Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 57 (2015) 572 – 580
Procedia Computer Science

3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)

A Novel Ant Colony Optimization Based Scheme for Substitution Box Design

Musheer Ahmad [a,*], Deepanshu Bhatia[a], Yusuf Hassan[a]

[a]*Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islania, New Delhi 110025, India*

Abstract

Efficient substitution boxes have crucial importance in the design of cryptographically strong modern block encryption algorithms. They are the only nonlinear blocks that decide the security strength of entire cryptosystem. In this paper, a meta-heuristic approach based on Ant Colony Optimization and chaos is put forward to retrieve a suitable configuration of strong 8×8 substitution box. The optimization is carried out by transforming initial S-box to travelling salesman problem through edge matrix. The cryptographic strength of optimized S-box is investigated against standard tests such as bijectivity, nonlinearity, strict avalanche criterion, output bits independence criterion and differential approximation probability. The performance comparison of generated S-box with some recent chaos-based S-boxes evidently proves that the proposed scheme is proficient to discover strong nonlinear component of block encryption systems.

Peer-review under responsibility of organizing committee of the 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)

Keywords: Ant colony optimization; substitution box; chaotic maps; block ciphers; travelling salesman problem.

1. Introduction

Substitution boxes are one of the core components of modern cryptographic symmetric systems and are extensively used in modern block encryption algorithms such as DES (1977), IDEA (1991), AES (2001) etc. Shannon [1] suggested two fundamental properties of confusion and diffusion for the design of strong encryption

* Corresponding author. Tel.: +91-112-698-0281; Fax: +91-112-698-1261.
E-mail address: musheer.cse@gmail.com (M. Ahmad).

Peer-review under responsibility of organizing committee of the 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)
doi:10.1016/j.procs.2015.07.394

systems. In block cryptosystems, the substitution-boxes impart the property of confusion to obscure the relationship between secret key and ciphertext and the permutation-boxes provides the property of diffusion to spread out the redundancy in plaintext over ciphertext by rearranging the bits. The only nonlinear components in these cryptographic systems are the S-boxes which determine the security strength and robustness of entire security systems. Therefore, the main challenge in designing strong block cryptosystems is the design of a efficient S-box. Mathematically, an $n \times m$ S-box is a vector Boolean function F: $F_2^n \rightarrow F_2^m$, m may not be equal to n, which can be represented as a vector (f_0, f_1, …, f_{m-1}), where f_i is a Boolean function from F_2^n into F_2 with output range in Galois field GF(2). The traditional method for S-box design relied extensively on retrieving n Boolean functions of m bits and presenting them as $n \times m$ S-box.

The onset of linear and differential cryptanalysis paved way for the development of efficient S-boxes with varying methodologies that are based on different techniques. Detombe and Tavares [2] created a 5×5 S-box using near bent Boolean functions of five variables. However, their algorithm was useful only to generate S-boxes having odd input bit number. Jakimoski [3] designed a random 8×8 S-box through a chaos based approach. Similarly, Tang [4] proposed an 8×8 S-boxes based on discretized chaotic map, Özkaynak [5] proposed 8×8 S-box design based on chaotic Lorenz system. A prominent feature of chaos based approach is the use of random distribution property of chaotic maps. Unfortunately, most of the S-boxes generated using chaos-based methods are not as high performing as the one used in AES and thus the scope of performance improvisation is still there. Unlike previous algorithms that generate random S-boxes, a novel optimization based design methodology has been presented in this paper. Firstly, an initial S-box is generated using modulated chaotic variable samples obtained by combining chaotic logistic map and tent map. The initial S-box is transformed to a TSP problem through edge matrix. An efficient S box is further obtained by performing rotation on intermediate S-box according to the optimal path, in edge matrix, found using ACO technique. The remaining part in this paper is organized as follows: Section 2 gives the basic overview of the Ant Colony Optimization and its subclasses. In Section 3 the proposed meta-heuristic approach for optimized S-box design is explained. Performance investigation and comparison of generated S-box is discussed in Section 4 which is followed by conclusions of the work in last Section.

2. Ant Colony Optimization

The research paradigm that has been emphasized in this paper is the usage of meta-heuristic approach. The meta-heuristic approaches are applied to NP hard problems in an attempt to have a near optimal solution. A well acknowledged meta-heuristic approach, known as Ant Colony Optimization, used to design efficient cryptographic substitution box, is described in this section.

2.1. Introduction

Ant colony optimization is inspired from the behavior of natural ants. These ants are capable of finding shortest path between food source and their nest without any visual help by using the pheromone information feedback mechanism [6, 7, 8]. When an ant finds food, it walks backs to the colony leaving behind trails of markers, referred as *pheromones,* indicating the path leads to food. The other ants of the colony are expected to follow the same path with a certain probability. If they do so, they drop their own pheromone trails on the path, reinforcing it. As the ants drop pheromones each time they travel, shorter paths tend to be stronger, thus optimizing the solution. Dorigo [9] described the three step procedure for implementing artificial ants in ant colony optimization. Ant colony system [10] in ACO is explained in the subsequent subsection.

2.2. Ant Colony System

Travelling salesman problem is a problem of finding minimal cost closed tour by visiting each city once. It is characterized by a complete, connected graph where each connection has a weight corresponding to the distance that it connects to. The distance d_{ij} between cities i and j is the weight between nodes i and j, where $i \neq j$. Initially, m ants are placed randomly on nodes of graph. Each node in graph acts as state for the ant. The ants build the solution by randomly walking on graph having initial state either having empty memory or a memory containing sequence of

size unity and selecting best possible unvisited state until no state has been unvisited.

2.2.1. Edge Selection

In Ant system [11], each ant computes a set $A_i(x)$ on every iteration where for an ant k, the probability p_{ij}^k, of going to the state j from i depends upon the attractiveness of the move indicating the desirability, η_{ij} and the trail level of the move, τ_{ij} indicating how profitable that move has been in the past. The heuristic factor between the nodes i and j is given by $\eta_{ij} = 1/d_{ij}$ where d_{ij} is the distance between those nodes. In general, the probability with which the k^{th} ant moves from state i to j is given as

$$p_{ij}^k = \frac{\left(\tau_{ij}^{\alpha}\right)\left(\eta_{ij}^{\beta}\right)}{\sum_{l \epsilon N_i^k}\left(\tau_{il}^{\alpha}\right)\left(\eta_{jl}^{\beta}\right)} \tag{1}$$

where N_i^k is the set of the probable neighbors of the k^{th} ant from state i, $\alpha \geq 0$ controls the influence of τ_{ij} and $\beta \geq 1$controls the influence of η_{ij}. The probability of choosing a node outside N_i^k is 0.

In ACS, the edge selection rule, called as Rule 2, is given as

$$s = \begin{cases} \max_{j \epsilon N_i^k}\left(\tau_{ij}\right)\left(\eta_{ij}^{\beta}\right) & \text{if } q \leq q_0 \\ \text{Rule (1)} & \text{otherwise} \end{cases} \tag{2}$$

q is a random number in range [0, 1], q_0 is an exploitation parameter $(0 \leq q_0 \leq 1)$, Rule 1 follows equation (1). Rule 1 and 2 favors selection of shorter edges or edge having large pheromone deposit, with α=1 for Rule 1 and $2 \leq \beta \leq 5$ for Rule 1 and 2.

2.2.2. Pheromone Update

ACS updates the pheromone value according to two rules. Once it is performed when all the ants have selected next best state while building their solution, known as *Local Pheromone Update* rule and is given by

$$\tau_{ij}^k \leftarrow (1 - \xi)\tau_{ij}^k + \xi\tau_0 \tag{3}$$

where $0 < \xi < 1$ is a parameter. τ_0 is the initial pheromone value calculated as $\tau_0 = 1 \,/\, n\mathcal{L}_{nn}$ where n is the number of cities and $\mathcal{L}_{nn}$ is the length of nearest neighbour tour.

After completion of an iteration, the pheromone value of only the best path is lowered by a constant factor and the routes traced by only the globally best ant from the beginning is updated, known as *Global Pheromone Update* rule. The pheromone update is given by

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \tag{4}$$

Where $0 < \rho < 1$ is pheromone evaporation coefficient, $(1 - \rho)\tau_{ij}$ implements the pheromone decay and $\Delta\tau_{ij}^{bs}$ determines the pheromone to be deposited on the visited routes, which is given by

$$\Delta\tau_{ij}^{bs} = \begin{cases} 1/L_{bs} & \text{if } ij \textit{ is a transition in global best} \text{ tour} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where L_{bs} is the length of the globally best ant's tour from the beginning.

3. Proposed Scheme for Optimized S-box

The chaotic logistic map and tent map defined in equations (6) and (7) are used to generate initial S-box and control parameters of proposed meta-heuristic scheme employed for efficient S-box retrieval.

$$x_{i+1} = \mu x_i(1 - x_i) \tag{6}$$

$$y_{i+1} = \begin{cases} y_i/b & 0 < y_i \le b \\ (1 - y_i)/(1 - b) & b < y_i < 1 \end{cases} \tag{7}$$

Where, x_i, $y_i \in (0, 1)$ for $i = 0, 1, \ldots \infty$, are state variable of chaotic maps (6) and (7), respectively, $\mu \in (3.57, 4)$ and $b \in (0, 1)$ are their system parameters with permissible range. The states of Tent map are chaotically modulated to improve the statistics of its generated output sequence y, as shown in [12].

3.1. Initial S-box Generation

The steps of initial S-box generation are as follows:

1. Take proper initial conditions for y, μ, b, t_1, t_2 and an empty array S.
2. Iterate chaotic Logistic map (6) for t_1 times with $x_0=y$, let x be the output value.
3. Iterate Tent map (7) for t_2 times with $y_0=x$, let the value obtained after iterations is y.
4. Extract a random number out of y in the range [0, 255].
5. Add the number in array S if it doesn't belong to S.
6. Repeat Step 2-5 until 256 unique samples have been recorded.
7. Reshape array S into a 2-D matrix with rows 1~16 and columns 1~16 to get initial S-box S.

3.2. Edge Matrix

Edge matrix E is realization of a completely connected directed graph between 256 samples stored in the initial S-box treating them as nodes. The weight between two nodes is assigned randomly using random number generated in $[1, W_{max}]$ on further iterating the combination of chaotic Logistic and Tent maps. Where, W_{max} is the maximum weight that can be assigned to an edge, the self-looping should be avoided and can be achieved by setting $E_{i,i} > W_{max}$.

3.3. Working Algorithm

Before providing proposed ant colony optimization based scheme for S-box design, few significant terms are briefly described: Initial pheromone value, τ_0, is given as $\tau_0 = 1/n\mathcal{L}_{nn}$, where $\mathcal{L}_{nn}$ is calculated using Nearest neighbour algorithm [13], τ is a matrix of size $n_1 \times n_1$ holding pheromone value for each edge. A is a 2-D matrix of $m \times n_1$ where k^{th} row stores path traversed by k^{th} ant. L is array of size m to hold the length of tour of m ants.

S.01. Set $miss = 0$.
S.02. Generate a random value n_1 in range [1, 256].
S.03. Select unique n_1 numbers in $[1, 256]$ and store it in array $Node$.
S.04. Calculate $\mathcal{L}_{nn}$, path obtained here is stored as P_{gb} and initialise each element in τ as τ_0.
S.05. Initialise array L and itr as 0 and matrix A as empty.
S.06. Randomly place m ants on any position out of n_1 positions with start state as i_k for k^{th} ant. Add i_k in each row of A.
S.07. *for* r=2 to n_1 do
  S.7.1. *for* k=1 to m do
    S.7.1.a. $i \leftarrow$ Current state of ant k, $j \leftarrow$ Next state of ant k from equation 2.
    S.7.1.b. $L_k = L_k + E_{Node(i),Node(j)}$
    S.7.1.c. $\tau_{ij}^k \leftarrow (1 - \xi)\tau_{ij}^k + \xi\tau_0$
    S.7.1.d. Add j to A_k. Move the ant k to its next state j.
  end
  S.7.2 $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs}$

end

S.08. Find local best path P_{lb} in array A_k with minimum cost for current iteration.
S.09. IF $cost(P_{lb}) < cost(P_{gb})$ THEN $P_{gb} = P_{lb}$.
S.10. $itr = itr + 1$
S.11. Repeat Step S.06-S.10 until $itr < itr_{max}$.
S.12. Generate another random integral n_2 in range (0, n_1).
S.13. Obtain new S-box S_1 by rotating samples anti-clockwise by n_2 times on optimal path P_{gb}.
S.14. IF $avg(nonlinearity(S_1)) > avg(nonlinearity(S))$ THEN $S = S_1$, ELSE $miss = miss + 1$.
S.15. Repeat Step S.02-S.14 until $miss < miss_{max}$.
S.16. S holds the final configuration of optimized S-box.

3.4. Mutation

Edge matrix E is changed if average nonlinearity of S-box doesn't get improved after c number of iterations. The previous edge matrix E is discarded and a new matrix E' is generated which is having fresh random weights of edges to remove stagnation.

4. Performance Evaluation

The initial values y=0.123, μ =3. 99, b= 0.49, t_1=13 and t_2=8 are set for chaotic Logistic map and Tent map. The initial S-box, generated using method provided in Section 3.1, is utilized to form edge matrix E as discussed in Section 3.2. The experimental values taken to operate proposed approach are: pheromone evaporation coefficient ρ=0.15, local updation parameter ξ=0.1, q_0=0.65 and m=10, $miss_{max}$=1000, itr_{max}=1000, c=50. The algorithm discussed in Section 3.3 is applied and optimized S-box is obtained when $miss$ reaches the value of $miss_{max}$. The optimized S-box retrieved for given experimental set-up is depicted in Table 1. The performance of this S-box has been tested against various standard statistical parameters. The proposed S-box has also been compared with few of the recent chaos-based S-boxes.

Table 1. Proposed ACO based optimized Substitution box

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

4.1. Bijectivity

For an $n \times n$ S-box, the Boolean functions f_i is bijective if it satisfies the equation

$$wt(a_1 f_1 \oplus a_2 f_2 \oplus \dots \oplus a_n f_n) = 2^{n-1}$$

Where the $a_i \in \{0,1\}, (a_1, a_2, \dots, a_n) \neq (0,0,\dots,0)$and *wt(.)* is the hamming weight. This property ensures the distinct mapping of input vectors to the output vectors. It has been examined that the proposed S-box satisfies the condition of bijectivity.

4.2. Nonlinearity

Nonlinearity of a Boolean function $f(x)$: $\mathrm{F}_2^n \rightarrow \mathrm{F}_2$ is defined as the minimum distance from the set of affine functions and it ensures that output vectors are not linearly mapped from input vectors. It is evaluated as

$$N_f = 2^{n-1} - \frac{1}{2} \max_{w \in GF(2^n)} \left| S_{(f)}(w) \right|$$

$$S_{(f)}(w) = \sum_{x \in GF(2^n)} (-1)^{f(x)} (-1)^{x \cdot w}$$

Where $S_{(f)}(w)$ is the Walsh spectrum of $f(x)$ and $x \cdot w$ is the dot product of x and w. Nonlinearity of eight Boolean functions involved in proposed S-box are 108, 106, 106, 106, 106, 110, 106 and 108 making an average nonlinearity of 107. The comparison made in Table 2 shows the upright performance of generated S-box than some chaos-based S-boxes investigated in [3-5, 14-18] in terms of minimum, maximum and average nonlinearity scores.

Table 2. Comparison of nonlinearity scores of some S-boxes

S-box	$\mathbf{N_1}$	$\mathbf{N_2}$	$\mathbf{N_3}$	$\mathbf{N_4}$	$\mathbf{N_5}$	$\mathbf{N_6}$	$\mathbf{N_7}$	$\mathbf{N_8}$	**Min**	**Max**	**Mean**
Proposed	108	106	106	106	106	110	106	108	106	110	107
Jakimoski *et al.* [3]	98	100	100	104	104	106	106	108	98	108	103.3
Chen *et al.* [14]	100	102	103	104	106	106	106	108	100	108	104.4
Asim *et al.* [15]	107	103	100	102	96	108	104	108	96	108	103.5
Tang *et al.* [4]	103	102	104	101	108	106	102	105	101	108	103.8
Özkaynak *et al.* [5]	104	100	106	102	104	102	104	104	100	104	103.3
Khan *et al.* [16]	100	108	106	104	102	102	106	108	100	108	104.5
Ahmad *et al.* [17]	106	108	108	106	104	106	104	106	104	108	106
Gondal *et al.* [18]	98	100	106	104	106	100	106	104	98	106	103

4.3. Strict Avalanche Criteria

Strict avalanche criterion was first proposed by Webster and Tavares [19]. If a Boolean function satisfy SAC, it means the output bit would change with a probability of half whenever a single input bit is changed. This can be screened by determining dependency matrix of S-box under examination. The dependency matrix for the proposed S-box is provided in Table 3. The SAC of the proposed S-box is 0.501464 which is having a negligible deviation of 0.001464 from the ideal value 0.5. The SAC score for our S-box is quite better than scores of S-boxes generated in [3-5, 15-18].

Table 3. Dependency matrix for avalanche effect

0.54687	0.54687	0.53125	0.53125	0.42187	0.53125	0.50000	0.51562
0.54687	0.53125	0.46875	0.46875	0.53125	0.51562	0.46875	0.50000
0.54687	0.43750	0.43750	0.50000	0.48437	0.50000	0.48437	0.43750
0.46875	0.51562	0.48437	0.51562	0.48437	0.51562	0.50000	0.56250
0.45312	0.48437	0.50000	0.56250	0.48437	0.46875	0.46875	0.53125
0.51562	0.51562	0.48437	0.54687	0.51562	0.45312	0.50000	0.46875
0.50000	0.53125	0.51562	0.54687	0.51562	0.50000	0.48437	0.51562
0.45312	0.48437	0.40625	0.56250	0.51562	0.50000	0.53125	0.54687

Table 4. BIC non-linearity for $f_j \oplus f_k$

--	106	104	106	100	102	106	106
106	--	108	106	106	106	106	102
104	108	--	104	104	98	102	106
106	106	104	--	100	108	102	106
100	106	104	100	--	104	104	106
102	106	98	108	104	--	98	106
106	106	102	102	104	98	--	106
106	102	106	106	106	106	106	--

4.4. Output Bits Independent Criteria

The output bits independent criteria is another desirable property used to gauge S-boxes, which was also introduced by Webster and Tavares [19]. The near ideal value of BIC signifies that the avalanche variables are pairwise independent for a given set of avalanche vectors generated by complementing of just one bit. Adam and Tavares pointed out that for Boolean function f_j and $f_k (j \neq k)$ for two output bits, $f_j \oplus f_k$ should be highly non-linear to ensure the correlation of approximately zero when any single input bit is inverted. The results of BIC analysis for nonlinearity and strict avalanche criteria of proposed S-box are shown in Table 4 and Table 5, respectively. The BIC nonlinearity analysis has an average value of 104.214. The average of BIC-SAC stands at 0.5016043, which is very close to the ideal value ½. Thus, the proposed S-box fulfils the BIC requirement. The comparison drawn in Table 6 evidences the better BIC performance.

Table 5. Dependency matrix for BIC-SAC

--	0.5019	0.4883	0.4941	0.5000	0.5078	0.5078	0.4941
0.5019	--	0.5215	0.5293	0.5019	0.4922	0.51172	0.4980
0.4883	0.5215	--	0.4980	0.4941	0.5019	0.5234	0.5078
0.4941	0.5293	0.4980	--	0.4707	0.5039	0.5059	0.5137
0.5000	0.5019	0.4941	0.4707	--	0.5019	0.4941	0.5195
0.5078	0.4922	0.5019	0.5039	0.5019	--	0.5000	0.4844
0.5078	0.5117	0.5234	0.5059	0.4941	0.5000	--	0.4766
0.4941	0.4980	0.5078	0.5137	0.5195	0.4844	0.4766	--

Table 6. Comparison of strict avalanche criteria, BIC criteria and max differential probability of some S-boxes

S-box	SAC	BIC-NL	BIC-SAC	DAP
Proposed	0.5014	104.214	0.5016	0.0391
Jakimoski *et al.* [3]	0.4972	104.2	0.5017	0.0468
Chen *et al.* [14]	0.4999	103.1	0.5024	0.0468
Asim *et al.* [15]	0.4938	103.6	0.4992	0.0391
Tang *et al.* [4]	0.5058	102.7	0.4958	0.0546
Özkaynak *et al.* [5]	0.5048	103.8	0.5009	0.0391
Khan *et al.* [16]	0.4978	103.6	0.501	0.0468
Ahmad *et al.* [17]	0.4965	103.3	0.5073	0.0468
Gondal *et al.* [18]	0.4961	104.1	NA	0.0468

4.5. Differential Approximation Probability

An S-box having differential uniformity uniquely maps an input differential Δx to an output differential Δy. An S-box is said to be immune to differential cryptanalysis, introduced by Biham and Shamir [20], if it has as low value of maximum differential approximation probability as possible. The differential approximation probability, DAP^s, measures differential uniformity and is mathematically defined as

$$DAP^s\ (\Delta x \rightarrow \Delta y) = \frac{\#\{x \in X | S(x) \oplus S(x \oplus \Delta x) = y\}}{2^m}$$

Where X is the set of input values and 2^m is the total number of elements. The maximum DAP for proposed S-box comes out as 0.0391 which is comparable to probability for S-boxes designed in [5, 15], but it is quite enriched than ones reported in [3, 4, 14, 16-18].

5. Conclusion

This paper proposes to design an effective scheme based on chaos and ant colony optimization for efficient S-box design. The results of experimental analysis of generated S-box against non-linearity, bit independent criterion and strict avalanche criterion and differential probability depict that almost all the necessary properties for a strong S-box have been fulfilled. None of the measures required for an S-box to be secure has been compromised with. Moreover, subsequent comparison with some of representative and recently designed S-boxes show the enriched performance of proposed S-box. The proposed scheme exhibits the efficacy of generating efficient S-boxes with better cryptographic features. Thus, the proposed S-box generation scheme is practically suitable to use in the design of strong block cryptosystems.

References

[1] C. E. Shannon. Communication theory of secrecy systems. Bell Systems Technical Journal, 1949, 28, 656-715.
[2] J. Detombe, S. Tavares. Constructing large cryptographically strong S-boxes. Advances in Cryptology: Proceedings of AUSCRYPT'92, 1993, 718, 165-181.
[3] G. Jakimoski, L. Kocrev. Chaos and cryptography: block encryption cyphers. IEEE Transaction on Circuits Systems – I, 2001, 48(2), 163-169.
[4] G. Tang, X. Liao. A method for designing dynamical S-boxes based on discretized chaotic map. Chaos, Solitons and Fractals, 2005, 23(5), 1901-1909.
[5] F. Özkayanak, A. B. Özer. A method for designing stron S-boxes based on chaotic Lorenz system. Physics Letters A, 2010, 374(36), 3733-3738.

[6] R. Beckers, J. L. Deneubourg, S. Goss. Trails and U-turns in the selection of the shortest path by the ant Lasius Niger. Journal of Theoretical Biology, 1992, 159, 397–415.
[7] S. Goss, S. Aron, J. L. Deneubourg, J. M. Pasteels. Self-organized shortcuts in the argentine ant. Naturwissenschaften, 1989, 76(12), 579–581.
[8] B. Hölldobler, E. O. Wilson. The Ants. Springer-Verlag, 1990.
[9] M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, T. Stützle. Ant Colony Optimization and Swarm Intelligence. Proceddings of 5th International Workshop: ANTS 2006.
[10]M. Dorigo, L. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, 1997, 1(1), 53 -66.
[11]M. Dorigo, V. Maniezzo, A. Colorni, The ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, , 1996, 26(1), 29-41.
[12]M. Ahmad, H. Chugh, A. Goel, P. Singla. A Chaos Based Method for Efficient Cryptographic S-box Design. International Symposium on Security in Computing and Communications, 2013, 130--137.
[13]T. Cover, P Hart. Nearest Neighbour pattern classification. IEEE Transactions on Information Theory, 1967, 13(1), 21-27.
[14]G. Chen, Y. Chen, X. Liao. An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps. Chaos, Solitons & Fractals, 2007, 31(3), 571–577.
[15]M. Asim, V. Jeoti. Efficient and simple method for designing chaotic S-boxes. ETRI Journal, 2008, 30(1), 170-172.
[16]M. Khan, T. Shah. An efficient construction of substitution box with fractional chaotic system. Signal, Image and Video Processing, 2013, doi: 10.1007/s11760-013-0577-4.
[17]M. Ahmad, P. M. Khan, M. Z. Ansari. A Simple and Efficient Key-Dependent S-Box Design Using Fisher-Yates Shuffle Technique. International Conference on Security in Networks and Distributed Systems, 2014, 540-550.
[18]M. A. Gondal, A. Raheem, I. Hussain. A Scheme for Obtaining Secure S-Boxes Based on Chaotic Baker's Map. 3D Research, 2014, 5-17.
[19]A. F. Webster, S. Tavares S. On the design of S-boxes. Advances in Cryptology - CRYPTO'85, 1986, 218, 523-534.
[20]E. Biham and A. Shamir, Differential cryptanalysis of DES like cryptosystem, Journal of Cryptology, 1991, 4(1), 3-72.