Multiobjective Artificial Bee Colony Algorithm for S-box Optimization

Guanjie Qin[1, a], Xuemin Cheng[2,b] and Jianshe Ma[3,c]

[1]Optical Memory National Engineering Research Center (OMNERC), Graduate School at Shenzhen, Tsinghua University, Shenzhen, China, 518055

[2]Optical Memory National Engineering Research Center (OMNERC), Graduate School at Shenzhen, Tsinghua University, Shenzhen, China, 518055

[3]Optical Memory National Engineering Research Center (OMNERC), Graduate School at Shenzhen, Tsinghua University, Shenzhen, China, 518055

[a]lincoln89@126.com, [b]chengxm@sz.tsinghua.edu.cn, [c]ma.jianshe@sz.tsinghua.edu.cn

Keywords: Artificial Bee Colony; Substitution box; Pareto optimization; Swarm intelligence

Abstract. Substitution box (S-box) is an important nonlinear component in block cipher algorithms. Evaluating the cryptographic properties of an S-box requires attention to criteria such as nonlinearity, differential properties, and diffusion properties. In this paper, Artificial Bee Colony algorithm was introduced for global optimization of random S-boxes, using Pareto improvement to identify highly profitable solutions. The experimental results demonstrated the effectiveness of the proposed algorithm, which simultaneously optimized their nonlinearity, differential properties, and diffusion properties. The proposed model thus offers a new tool for the optimization of random S-boxes.

Introduction

Block ciphers are algorithms that convert plaintext to ciphertext in fixed-length units of bits and are widely used in the fields of information storage and data processing. For example, the Advanced Encryption Standard (AES), which is based on a symmetric-encryption algorithm, plays an important role in coding and decoding [1]. Substitution boxes (S-boxes) first appeared in the Lucifer algorithm and were soon widely used to fuzz the relationship between key and ciphertext in block ciphers [2]. An S-box is equivalent to a complex Boolean vector function, which changes a vector of n bits into another vector of m bits. In general, the cryptographic properties of an S-box are evaluated with reference to criteria such as orthogonality, nonlinearity, differential properties, diffusion properties, algebraic degree, distribution of terms, and correlation immunity. In addition, a reliable S-box has no trapdoors [3]. Swarm intelligence is a subdiscipline of artificial intelligence that concerns the collective behavior of animal communities, from the interactions between individuals in communities to the interactions between communities and their environment. In 2005, Karaboga introduced the Artificial Bee Colony (ABC) algorithm as a model of swarm intelligence based on the behavior of honeybees when foraging for nectar. The ABC algorithm was first used to solve the function-optimization problem. Having confirmed its effectiveness, researchers extended both the algorithm and its field of application. Simulating honeybees' foraging behavior helps cryptographers to represent problems, adjust parameters, and solve multiobjective problems. In this paper, we use the ABC algorithm to optimize several of the cryptographic properties of an S-box.

Criteria for Evaluating S-boxes

In general, the performance of an S-box is determined by its ability to withstand cryptanalysis. In this paper, nonlinearity, differential properties, and diffusion properties are regarded as the main criteria for evaluating the proposed model.

Nonlinearity. The concept of nonlinearity was introduced in 1988 and nonlinearity is now regarded as a major cryptographic property of a Boolean function; it describes the distance between a Boolean function and a linear-function class. Let $f_{n,m}(x)$ be a Boolean vector function that maps the elements

in Galois field GF(2^n) to those in GF(2^m), where $n \geq m$ and L(x) is any n-variate linear function. The nonlinearity of $f_{n,m}(x)$ is as follows [4]:

$$N_f = \min d(f_{n,m}(x), L(x)), \tag{1}$$

where $d(f_1(x), f_2(x)) = \left|\{x \in GF(2^n) \left| f_1(x) \neq f_2(x)\right.\}\right|$.

Differential properties. Differential cryptanalysis is a form of chosen-plaintext attack wherein the difference in input, Δx, is statistically compared with the difference in the corresponding output, Δy, in a round. Difference-distribution tables provide a powerful tool for analyzing the differential properties of S-boxes. A difference-distribution table is a two-dimensional table with axes Δx and Δy. Assuming that the inputs and outputs of the S-box are elements in GF(2^n) and GF(2^m), respectively, the elements in the table can be expressed as $\alpha_{i,j}$, where $\alpha_{i,j} = \left|\{\Delta x = i, \Delta y = j\}\right|$ and $0 \leq i \leq n-1, 0 \leq j \leq m-1$ [5]. The difference uniformity, δ, of an S-box can be defined as follows:

$$\delta = \max\left\{i \neq 0 \left| \alpha_{i,j} \right.\right\}. \tag{2}$$

To resist differential cryptanalysis, the difference uniformity of an S-box should be as small as possible. In addition to difference uniformity, robustness, R_f, can be used to evaluate the security of an S-box. R_f is calculated as follows [6]:

$$R_f = (1-(\delta / 2^n))(1-(\sigma / 2^n)), \tag{3}$$

where δ is the difference uniformity of the S-box and σ is the number of nonzero elements in the first column of the difference-distribution table. A large value of R_f indicates good differential properties, which means that δ and σ attain small values at the same time.

Diffusion properties. The diffusion properties of an S-box are evaluated by measuring the randomness of the change in output caused by a change in input. In this paper, the strict avalanche criterion (SAC) is used to evaluate S-box diffusion properties.

The Boolean vector function of an S-box can be expressed as $F(x): GF(2^n) \rightarrow GF(2^m)$, $n \geq m$. Then $\forall e_i \in GF(2^n)$ and $W_h(e_i) = 1$, where $W_h(e_i)$ is the Hamming weight of e_i. Therefore, the Boolean vector function satisfies the SAC when the following equalities hold [7]:

$$\begin{aligned} &\sum_{x \in GF(2^n)} (F(x) \oplus F(x \oplus e_i)) \\ &= \sum_{x \in GF(2^n)} (F_1(x) \oplus F_1(x \oplus e_i), \cdots, F_m(x) \oplus F_m(x \oplus e_i)) \\ &= (2^{n-1}, \cdots, 2^{n-1}), \end{aligned} \tag{4}$$

where $\oplus$ represents XOR.

In this paper, we use an inverse-probability table to describe S-box diffusion properties, with $e_i \in 2^n$ as the abscissa and bit of output y_j as the ordinate. Each element $P_{j,i}$ in the table corresponds to the inverse probability of y_j with e_i. If an S-box satisfies the SAC, every element in the table will take a value of 0.5 [8]. Therefore, we use the total deviation value D_f to evaluate the diffusion properties of the S-box:

$$D_f = \sum_{j=1,i=1}^{j=m,i=n} \left| P_{j,i} - 0.5 \right|, \tag{5}$$

where $P_{j,i}$ denotes the values of elements in the inverse-probability table.

To ensure that the S-box has good diffusion properties, the total deviation value should be as small as possible, as an S-box satisfies the SAC when $D_f = 0$.

Multiobjective Artificial Bee Colony Algorithm

General description. The ABC algorithm provides a model of swarm intelligence based on the foraging behavior of honeybees, which is characterized by both cooperation and a clear division of duties. In the ABC algorithm, the solution space is analogous to the whole set of food sources, each of which has a different level of profitability. An employer forager is associated with a specific food source and carries the information required to evaluate that source. There are two types of free forager: onlookers and scouters [9].

ABC solution model used to optimize S-box. To achieve multiobjective optimization, Pareto improvement (PI) is used to evaluate the S-box and the progress of the solution is modified to ensure orthogonality and prevent fixed points. Algorithm 1 describes the stages of optimization using the ABC algorithm.

Algorithm 1: Optimization process
Initialize external population scale (EPS), employer forager number (EFN), onlooker number (ON), scouter number (SN), initial food sources (IFS), local exploitation threshold (LET), and maximum cycle number (MCN)
round = 0
while (round < MCN) **do**
Update the food sources and related information using the PI ranking strategy (PIRS) and the updating strategy for food sources (USFS)
Send employer foragers and obtain values for the nonlinearity N_f, robustness R_f, and total deviation value D_f of the corresponding food sources
Send onlookers to carry out local optimization using the onlookers' local optimization strategy (OLOS)
Send scouters on a random search for new food sources
Update the archived elite solutions and related information within the external population
round = round + 1
endwhile

Algorithm 2 describes the process of PIRS and USFS, which are used for sorting and updating the food sources.

Algorithm 2: Stages of PIRS and USFS
$\forall S_i, \forall S_j$ in the ranking table of solutions; $0 \neq i < j$
if (S_j dominates S_i) **then**
exchange S_i and S_j in the table
endif
$S_0 = S_{scouter}$
for each $S_i, i \neq 0$ in the ranking table **do**
if $LEN(S_i) \geq LET$ **then**
exchange the element order of S_i to reduce the Hamming distance between S_i and S_{i-1}
endif
endfor

During the cycle stage, the profitability of each food source is evaluated using the PIRS. According to the definition of a non-dominated solution proposed by Pareto [10], we assume that one solution, S, dominates another, S', if $N_f(S) \geq N_f(S')$, $R_f(S) \geq R_f(S')$, and $D_f(S) < D_f(S')$, where $N_f(S)$, $R_f(S)$, $D_f(S)$ represent the nonlinearity, robustness, and total deviation value of the

solutions, respectively. The USFS identifies the leading solutions and thus requires food sources to be sorted before they can be updated.

The aim of the local search is to identify the optimal solution among the food sources. The local search is conducted using the OLOS. Algorithm 3 describes the stages of the OLOS.

Algorithm 3: Stages of OLOS
Input food source S, search-scale integer ϕ, non-dominated solution set S_n, and corresponding onlooker number N_{onlooker}, and initialize search count ($C_S = 0$)
while ($\phi > 0$ and $C_S \le N_{\text{onlooker}}$) **do**
$A = \{S' \mid W_h(S' \oplus S) \le \phi\}$
$C_S = C_S + 1$
for each $S' \in A$ **do**
if (S' dominates S) **then**
$S_n \leftarrow S'$, $LEN(S) = 0$
else if (S dominates S') **then**
$LEN(S) = LEN(S) + 1$
else
add S' to S_n
endif
endfor
endwhile

The results of the local optimization are used to update the elite solutions archived in the external-population set.

Results and Comparison

We used 10,000 random S-boxes in our experiments to test the optimization of nonlinearity, differential properties, and diffusion properties. The results show that the target properties were all optimized at the same time.

Fig. 1 shows the frequency distribution of nonlinearity, robustness, and value of total deviation from the SAC before and after optimization.

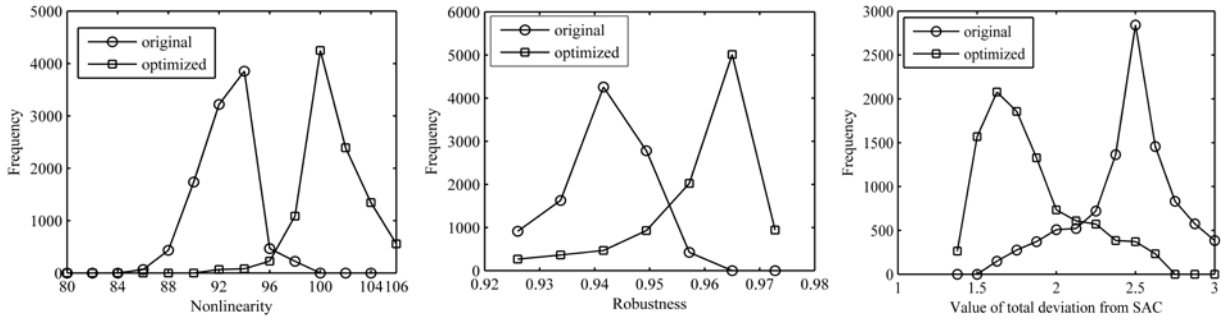

Fig. 1. Frequency distribution of target parameters

The results show that the nonlinearity, differential properties, and diffusion properties of the random S-boxes were improved simultaneously.

Table 1 proposes one of the optimized S-box, with nonlinearity of 106, difference uniformity of 6, robustness of 0.9727 and deviation value from SAC of 1.4375.

Table 1. The proposed S-box after optimization

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

Table 2 gives a comparison between the proposed S-box and those proposed in [11-13], in terms of nonlinearity, difference uniformity, robustness and deviation value from SAC.

Table 2. Comparison of cryptographic properties

	Nonlinearity	**Difference Uniformity**	**Robustness**	**Deviation Value from SAC**
Proposed S-box	106	6	0.9727	1.4375
S-box[11]	106	12	0.9494	0.8594
S-box[12]	104	10	0.9572	1.8671
S-box[13]	106	10	0.9572	1.9999

Discussion and Conclusions

The ABC algorithm offers a model of swarm intelligence based on the social behavior of honeybees. Applying multiobjective evaluation criteria to the ABC algorithm makes the proposed model suitable for the optimization of S-boxes. In this model, solutions interact with each other while food sources are updated and the degree of local optimization is controlled by the number of dynamic onlookers and the LET. In addition, the randomness of the scouters' search prevents the algorithm from becoming trapped in a local optimum. In our experiments, a random search was used to initialize the food sources, giving them a low profitability at the outset, but clearly displaying the optimization process. If certain highly profitable food sources were artificially introduced at the outset, the optimization process would take less time. In conclusion, a combination of the multiobjective ABC algorithm and PI can be used to optimize the S-boxes used in block ciphers, and thereby increase their security and the safety of data encrypted for storage.

Acknowledgment

This work was supported in part by grants from the National Natural Science Foundation of China (No. 61275003 and No. 51327005), and in part by the Guangdong Project (No. 2012B091100014).

References

[1] N. Ahmad and S.M. Rezaul Hasan: Electron. Lett. Vol. 48 (2012), pp. 1456-1457.

[2] I. Ben-Aroya and E. Biham: J. Cryptol. Vol. 9 (1996), pp. 21-34.

[3] L. Jinomeiq, W. Baoduui and W. Xinmei: J. Syst. Eng. Electron.Vol. 18 (2007), pp. 427-433.

[4] K.C. Gupta and P. Sarkar: IEEE Trans. Inf. Theory Vol. 51 (2005), pp. 339-348.

[5] E. Biham and A. Shamir, in: *Advances in Cryptology — CRYPTO' 92*, edited by E. Brickell, Springer, Heidelberg (1993), in press.

[6] J. Seberry, X. M. Zhang, and Y. Zheng, in: 1st ACM Conference on Computer and Communications Security, Virginia (1993), pp. 172-182.

[7] S.V. Radhakrishnan and S. Subramanian: Comput. Electr. Eng. Vol. 39 (2013), pp. 1006-1015.

[8] M. Talebi and M. Abadi, in: 2014 Iranian Conference on Intelligent Systems (ICIS), Bam (2014), pp. 1-5.

[9] C. Zhang, J. Zheng, and Y. Zhou: Neurocomputing Vol. 151 (2015), pp. 1198-1207.

[10] F. Karimi and S. Lotfi, in: 2014 Iranian Conference on Intelligent Systems (ICIS), Bam (2014), pp. 1-6.

[11] M.A. Gondal, R. Abdul, and I. Hussain: 3D Research Vol.5 (2014), pp. 43-50.

[12] Y. Wang, K. W. Wong, X. Liao, and T. Xiang: Commun. Nonlinear Sci. Numer. Simul. Vol. 14 (2009), pp. 3089-3099.

[13] F. Özkaynak and A.B. Özer: Phys. Lett. A Vol. 374 (2010), pp. 3733-3738.