avg. Nonlinearity | Avg. SAC | BIC-SAC | BIC-Nonlinearity | Maximum I/O XOR |
|---|---|---|---|---|---|
| S-Box7 | 107.5 | 0.4971 | 0.5034 | 103.8571 | 10 |
| Ref.[2] | 105 | 0.4971 | 0.4999 | 102.96 | 10 |
| Ref.[4] | 103.25 | 0.5049 | 0.5007 | 103.82 | 10 |
| Ref.[17] | 103.5 | 0.4939 | 0.4992 | 103.64 | 10 |
| Ref.[12] | 108 | 0.5068 | 0.5017 | 103.36 | 10 |

indicates that all the S-boxes fulfill the requirement of BIC property.

TABLE XI. BIC-NONLINEARITY CRITERION FOR S-BOX7

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| - | 106 | 104 | 106 | 106 | 100 | 102 | 106 |
| 106 | - | 106 | 102 | 104 | 104 | 102 | 104 |
| 104 | 106 | - | 104 | 104 | 104 | 106 | 106 |
| 106 | 102 | 104 | - | 104 | 106 | 104 | 106 |
| 106 | 104 | 104 | 104 | - | 104 | 104 | 106 |
| 100 | 104 | 104 | 106 | 104 | - | 102 | 100 |
| 102 | 102 | 106 | 104 | 104 | 102 | - | 96 |
| 106 | 104 | 106 | 106 | 106 | 100 | 96 | - |

TABLE XII. BIC-SAC CRITERION FOR S-BOX7

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| - | 0.496 | 0.500 | 0.501 | 0.509 | 0.501 | 0.482 | 0.505 |
| 0.496 | - | 0.529 | 0.511 | 0.490 | 0.498 | 0.509 | 0.513 |
| 0.500 | 0.529 | - | 0.501 | 0.505 | 0.505 | 0.523 | 0.501 |
| 0.501 | 0.511 | 0.501 | - | 0.505 | 0.515 | 0.509 | 0.523 |
| 0.509 | 0.490 | 0.505 | 0.505 | - | 0.482 | 0.484 | 0.474 |
| 0.501 | 0.498 | 0.505 | 0.515 | 0.482 | - | 0.511 | 0.515 |
| 0.482 | 0.509 | 0.523 | 0.509 | 0.484 | 0.511 | - | 0.484 |
| 0.505 | 0.513 | 0.501 | 0.523 | 0.474 | 0.515 | 0.484 | - |

## V. CONCLUSION

A method for obtaining cryptographically strong S-boxes based on Chaotic map and genetic algorithm is presented. This method took advantage of the properties of chaotic functions and genetic algorithm. Genetic algorithm techniques are used, mainly, to produce a high nonlinearity value. Since the goal of our algorithm was to generate S-boxes with a high nonlinearity value, this does not mean that the other criteria are not fulfilled. The results of numerical analysis of the seven S-boxes generated by our algorithm have shown that all the criteria for a good S-box are approximately fulfilled and they have a high immunity to resist the differential cryptanalysis.
In future work, we intend to use the generated S-boxes in developing an image encryption algorithm.

## REFERENCES


[1] Jakimoski G, Kocarev L. Chaos and cryptography: block encryption ciphers based on chaotic maps. IEEE Trans Circuits Syst I ;48:163 (2001).

[2] Tang G, Liao XF, Chen Y. A novel method for designing S-boxes based on chaotic maps. Chaos Solitons Fractals ;23:413-419 (2005).

[3] Chen G, Chen Y, Liao XF. An extended method for obtaining S-boxes based on 3-dimensional chaotic baker maps. Chaos Solitons Fractals ;31:571-579 (2007).

[4] Ozkaynak F, Ozer AB. A method for designing strong S-boxes based on chaotic Lorenz system. Phys Lett A ;374:3733-3738. (2010)

[5] Hussain, I., Shah, T., Gondal, M.A.: A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm. Nonlinear Dyn. 70 , 1791–1794 (2012).

[6] Zaibi, G., Peyrard, F., Kachouri, A., Fournier-Prunaret, D., Samet, M.: Efficient and secure chaotic S-Box for wireless sensor network. Security Comm. Networks. DOI: 10.1002/sec.728 (2013).

[7] Hussain, I., Shah, T., Gondal, M.A.: Image encryption algorithm based on PGL(2,GF($2^8$)) S-boxes and TD-ERCS chaotic sequence. Nonlinear Dyn. (2012). doi:10.1007/ s11071-012-0440-0

[8] Wang, X., Teng, L.: An image blocks encryption algorithm based on spatiotemporal chaos. Nonlinear Dyn. 67(1), 365371 (2012).

[9] Ye, G., Wong, K.W.: An image encryption scheme based on time-delay and hyperchaotic system. Nonlinear Dyn. 71(12), 259267 (2013).

[10] Zhang, X., Zhao, Z.: Chaos-based image encryption with total shuffling and bidirectional diffusion. Nonlinear Dyn. 75(12), 319330 (2014).

[11] Wang, X., Wang, Q.: A novel image encryption algorithm based on dynamic S-boxes constructed by chaos. Nonlinear Dyn.75(3), 567576 (2014).

[12] Wang, Y., Wong, K.W., Li, Wong., Li, Y.: A novel method to design

S-box based on chaotic map and genetic algorithm. Physics Letters A. 376, 827-833 (2012).

[13] Webster, A., Tavares, S.: On the design of S-boxes. In: Advances in cryptology: proc of CRYPTO85. Lecture notes in computer science; 523-34 (1986).

[14] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. J Cryptol. 4:372 (1991)

[15] Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press. 183,(1975)

[16] Holland, J.H.: Genetic algorithms. Scientific American, v.267, no. l, p. 44–50 (1992).

[17] Asim, M., Jeoti, V.: Efficient and simple method for designing chaotic Sboxes. ETRI J. x1, 170-172 (2008).

[18] F, Ozkaynak., S, Yavuz.:Designing chaotic S-boxes based on time-delay chaotic system. Nonlinear Dyn.74(3), 551557 (2013).

[19] H, Liu., A, Kadir., Y, Niud.: Chaos-based color image block encryption scheme using S-box. Int J Electron Commun (AEU), Volume 68, Issue 7,676–686 (2014).

[20] D, Lambi.: A novel method of S-box design based on chaotic map and composition method. Chaos, Solitons and Fractals (58) 16-21 (2014).

[21] A. Kanso, M. Ghebleh, A novel image encryption algorithm based on a 3D chaotic map. Commun Nonlinear Sci Numer Simulat (2012);17:2943-59.

[22] Solak E, okal C. Comment on encryption and decryption of images with chaotic map lattices. Chaos Interdiscip J Nonlinear Sci 2008;18(3). Art No. 03810.

[23] Arroyo D, Rhouma R, Alvarez G, Li S, Fernandez V. On the security of a new image encryption scheme based on chaotic map lattices. Chaos Interdiscip J Nonlinear Sci (2008);18. Art No. 033112.

[24] E.Solak,C.Cokal, O.T.Yildiz,T.Biyikoglu, Cryptanalysis of Fridrichs chaotic image encryption,International Journal of Bifurcation and Chaos 20(5)(2010)1405-1413.

[25] C. Li, D. Arroyo, K.-T. Lo, Breaking a chaotic cryptographic scheme based on composition maps, International Journal of Bifurcation and Chaos 20 (8) (2010) 2561–2568.

[26] E.Solak,C.Cokal, Algebraic break of image ciphers based on discretized chaotic map lattices,Information Sciences 181 (1) (2011) 227-233.

[27] D.Arroyo,G.Alvarez,J.M.Amigo, S.Li,Cryptanalysis of a family of self-synchronizing chaotic stream ciphers,Communications in Nonlinear Science and Numerical Simulation 16 (2) (2011) 805-813.

[28] Li, C., Zhang, L.Y., Wong, R.O., Shu, S.: Breaking a novel colour image encryption algorithm based on chaos. Nonlinear Dyn.70 , 2383–2388 (2012)

[29] Wang, X., Liu, L.: Cryptanalysis of a parallel sub-image encryption method with high-dimensional chaos. Nonlinear Dyn.73 , 795-800 (2013)

[30] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, International journal of Bifurcation and Chaos 8 (1998) 1259–1284.

[31] X.J. Tong, The novel bilateral–Diffusion image encryption algorithm with dynamical compound chaos, The Journal of Systems and Software 85 (2012) 850-858

[32] C. Li, S. Li, G. Chen, W.A. Halang, Cryptanalysis of an image encryption scheme based on a compound chaotic sequence Image and Vision Computing 27 (2009) 1035-1039

[33] D. Stinson, Cryptography: Theory and Practice, CRC Press, 1995.