A new approach to design S-box generation algorithm based on genetic algorithm

Ünal Çavuşoğlu*

Department of Software Engineering,
Sakarya University,
Serdivan, Sakarya 54187, Turkey
Email: unalc@sakarya.edu.tr
*Corresponding author

Abdullah Hulusi Kökçam

Department of Industrial Engineering,
Sakarya University,
Serdivan, Sakarya 54187, Turkey
Email: akokcam@sakarya.edu.tr

Abstract: Substitution box (S-box) is one of the most important structures used for byte change operation in block encryption algorithms. An S-box structure with strong cryptological properties makes the encryption algorithm much more resistant to attacks. In this article, a powerful S-box generation algorithm design is presented using genetic algorithm (GA). In the GA-based S-box generation algorithm, the nonlinearity value which is one of the most important S-box evaluation criteria, has been processed. Quality of the generated S-boxes is determined by performance tests. Obtained performance results are compared with the S-boxes in the literature. It has been found that the presented algorithm generates S-boxes with strong cryptological properties.

Keywords: genetic algorithms; information security; nonlinearity; substitution box; S-box.

Reference to this paper should be made as follows: Çavuşoğlu, Ü. and Kökçam, A.H. (2021) 'A new approach to design S-box generation algorithm based on genetic algorithm', *Int. J. Bio-Inspired Computation*, Vol. 17, No. 1, pp.52–62.

Biographical notes: Ünal Çavuşoğlu received his BSc in Computer Engineering from the Yıldız Technical University, Istanbul, Turkey, MSc in Computer Engineering in 2014 and PhD in Computer Engineering in 2017 from the Sakarya University, Sakarya, Turkey. He is currently an Associate Professor at the University of Sakarya since 2020. His main research interests are in the fields of computer and communication networks, cryptology and information security, chaotic system and applications.

Abdullah Hulusi Kökçam received his BSc from the Industrial Engineering Department, Selcuk University in 2008 and MSc from the Industrial Engineering Department, Sakarya University in 2010. He received his PhD from the Department of Industrial Engineering, Sakarya University, in 2017. He is currently working as a research assistant at the Industrial Engineering Department, Sakarya University. He has been worked on fuzzy project scheduling with meta-heuristic methods in his MSc thesis and an alternative model for assessing school success rate in higher education entrance exam in his PhD thesis. His research area of interests are project scheduling, meta-heuristic methods, optimisation, test measurement, and school performance.

1 Introduction

Importance of information security increases day by day. Different encryption designs are implemented for the protection of the data. Block cipher algorithms encrypt data in blocks of different sizes. One of the most basic structure used in block encryption algorithm is the substitution box (S-box). S-boxes are widely used in designing of many encryption algorithms such as AES (Gladman, 2001), DES (Davies, 1983) and Skipjack (Brickell et al., 1995). S-box performs the mixing process in the encryption algorithm. An S-box structure with strong cryptological features makes the encryption process very resistant to attacks. In the S-box design of the AES algorithm, powerful S-boxes are generated using the finite field theorem. However, different approaches are presented in the literature for generating

Copyright © 2021 Inderscience Enterprises Ltd.

S-boxes using less processing power. The main purpose of these studies is to generate powerful S-box structures. According to literature S-box generation algorithms (GAs) and S-box-based encryption algorithms are developed using different approaches such as chaotic systems. There are also hybrid designs where the chaotic system and genetic algorithm are used together.

Chaotic systems meet the confusion and diffusion properties of the basic requirements of encryption processes because of their strong random dynamics and sensitive dependence on the initial conditions and system parameters. Due to the characteristics of the chaotic systems, it is found that there is a close relationship between chaos and cryptology science (Amigo et al., 2007; Jakimoski and Kocarev, 2001). Peng et al. (2012) and Zaibi et al. (2014) presented a new dynamic chaotic-based S-box GA to be used in block encryption algorithms. Tang and Liao (2005) and Tang et al. (2005) offered new S-box GAs that use different approaches in their work. Liu et al. (2014) realised a encryption algorithm that use chaos-based S-box GA generated by using Chen chaotic map. In addition to all these, there are S-box studies in the literature that are produced using many different approaches (Khan et al., 2012; Özkaynak and Özer, 2010; Chen, 2008; Özkaynak and Yavuz, 2013; Chen et al., 2007; Wang et al., 2009; Hussain et al., 2012).

Genetic algorithm is a meta-heuristic method that is mostly used to solve problems in NP-hard class in which cannot be solved in a reasonable amount of time. Genetic algorithm is a robust adaptive numerical optimisation method based on natural selection process which can be applied to wide variety of problems. Natural selection is applied as good solutions are most likely to be selected for generating next population. It differs from other methods whereas most of them work on a single solution, GA uses population of solution. Selection, crossover and mutation operators are three basic components of GA which promotes to find near-optimal solutions by avoiding local optimal solutions and concentrating on global optimal solution (Coley, 1999).

There are many studies with GA in distinct areas such as image processing (Singh et al., 2017), medicine (Akbar et al., 2017), spacecraft trajectories (Fung et al., 2017), analysis of time series (Messias et al., 2016), solid state physics (Dong et al., 2017), aeronautics (Renzi, 2016), robotics (Montazeri et al., 2017), scheduling (Demir and Erden, 2017), face recognition (Sukhija et al., 2016), vehicle routing (Hiassat et al., 2017), facility layout (Paes et al., 2017), and software testing (Khan et al., 2017). GA is also used in network intrusion detection systems (Li, 2004), misuse detection systems (Banković et al., 2007), network security events analysis (Liu et al., 2016), network security and quality of service optimisation (Zhao et al., 2016). Pareek and Patidar (2016) used GA to protect medical images which are send through communication networks. Mishra et al. (2016) encrypted secret messages and hid in cover image using Z-transform and GA. Kumar and Chatterjee (2016) used GA to generate keys for stream ciphers. Shankar and Eswaran (2016) used GA to generate private key for decrypting an image, which is encrypted with elliptic curve cryptography technique. Picek et al. (2015) proposed a new method to generate different sized S-boxes with cartesian genetic programming approach. Picek et al. (2017) suggested new S-boxes using cellular automata rules having strong cryptographic properties and low application cost. Picek et al. (2016) presented a new cost function to evaluate S-boxes. Kapuściński et al. (2016, 2017) studied on invertible S-boxes and compared two multi-objective GA which are non-dominated sorting genetic algorithm II (NSGA-II) and its steady-state version. Ivanov et al. (2016) used GA in a reverse way to generate large number of strong bijective S-boxes. Wang et al. (2012) proposed a method for S-box generation based on chaotic map and GA. Kalaiselvi and Kumar (2016) suggested enhanced AES cryptosystem using GA in substitution-permutation boxes (SP-Boxes) and implementing nonlinear neural network in substitution-permutation network.

The aim of this study, unlike the S-box GAs in the literature, only the functions of the genetic algorithm are utilised, and a simple GA with strong cryptographic properties are presented using the features of the genetic algorithm. Table 2 shows that the S-box structure of the AES algorithm has the best values. In this study an algorithm is designed that generates S-boxes with the performance values closest to the AES algorithm and got better performance values from studies in the literature which uses different methods.

Using a multi-objective approach in the developed genetic algorithm-based S-box generation algorithm does not increase the benefit for the generation of S-boxes with higher security. From this point of view, it is shown with the proposed algorithm in the article that S-box structures with strong cryptological features can be generated by making use of the features of the genetic algorithm with a single objective. S-boxes with high nonlinearity values were obtained as a result of the generation made by the genetic algorithm-based algorithm in a huge solution space. The main contribution of this article to literature is to show that genetic algorithms are an effective method to be used in S-box generation, and they have been shown to have better performance compared to the S-boxes generated by other techniques in the literature.

The remaining sections are as follows: information about S-boxes and performance analyses are given in Section 2. In Section 3, proposed genetic algorithm-based S-box GA is introduced in detail. In Section 4, performance tests of generated S-boxes are carried out and compared with literature. Evaluation of the study is given in Section 5.

2 The performance tests of S-box

2.1 Nonlinearity

Nonlinearity (Adams and Tavares, 1990) is one of the most important S-box performance criteria. In computing the nonlinearity value of the S-box is calculated using the following equation. This calculation is used using boolean function and affine transformation operations.

$$N(f) = 2^{n-1}\left(1 - 2^{-n}\max_{\omega \in GF(2^n)}|S_f(\omega)|\right) \quad (1)$$

Walsh spectrum analysis is used for nonlinearity calculation. The formula for the Walsh spectrum analysis is given below. In this formula, $\omega \in GF(2^n)$ and $x.\omega$ is the result of inner product operation of x and ω.

$$S_f(\omega) = \sum_{\omega \in GF(2^n)} (-1)^{f(x)\oplus x.\omega} \quad (2)$$

2.2 Strict avalanche criterion

Strict avalanche criteria (SAC) (Webster and Tavares, 1985) is an evaluation criterion in which the probability of changing output bits is calculated according to the variation on the input bits. In this method, the calculated value expresses the possibility of the half change on the output bits when an input bit changes. The optimum value is 0.5 in the SAC test. If the obtained value is close to 0.5, it indicates that the produced S-box has a good SAC value.

2.3 Output bits independence criterion

The method of independence of output bits is one of the most important in the S-box performance criteria suggested by Webster and Tavares (1985). Using bits independence criterion (BIC) method, the cluster of vectors produced by inverting a bit of plaintext is tested independently of all pairs of avalanche variants. Correlation coefficient value is used to determine the relation between the avalanche variable pairs. $f_j \oplus f_k (j \neq k, 1 \leq j, k \leq n)$ permit strict avalanche nonlinearity performance criteria, if the Boolean functions of two output bits f_i and f_k permit BIC. The correlation coefficient value is computed to determine independence rate between independent pairs of variable pairs. Equation (3) gives the formula to calculate the correlation coefficient value.

$$\rho\{A, B\} = \frac{\{A, B\}}{\sigma\{A\}\sigma\{B\}} \quad (3)$$

where ρ is correlation coefficient value, cov means covariance value and two random avalanche variable pairs A, B and standard deviations $\sigma\{A\}\sigma\{B\}$ values.

2.4 Differential approximation probability

The differential approximation probability (DP) method is known as the differential cryptanalysis method proposed by Biham and Shamir (1991), where the XOR distribution balance between the S-box input and output bits is determined. It is expected that each output will have an equal probability when compared with XOR value input values. If the XOR distribution probability values between the calculated input and output bits are close, this indicates that the S-box is resistant to differential attack. Equation (4) gives the formula used to calculate the DP value.

$$DP_f = \max_{\Delta_x \neq 0, \Delta_y}(\#x \in X, f_x \oplus f(x \oplus \Delta_x) = \Delta_y/2^n) \quad (4)$$

where

$2^n \rightarrow$ the cardinality of all possible input values (x)

$\Delta_x \rightarrow$ the input differantial

$\Delta_y \rightarrow$ the output differantial.

In equation (4), Δ_x and Δ_y represent the calculated input and output differential values. Δ_x represents the input differential for each element. Each of these Δ_x inputs must correspond to the Δ_y output differential value. The calculation of these values determines the difference in the output values of the changes made to the input values.

2.5 Linear approximation probability

The linear approximation probability LP is an evaluation criterion that measures the maximum value of an imbalance in a case. In the linear cryptanalysis, the probability of linearity between input and output values is investigated on the sub-byte operation on the S-box. The linear relationship between the data before the sub-byte operation and the data to be encrypted is utilised. In this process, the deviation is calculated by comparing with all possible key bits. Linear approximation table (LAT) are constructed for LP analysis. The largest value in the linear approach table specifies the complexity of the linear probability. According to the formula given by Matsui (1993), the linear approximation probablity value is calculated using the following formula.

$$LP = \max_{r_x, r_y \neq 0}\left|\frac{\{x \epsilon X \mid x.r_x = S(x).r_y\}}{2^m} - \frac{1}{2}\right| \quad (5)$$

The parity value of the selected input bits specified by the r_x value equals the output bits of the r_y value. r_x and r_y input and output values and X represents all possible input values in equation.

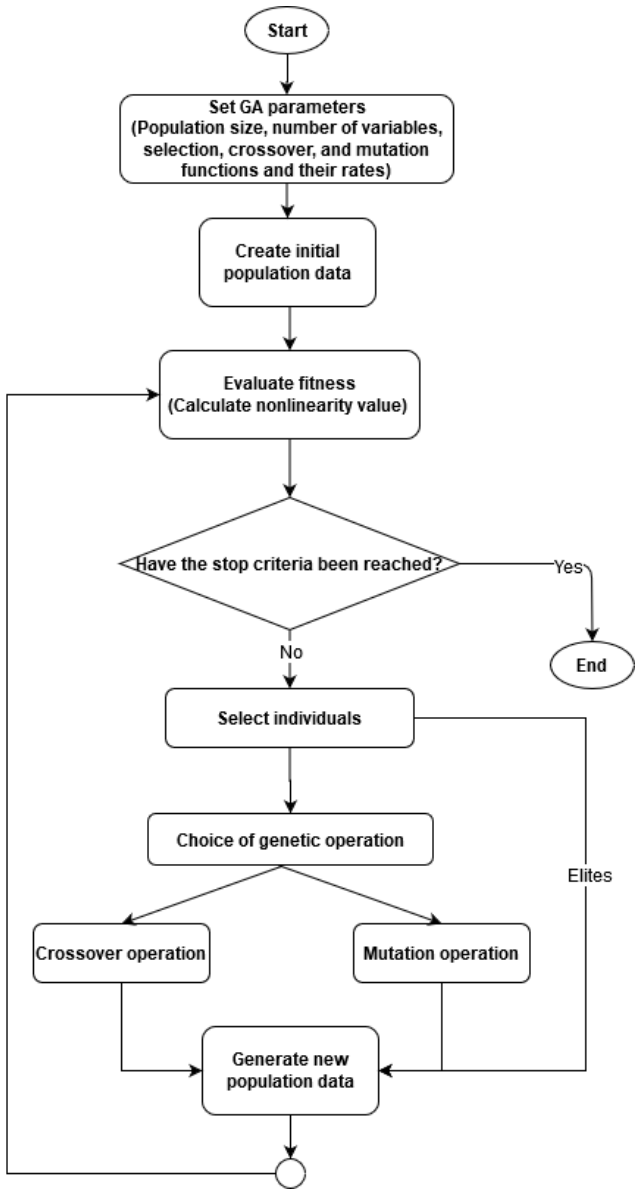
3 The new S-box GA design

In this section, a genetic algorithm-based S-box GA is designed to generate S-boxes with strong cryptographic properties. By using GA in S-box generation, the required computational time is reduced, thus providing stronger encryption against timing attacks (Kalaiselvi and Kumar, 2016). MATLAB GA solver is used in the application.The genetic algorithm initially generates a random initial population. The size of the population directly affects the

duration of the process. As the population size increases, the time of obtaining results increases proportionally. In the case of very low population use, sufficient diversity cannot be achieved and the solution cannot be improved.

New populations are generated using the previous populations. In the generation of new populations, the fitness value for the current population is calculated and the individuals are selected. Best individuals which are called elites are transferred to the next population without any change. Other individuals in the population are undergo crossover and mutation operations to become a member of a new population. Crossover operator is used to transfer the characteristics of the good individuals to the next generation. On the other hand mutation operator is used to avoid from stuck on to the local maximum.

Figure 1 Proposed S-box GA block diagram



The algorithm continues to work until the determined stopping criterion is met. Block diagram of suggested genetic algorithm-based S-box GA is shown in Figure 1.

In this algorithm, GA is used to search 256-bit arrays with high nonlinearity value (minx > 106 and max =< 112) for S-boxes. Because of the similarity of the problem to the well-known traveling salesman problem (TSP) structure, the problem is solved like a TSP problem. In TSP, a salesman has to travel all of the places provided to him/her with the objective of shortest distance possible, never visiting a place more than once. Here, instead of the shortest distance, the fitness function is changed to search for a solution with high nonlinearity value. Nonlinearity value is one of the most important criteria used in S-box evaluation. The structure of the chromosomes consist of 256 unique genes, which are different from each other. In the solution process, crossover and mutation operators of GA is implemented to reach the global optimum by avoiding local optimums. The proposed GA works as:

1 The algorithm first generates the initial population (100 individuals) with random permutations.

2 Next population is produced using the individuals in the current population until the stopping criteria is met. The following steps are being implemented while creating a new population:

   a Fitness values (nonlinearity value) of all individuals in the current population are calculated using fitness function.

   b Calculated raw fitness values are scaled to a more useful number range.

   c Parents are selected from current population based on their scaled fitness scores and processed to generate new population.

   - Some of high scored parents (elites) are selected and directly transferred to next population.
   - Rest of the parents are subject to crossover or mutation operation and resulting children transferred to next population.

3.1 Chromosome structure

A chromosome consist of 256 unique genes, which represents the values of an S-box as given in Figure 2.

Figure 2 Chromosome structure

X_1	X_2	X_3	...	X_{255}	X_{256}

3.2 Fitness function

Fitness function finds the nonlinearity value of an individual (chromosome) using equations (1) and (2). Fitness of all individuals are found through calculating their nonlinearity values. Objective is to find highest nonlinearity valued individual in which represents an S-box.

Table 1 Properties of genetic operators and fitness functions in S-box studies with genetic algorithm

Study	*Crossover*	*Mutation*	*Fitness*
Picek et al. (2014)	Partially mapped, position-based and order crossovers	Insert, inversion and swap mutations	Nonlinearity value and transparency order
Picek et al. (2015)	Not revealed	Not revealed	Nonlinearity value and differential uniformity
Kalaiselvi and Kumar (2016)	Randomly apply single point, two point and uniform crossover	Flip chromosomes to generate new one	Not revealed
Ivanov et al. (2016)	Choose pair of parents and select two random points on them to generate pair of children	Correct bijective property of crossover children by checking for element repetition	Calculate nonlinearity and differential uniformity values
Picek et al. (2017)	Exchange randomly selected branches between two parent trees	Replace the branch with randomly generated one	First check function for balance with Kronecker delta function, if its balanced then calculate nonlinearity value and differential uniformity
Our study	Reverse array between two randomly selected points	Swap two randomly selected genes	Calculate nonlinearity value

Table 2 The comparison table of S-boxes

S-box	*Nonlinearity*			*BIC-SAC*	*BIC-nonlinearity*	*SAC*			*DP*	*LP*
	Min	*Avg*	*Max*			*Min*	*Avg*	*Max*		
Proposed S-box1	105	107	110	0.5011	102.75	0.4375	0.5197	0.625	10	0.132
Proposed S-box2	105	106	110	0.5039	104.8	0.4218	0.5061	0.6406	10	0.125
Proposed S-box3	102	106	112	0.5031	104.7	0.3906	0.4960	0.5781	12	0.125
Chen (2008)	102	104	106	0.4971	103.2	0.3750	0.4980	0.6093	10	0.136
Khan et al. (2012)	95	102	107	0.5011	100.28	0.3906	0.5034	0.6250	12	0.136
Tang and Liao (2005)	101	103.8	108	0.4958	102.6	0.3906	0.5058	0.5781	14	0.140
Khan et al. (2013)	98	104	108	0.5048	102.857	0.2812	0.4953	0.6093	12	0.140
Hussain et al. (2012)	102	104	108	0.5021	104.071	0.3906	0.5056	0.5937	12	0.125
Özkaynak and Özer (2010)	100	103	106	0.5009	103.714	0.4218	0.5048	0.5937	10	0.125
Wang et al. (2009)	102	104	106	0.5070	103.8	0.4850	0.5072	0.5150	12	0.136
Çavuşoğlu et al. (2017)	104	106	108	0.49763	103.857	0.3906	0.5063	0.5937	12	0.164
Brickell et al. (1995)	104	105.7	108	0.4994	104.1	0.3986	0.5032	0.5938	12	0.109
Gladman (2001)	112	112	112	0.5046	112	0.4531	0.5048	0.5625	4	0.062

Table 3 Proposed S-box1

144	136	240	229	79	231	110	69	215	238	101	120	239	153	163	82
127	90	148	247	246	205	171	162	24	27	250	142	253	6	113	68
98	86	243	220	201	200	222	248	203	32	177	43	132	146	202	37
60	130	172	157	121	181	88	85	105	44	251	252	235	1	76	92
193	20	91	180	34	198	213	187	145	176	38	219	109	191	35	161
94	190	30	10	16	131	112	63	51	3	61	15	46	134	218	9
167	233	40	126	223	84	216	103	182	87	115	125	122	114	192	102
2	39	57	209	225	25	129	111	81	169	26	75	230	155	212	139
78	158	116	54	71	185	186	11	18	106	149	236	227	204	22	242
4	108	107	96	226	245	70	33	21	138	164	104	52	150	58	49
140	151	89	67	210	64	74	194	241	199	166	156	80	255	228	188
12	118	45	159	56	207	141	256	128	13	73	197	133	95	14	42
53	99	214	47	217	48	23	77	72	123	160	17	170	168	135	237
55	147	208	221	174	196	28	195	65	224	211	83	59	29	93	143
119	152	50	254	19	100	178	234	36	8	183	137	232	206	184	7
154	117	5	41	189	244	165	124	249	179	97	175	31	173	62	66

3.3 Scaling

In the scaling process, ranking is done by ordering the individual scores. Accordingly the scaled fitness score of an individual in the r^{th} order is calculated by the formula $1/\sqrt{r}$.

Table 4 Proposed S-box2

179	39	164	119	118	223	232	25	106	8	197	92	127	12	153	125
3	24	204	188	154	98	205	227	249	54	93	9	215	230	224	69
146	110	198	44	159	152	30	53	225	169	126	252	136	231	187	199
10	2	33	155	19	234	6	174	102	165	203	103	157	135	242	202
84	148	37	60	5	41	49	190	13	7	16	62	243	226	29	56
107	177	134	158	101	85	96	211	14	74	214	17	238	112	32	75
181	233	217	218	72	220	184	163	97	48	191	11	213	196	200	129
22	64	52	128	45	186	207	208	61	27	151	46	80	247	87	28
144	23	109	76	201	50	40	251	131	115	123	236	185	70	36	90
100	246	83	162	195	182	209	63	81	171	4	168	142	254	35	73
172	121	175	166	239	43	94	143	88	122	229	253	86	71	176	111
160	132	192	18	99	150	78	250	245	108	212	124	120	34	228	161
147	219	133	91	59	89	55	173	114	105	113	140	1	21	38	156
58	248	42	178	145	194	65	256	117	170	241	104	210	95	77	141
67	206	31	167	149	138	244	66	79	116	57	216	68	235	237	51
20	193	137	47	183	26	139	240	255	15	82	180	130	189	222	221

Table 5 Proposed S-box3

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

Table 6 The dependence matrix for proposed S-box1

0.570312	0.523438	0.523438	0.585938	0.445312	0.507812	0.539062	0.445312
0.546875	0.437500	0.531250	0.531250	0.515625	0.437500	0.453125	0.468750
0.453125	0.515625	0.546875	0.468750	0.484375	0.484375	0.484375	0.562500
0.562500	0.546875	0.546875	0.500000	0.531250	0.468750	0.453125	0.484375
0.484375	0.562500	0.453125	0.437500	0.468750	0.515625	0.453125	0.562500
0.546875	0.468750	0.468750	0.562500	0.546875	0.609375	0.515625	0.484375
0.484375	0.515625	0.531250	0.578125	0.531250	0.531250	0.531250	0.531250
0.421875	0.546875	0.562500	0.500000	0.500000	0.515625	0.468750	0.468750

Table 7 The dependence matrix for proposed S-box2

0.507812	0.460938	0.554688	0.460938	0.445312	0.476562	0.507812	0.554688
0.484375	0.515625	0.500000	0.437500	0.546875	0.531250	0.453125	0.468750
0.515625	0.437500	0.515625	0.578125	0.515625	0.546875	0.484375	0.484375
0.468750	0.500000	0.484375	0.484375	0.515625	0.531250	0.468750	0.578125
0.484375	0.468750	0.484375	0.515625	0.500000	0.500000	0.484375	0.515625
0.468750	0.546875	0.515625	0.531250	0.500000	0.531250	0.546875	0.437500
0.500000	0.515625	0.515625	0.484375	0.453125	0.468750	0.453125	0.484375
0.515625	0.468750	0.546875	0.546875	0.531250	0.531250	0.531250	0.500000

3.4 Selection

Parents which are used to create next generations are selected from population according to their scaled fitness scores. Some of best individuals (elites) are directly transferred to next generation without any change, thus no crossover and mutation operations are applied on them. Number of elites are 7, in this model. Rest of the individuals (93 parents) are randomly selected and subject to a change, which are done by crossover (80% = 74 individuals) or mutation (20% = 19 individuals) operators.

Table 8 The dependence matrix for proposed S-box3

0.531250	0.578125	0.484375	0.500000	0.515625	0.515625	0.531250	0.484375
0.468750	0.500000	0.484375	0.500000	0.500000	0.500000	0.500000	0.500000
0.453125	0.546875	0.578125	0.531250	0.515625	0.468750	0.468750	0.437500
0.453125	0.453125	0.531250	0.468750	0.531250	0.562500	0.500000	0.468750
0.500000	0.531250	0.468750	0.500000	0.500000	0.468750	0.421875	0.531250
0.515625	0.515625	0.468750	0.468750	0.546875	0.500000	0.406250	0.421875
0.437500	0.500000	0.390625	0.562500	0.468750	0.500000	0.468750	0.531250
0.531250	0.500000	0.500000	0.500000	0.531250	0.453125	0.515625	0.531250

Table 9 BIC-nonlinearity matrix for proposed S-box1

—	101	107	101	99	99	103	107
101	—	98	102	106	104	104	106
107	98	—	102	106	100	104	106
101	102	102	—	102	108	104	106
99	106	106	102	—	96	100	102
99	104	100	108	96	—	104	100
103	104	104	104	100	104	—	100
107	106	106	106	102	100	100	—

Table 10 BIC-SAC matrix for proposed S-box1

—	0.500000	0.484375	0.509766	0.511719	0.501953	0.513672	0.519531
0.500000	—	0.478516	0.507812	0.488281	0.476562	0.494141	0.490234
0.484375	0.478516	—	0.498047	0.511719	0.503906	0.511719	0.496094
0.509766	0.507812	0.498047	—	0.501953	0.509766	0.507812	0.498047
0.511719	0.488281	0.511719	0.501953	—	0.494141	0.505859	0.503906
0.501953	0.476562	0.503906	0.509766	0.494141	—	0.500000	0.484375
0.513672	0.494141	0.511719	0.507812	0.505859	0.500000	—	0.527344
0.519531	0.490234	0.496094	0.498047	0.503906	0.484375	0.527344	—

Table 11 BIC-nonlinearity matrix for proposed S-box2

—	109	105	103	101	99	109	103
109	—	106	106	102	108	108	104
105	106	—	106	102	104	100	102
103	106	106	—	104	104	108	108
101	102	102	104	—	102	108	106
99	108	104	104	102	—	108	108
109	108	100	108	108	108	—	104
103	104	102	108	106	108	104	—

Table 12 BIC-SAC matrix for proposed S-box2

—	0.500000	0.490234	0.513672	0.505859	0.503906	0.492188	0.525391
0.500000	—	0.511719	0.496094	0.515625	0.500000	0.527344	0.533203
0.490234	0.511719	—	0.529297	0.503906	0.480469	0.498047	0.501953
0.513672	0.496094	0.529297	—	0.503906	0.509766	0.496094	0.490234
0.505859	0.515625	0.503906	0.503906	—	0.492188	0.496094	0.484375
0.503906	0.500000	0.480469	0.509766	0.492188	—	0.509766	0.500000
0.492188	0.527344	0.498047	0.496094	0.496094	0.509766	—	0.500000
0.525391	0.533203	0.501953	0.490234	0.484375	0.500000	0.500000	—

3.5 Crossover

Crossover operation is done by selecting two random points over parent chromosome and reversing the array between these points as given in Figure 3.

3.6 Mutation

Mutation operation is done by selecting two random genes over parent chromosome and swapping the values of these two genes as given in Figure 4.

Table 13 BIC-nonlinearity matrix for proposed S-box3

—	104	108	104	104	106	104	106
104	—	104	100	108	106	102	104
108	104	—	106	108	106	104	106
104	100	106	—	106	104	108	102
104	108	108	106	—	100	106	104
106	106	106	104	100	—	106	104
104	102	104	108	106	106	—	104
106	104	106	102	104	104	104	—

Table 14 BIC-SAC matrix for proposed S-box3

—	0.511719	0.480469	0.494141	0.533203	0.511719	0.484375	0.503906
0.511719	—	0.501953	0.507812	0.498047	0.509766	0.486328	0.498047
0.480469	0.501953	—	0.509766	0.521484	0.496094	0.539062	0.527344
0.494141	0.507812	0.509766	—	0.525391	0.478516	0.521484	0.511719
0.533203	0.498047	0.521484	0.525391	—	0.482422	0.494141	0.480469
0.511719	0.509766	0.496094	0.478516	0.482422	—	0.494141	0.484375
0.484375	0.486328	0.539062	0.521484	0.494141	0.494141	—	0.500000
0.503906	0.498047	0.527344	0.511719	0.480469	0.484375	0.500000	—

Figure 3 Crossover operation

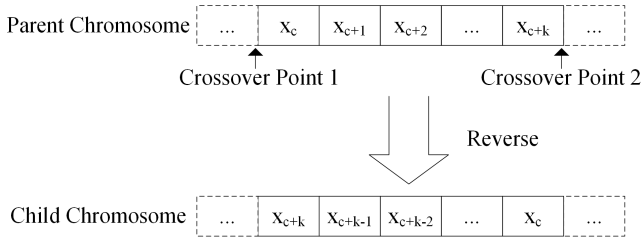
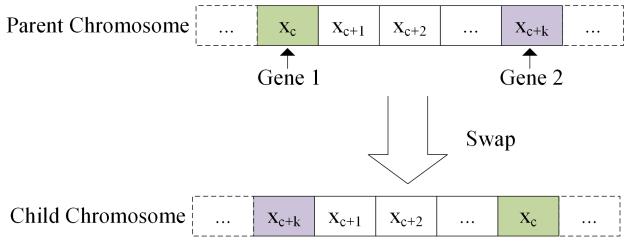


Figure 4 Mutation operation



3.7 Stopping criteria

The algorithm has three different stopping criteria. The first of these is the achievement of the targeted fitness (nonlinearity) value, which is 112. The second is that there is no improvement in the solution over the consecutive 3,000 generations. The final is that the maximum number of iterations, which are 100,000 is reached. Table 1 shows the comparison of some studies in the literature using genetic algorithms in S-box generation. The information about crossover, mutation techniques and fitness functions are presented in the table.

4 Performance analysis results of generated S-boxes

Performance analyses are performed on three S-boxes obtained from genetic algorithm-based S-box generation algorithm developed in this section and performance results are examined. There are many S-boxes generated which are close to the performance values of the S-boxes presented in this study. However, only three of them are presented to show its performance.

As a performance analysis, nonlinearity, SAC, output BIC, DP and LP analyses are performed. The results of the proposed S-boxes are compared with the literature. The performance results of the proposed S-boxes and some studies in the literature are given in Table 2.

Proposed S-box1, S-box2, S-box3 are shown in Tables 3–5. It is found that the nonlinearity values of the proposed S-box1 are 105, 108, 106, 108, 110, 106, 110, 108, the nonlinearity values of the proposed S-box2 are 105, 106, 106, 110, 106, 106, 106, 106 and the nonlinearity values of the proposed S-box3 are 106, 108, 102, 106, 106, 106, 106, 112. As can be seen from Table 2, the nonlinearity max., min., avg. values of proposed S-box1 are 110, 105, 107 respectively. The nonlinearity max., min. and avg. values of proposed S-box2 are 108, 106, 106 respectively. The nonlinearity max., min. and avg. values of proposed S-box3 are 112, 102, 106 respectively. When these values compared in Table 2 with all the results show that the proposed S-boxes has the best max., min. and avg. nonlinearity values after the S-box of AES algorithm construction.In addition, another important result is seen in Table 2 where the max. nonlinearity value of S-box3 is equal to the value of the AES algorithm. This value is better than all of the compared values in the literature.

The min., max. and avg. SAC values for the proposed S-boxes are shown in Table 2. SAC matrices of the proposed S-box1, S-box2 and S-box3 are given in Tables 6–8. When the SAC min., max. and avg. values of the proposed S-boxes are examined and compared with the literature, it is seen that they are very close to the optimum value of 0.5 which is very close to the AES algorithm.

Another important performance criterion is BIC. The BIC is assessed in two ways: BIC-SAC and BIC-nonlinearity. The BIC-SAC and BIC-nonlinearity matrices of the proposed S-boxes are given in Tables 9–14 are given. As shown in Table‘2, the BIC-SAC and BIC-nonlinearity value of the proposed S-box1 is found to be 0.5011 and 102.75, the proposed S-box2's BIC-SAC and BIC-nonlinearity value is found to be 0.5039 and 104.893. The proposed S-box3's BIC-SAC and BIC-nonlinearity value is found to be 0.5031 and 104.786. According to these results, the BIC-SAC value of the proposed S-box1 is among the best values and the proposed S-box2 has the best BIC-nonlinearity values after the S-box of AES algorithm BIC-nonlinearity. The BIC-nonlinearity value of the proposed S-box2 is good for the compared values after the AES algorithm.

As shown in Table 2, DP and LP are two important analyses used in determining the resistance to attack of S-boxes. Since the dimensions of the DP and LP matrices of the proposed S-boxes are large, they are not given in the article. The DP values of proposed S-box1 and 2 are found to be 10. Proposed all S-box has the best DP value values after AES algorithm. The LP values of proposed S-box2 and S-box3 are 0.125 and S-box1's LP value is 0.132. It is seen that the LP value of the proposed S-box2 has a good value compared to the literature studies after the AES and Skipjack algorithm S-boxes.

5 Conclusions

In this article, a GA-based S-box GA is presented. The proposed S-box performance criteria are described, and the proposed S-box GA is explained in detail. The suggested algorithm does not require complex matrix operations and is designed using the basic structures of the genetic algorithm. As an evaluation criterion for the algorithm, nonlinearity value is used which is one of the most important performance criteria. The performance of the generated S-boxes with the genetic-based algorithm proposed in the study is compared with those produced using different techniques in the literature. Comparison results are given in Table 2. When the comparison table is examined, it is seen that Nonlinearity value emerged as a more determinant value within the evaluation criteria. For this reason, Nonlinearity value was used as a determinant in the developed algorithm. It is seen that the results obtained in other evaluation criteria are generally close to each other and not sufficient for evaluation.

When all the performance results are evaluated together, it is found that the S-boxes generated with the developed algorithm achieved the results close and equal to the AES algorithm and obtained better results than the studies in the literature. One of the most important results in the study is that the maximum nonlinearity value of S-box3 is equal to the value of the AES algorithm. Obtaining a maximum nonlinearity value equivalent to the AES algorithm is an indication of the success of the genetic algorithm in such problems. As a result, suggested S-boxes are robust to attacks and have robust cryptographic features that can be used in encryption algorithm designs. With this study, it has been shown that a meta-heuristic method such as GA can be used effectively in this problem which has a huge solution space by using limited resources. In future studies, it is intriguing to investigate the effectiveness of different meta-heuristic methods or their hybrids in solving this problem.

References

Adams, C. and Tavares, S. (1990) 'The structured design of cryptographically good S-boxes', *Journal of Cryptology*, Vol. 3, No. 1, pp.27–41.

Akbar, S., Hayat, M., Iqbal, M. and Jan, M.A. (2017) 'IACP-GAENSC: evolutionary genetic algorithm based ensemble classification of anticancer peptides by utilizing hybrid feature space', *Artificial Intelligence in Medicine*, Vol. 79, pp.62–70.

Amigo, J.M., Kocarev, L. and Szczepanski, J. (2007) 'Theory and practice of chaotic cryptography', *Physics Letters A*, Vol. 366, No. 3, pp.211–216.

Banković, Z., Stepanović, D., Bojanić, S. and Nieto-Taladriz, O. (2007) 'Improving network security using genetic algorithm approach', *Computers & Electrical Engineering*, Vol. 33, No. 5, pp.438–451.

Biham, E. and Shamir, A. (1991) 'Differential cryptanalysis of DES-like cryptosystems', *Journal of Cryptology*, Vol. 4, pp.3–72.

Brickell, E.F., Denning, D.E., Kent, S.T., Maher, D.P. and Tuchman, W. (1995) 'Skipjack review: interim report', in *Building in Big Brother*, pp.119–130, Springer-Verlag, Inc., New York.

Çavuşoğlu, Ü. Kaçar, S., Pehlivan, I. and Zengin, A. (2017) 'Secure image encryption algorithm design using a novel chaos based S-box', *Chaos, Solitons & Fractals*, Vol. 95, pp.92–101.

Chen, G., Chen, Y. and Liao, X. (2007) 'An extended method for obtaining S-boxes based on three-dimensional chaotic baker maps', *Chaos, Solitons & Fractals*, Vol. 31, No. 3, pp.571–579.

Chen, G. (2008) 'A novel heuristic method for obtaining S-boxes', *Chaos, Solitons & Fractals*, Vol. 36, No. 4, pp.1028–1036.

Coley, D.A. (1999) *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific Publishing Co., Inc., Singapore.

Davies, D.W. (1986) 'Some regular properties of the 'data encryption standard' algorithm', in *Advances in Cryptology*, pp.89–96, Springer.

Demir, H.İ. and Erden, C. (2017) 'Solving process planning and weighted scheduling with wnoppt weighted due-date assignment problem using some pure and hybrid meta-heuristics', *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, Vol. 21, No. 2, pp.210–222.

Dong, Y-H., Lu, W-C., Xu, X., Zhao, X., Ho, K.M. and Wang, C.Z. (2017) 'Theoretical search for possible AU-SI crystal structures using a genetic algorithm', *Physical Review B*, Vol. 95, No. 13, p.134109.

Fung, K.K.H., Lewis, G.F. and Wu, X. (2017) 'The optimisation of low-acceleration interstellar relativistic rocket trajectories using genetic algorithms', *Acta Astronautica*, Vol. 133, pp.258–268.

Gladman, B. (2001) 'A specification for Rijndael', *The AES Algorithm*, Vol. 311, pp.18–19 [online] http://fp.gladman.plus.com/cryptography_technology/rijndael/aes.spec.

Hiassat, A., Diabat, A. and Rahwan, I. (2017) 'A genetic algorithm approach for location-inventory-routing problem with perishable products', *Journal of Manufacturing Systems*, Vol. 42, pp.93–103.

Hussain, I., Shah, T. and Gondal, M.A. (2012) 'A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm', *Nonlinear Dynamics*, Vol. 70, No. 3, pp.1791–1794.

Ivanov, G., Nikolov, N. and Nikova, S. (2016) 'Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties', *Cryptography and Communications*, Vol. 8, No. 2, pp.247–276.

Jakimoski, G. and Kocarev, L. (2001) 'Chaos and cryptography: block encryption ciphers based on chaotic maps', *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 48, No. 2, pp.163–169.

Kalaiselvi, K. and Kumar, A. (2016) 'Enhanced AES cryptosystem by using genetic algorithm and neural network in S-box', in *IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, IEEE, pp.1–6.

Kapuściński, T., Nowicki, R.K. and Napoli, C. (2016) 'Application of genetic algorithms in the construction of invertible substitution boxes', in *International Conference on Artificial Intelligence and Soft Computing*, Springer, pp.380–391.

Kapuściński, T., Nowicki, R.K. and Napoli, C. (2017) 'Comparison of effectiveness of multi-objective genetic algorithms in optimization of invertible S-boxes', in *International Conference on Artificial Intelligence and Soft Computing*, Springer, pp.466–476.

Khan, M., Shah, T., Mahmood, H., Gondal, M.A. and Hussain, I. (2012) 'A novel technique for the construction of strong S-boxes based on chaotic Lorenz systems', *Nonlinear Dynamics*, Vol. 70, No. 3, pp.2303–2311.

Khan, M., Shah, T. and Gondal, M.A. (2013) 'An efficient technique for the construction of substitution box with chaotic partial differential equation', *Nonlinear Dynamics*, Vol. 73, No. 3, pp.1795–1801.

Khan, R., Amjad, M. and Srivastava, A.K. (2017) 'Generation of automatic test cases with mutation analysis and hybrid genetic algorithm', in *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, IEEE, pp.1–4.

Kumar, A. and Chatterjee, K. (2016) 'An efficient stream cipher using genetic algorithm', in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, IEEE, pp.2322–2326.

Li, W. (2004) 'Using genetic algorithm for network intrusion detection', *Proceedings of the United States Department of Energy Cyber Security Group*, Vol. 1, pp.1–8.

Liu, H., Kadir, A. and Niu, Y. (2014) 'Chaos-based color image block encryption scheme using S-box', *AEU-International Journal of Electronics and Communications*, Vol. 68, No. 7, pp.676–686.

Liu, W., Yang, D. and Zhang, Y. (2016) 'Network security events analyze method based on neural networks and genetic algorithm', *DEStech Transactions on Engineering and Technology Research*, MIME.

Matsui, M. (1993) 'Linear cryptanalysis method for DES cipher', in *Workshop on the Theory and Application of of Cryptographic Techniques*, Springer, pp.386–397.

Messias, V.R., Estrella, J.C., Ehlers, R., Santana, M.J., Santana, R.C. and Reiff-Marganiec, S. (2016) 'Combining time series prediction models using genetic algorithm to autoscaling web applications hosted in the cloud infrastructure', *Neural Computing and Applications*, Vol. 27, No. 8, pp.2383–2406.

Mishra, A., Kumar, P., Misra, P. and Pathak, A.K. (2016) 'An approach for information hiding using inverse z-transform and genetic algorithm', *JIMET*, Vol. 1, No. 1.

Montazeri, A., West, C., Monk, S.D. and Taylor, C.J. (2017) 'Dynamic modelling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm', *International Journal of Control*, Vol. 90, No. 4, pp.661–683.

Özkaynak, F. and Özer, A.B. (2010) 'A method for designing strong S-boxes based on chaotic Lorenz system', *Physics Letters A*, Vol. 374, No. 36, pp.3733–3738.

Özkaynak, F. and Yavuz, S. (2013) 'Designing chaotic S-boxes based on time-delay chaotic system', *Nonlinear Dynamics*, Vol. 74, No. 3, pp.551–557.

Paes, F.G., Pessoa, A.A. and Vidal, T. (2017) 'A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem', *European Journal of Operational Research*, Vol. 256, No. 3, pp.742–756.

Pareek, N.K. and Patidar, V. (2016) 'Medical image protection using genetic algorithm operations', *Soft Computing*, Vol. 20, No. 2, pp.763–772.

Peng, J., Jin, S., Lei, L. and Liao, X. (2012) 'Construction and analysis of dynamic S-boxes based on spatiotemporal chaos', in *2012 IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, IEEE, pp.274–278.

Picek, S., Cupic, M. and Rotim, L. (2016) 'A new cost function for evolution of S-boxes', *Evolutionary Computation*, Vol. 24, No. 4, pp.695–718.

Picek, S., Ege, B., Batina, L., Jakobovic, D., Chmielewski, L. and Golub, M. (2014) 'On using genetic algorithms for intrinsic side-channel resistance: the case of AES S-box', in *Proceedings of the First Workshop on Cryptography and Security in Computing Systems*, ACM, pp.13–18.

Picek, S., Mariot, L., Yang, B., Jakobovic, D. and Mentens, N. (2017) 'Design of S-boxes defined with cellular automata rules', in *Proceedings of the Computing Frontiers Conference*, ACM, pp.409–414.

Picek, S., Miller, J.F., Jakobovic, D. and Batina, L. (2015) 'Cartesian genetic programming approach for generating substitution boxes of different sizes', in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ACM, pp.1457–1458.

Renzi, C. (2016) 'A genetic algorithm-based integrated design environment for the preliminary design and optimization of aeronautical piston engine components', *The International Journal of Advanced Manufacturing Technology*, Vol. 86, Nos. 9–12, pp.3365–3381.

Shankar, K. and Eswaran, P. (2016) 'An efficient image encryption technique based on optimized key generation in ECC using genetic algorithm', in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp.705–714, Springer, New Delhi.

Singh, V., Misra, A.K. and Singh, V. (2017) 'Cardiac image segmentation using improved genetic algorithm', *Indian Journal of Science and Technology*, Vol. 10, No. 7.

Sukhija, P., Behal, S. and Singh, P. (2016) 'Face recognition system using genetic algorithm', *Procedia Computer Science*, Vol. 85, pp.410–417.

Tang, G. and Liao, X. (2005) 'A method for designing dynamical s-boxes based on discretized chaotic map', *Chaos, Solitons & Fractals*, Vol. 23, No. 5, pp.1901–1909.

Tang, G., Liao, X. and Chen, Y. (2005) 'A novel method for designing S-boxes based on chaotic maps', *Chaos, Solitons & Fractals*, Vol. 23, No. 2, pp.413–419.

Wang, Y., Wong, K-W., Liao, X. and Xiang, T. (2009) 'A block cipher with dynamic S-boxes based on tent map', *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14, No. 7, pp.3089–3099.

Wang, Y., Wong, K-W., Li, C. and Li, Y. (2012) 'A novel method to design S-box based on chaotic map and genetic algorithm', *Physics Letters A*, Vol. 376, No. 6, pp.827–833.

Webster, A.F. and Tavares, S.E. (1985) 'On the design of S-boxes', in *Conference on the Theory and Application of Cryptographic Techniques*, Springer, pp.523–534.

Zaibi, G., Peyrard, F., Kachouri, A., Fournier-Prunaret, D. and Samet, M. (2014) 'Efficient and secure chaotic S-box for wireless sensor network', *Security and Communication Networks*, Vol. 7, No. 2, pp.279–292.

Zhao, X., Lin, Q., Chen, J., Wang, X., Yu, J. and Ming, Z. (2016) 'Optimizing security and quality of service in a real-time database system using multi-objective genetic algorithm', *Expert Systems with Applications*, Vol. 64, pp.11–23.