World Applied Sciences Journal 28 (12): 2003-2009, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.28.12.1181

A Novel Method for Designing S-Boxes Based on Chaotic Logistic Maps Using Cipher Key

[1]Mona Dara and [2]Kooroush Manochehri

[1]Department of Computer Engineering and IT,
Amirkabir Uinversity of Technology, 424 Hafez Ave, Tehran, Iran
[2]Department of Computer Engineering and IT,
Islamic Azad University (Parand Branch), Bahonar Ave, Parand, Iran

Abstract: Advanced Encryption Standard (AES) block cipher system is widely used in cryptographic applications. Substitution boxes (S-boxes) are keystone of modern symmetric cryptosystems. They bring nonlinearity to cryptosystems and strengthen their cryptographic security. The S-box component that used in classic AES is fixed. If we generate this S-box dynamically, the cryptographic strength of AES cipher system would be increased. In this paper we use Chaotic Logistic Map to generate S-box for AES using its cipher key. The purpose of the proposed approach is to generate more secure S-boxes. The generated S-boxes will increase the complexity and also has better results in security analysis. Proposed S-box is analyzed and tested for the following criteria: avalanche effect, strict avalanche effect, bit independency criterion, nonlinearity, equiprobable input/output XOR distribution and key sensitivity. The results pass all of them which justify that the proposed algorithm is effective in generating strong S-boxes.

Key words: Advanced Encryption Standard · Key-dependent S-box · Cryptosystem · Chaotic Logistic Map

INTRODUCTION

AES is the norms of electronic data encryption used by the U.S. National Institute of Standards and Technology. It is a symmetric block cipher system. Three key lengths that are used in AES are 128, 192 and 256 [1].

AES uses a 16 * 16 matrix of byte values, called an S-box that contains a permutation of all possible 256 8-bit values [2].

The S-box of AES is fixed. It plays an important role in security of AES because it is the only nonlinear component of the AES and determines the strength of it. Therefore, the design of strong S-box is important issue. One way to increase security is using dynamic S-box instead of fixed one.

There are many methodologies proposed for the design of dynamic S-box over the past decade.

Shivkumar and Umamaheswari introduced AES-RC4 method that uses key schedule algorithm of RC4 to generate key dependent S-boxes and then rotate S-boxes for each round [2].

In [3], a randomly key-dependent S-box and inverse S-box is constructed by two steps: In the first, key pseudo-expansion algorithm is used and finally key-dependent S-box will be generated by the other algorithm. The key pseudo-expansion transformation takes secret key of 16, 24, or 32 bytes and generates a key pseudo-schedule of 176 bytes, which is used in the second transformation.

In [4], different S-boxes are used in each round. To this end, two functions are used. The key is used to create a series of sub keys K_1, K_2, ... , K_n by first function. Second function generates i'th S-box from K_i.

There are a number of papers published by researchers to construct dynamic S-boxes based on chaos theory.

The ergodic, mixing, random like behavior and basic characteristics of chaos, such as sensitivity to initial conditions are most attractive feature of chaotic systems. In [5], Jakimoski and Kocarev have proposed a four-step method to create S-box based on chaotic logistic map. Their methodology includes N-th iteration of chaotic logistic map, where they choose N=1000.

Corresponding Author: Mona Dara, Department of Computer Engineering and IT, Amirkabir Uinversity of Technology, 424 Hafez Ave, Tehran, Iran. Tel: +989391541622.

In [6], Tang *et al*., proposed a method based on baker map that is composed of two steps: First, by iterating a chaotic logistic map, a 8-bit sequence of binary random variables is generated from a real value trajectory obtained and turn it to a decimal integer on the range of $0\text{-}2^n$, then an integer table can be obtained. Second, a key-dependent permuting is used to shuffle the table nonlinear by a Baker map.

In [7], another method proposed by Tang and Liao that is based on discretized chaotic map and it consists of three steps: First, an integer sequence that can be regarded as secret key *K, K*= X_0= {1, 2, ..., 2^n}, is obtained in an arbitrary way. Second, for a given $M = 2^n$ and A, iterating the chaotic map more than k times with the initial value X_0, one can obtain a permuted integer sequence $\{X\}$. Finally, by translating the $\{X\}$ to a $2^{n/2}\times 2^{n/2}$ table, the S-box is obtained.

In [8], Chen *et al*., proposed another three-step method for designing S-boxes based on three-dimensional baker map which has more intensive chaotic characters than the two-dimensional one.

In [9], Özkanak and Özer proposed a new scheme which uses the Lorenz system to generate the S-box and improve the cryptographic performance of S-box by a special shifting method.

In [10], Wang *et al*., have proposed another method based on chaos and genetic algorithm for obtaining cryptographically strong 8 * 8 S-boxes. This method is divided into two phases. In the first phase, the S-boxes are generated by iterating the chaotic maps. In the second phase, the S-boxes generated in the first phase are used as the initial individuals and a genetic algorithm is employed to improve their performance.

Most of the chaos based cryptosystems use explicitly the initial condition and iteration number of chaotic map as a secret key and this is usually the cause of weakness of them, as is shown in the many successful attacks against these cryptosystems [11-13].

To avoid this possible vulnerability in S-box of AES, we can make it dependent on cipher key. So it will change with every changing of cipher key and the result will be increasing the cryptographic strength of AES algorithm. In [14], Pareek *et al*., presented a method that uses an external secret key of maximum 128-bits length to drive the chaotic parameters. The encryption of each block of plaintext has been made dependent on the secret key and the cryptosystem is further made robust against any reasonable attack.

In [15], Pareek *et al*., proposed another method for image encryption based on chaotic logistic maps using an external secret key of 80-bits length. The initial conditions for the logistic map are derived using the external secret key by providing different weightage to all its bits.

In this paper, we propose a new method based on chaotic logistic map that uses cipher key to generate initial value. We use method which is proposed by Pareek *et al*. [15] for calculating an initial condition of logistic map based on external secret key and extend it to be used for 128, 192 or 256 bits cipher key of AES.

The rest of this paper is organized as follows: Section 2 introduces chaotic logistic map. In section 3, the proposed method is introduced. The performance of proposed S-box and Jakimoski's S-box is compared in section 4. The paper is concluded in session 5.

Chaotic Logistic Map: Chaos theory is a blanketing theory that covers most aspects of science, hence, it shows up everywhere in the world today: mathematics, physics, biology, finance, computer and even music. Chaotic systems have many interesting features, such as the sensitivity to the initial condition and control parameter, ergodicity and mixing property, which can be connected with some cryptographic properties of good ciphers, such as confusion/diffusion, balance and avalanche property. These properties make the chaotic systems a worthy choice for constructing the cryptosystems [17].

One of the most famous chaotic systems is logistic map that is a nonlinear return map given by:

$$X_{i+1} = rX_i(1\text{-}X_i) \quad (1)$$

where, X_i *(*i = 0, 1,...*)* is the state value of the logistic map and r is the control parameter.

The logistic map is often used in cryptography because its chaotic orbits X_i (0,1) when the initial point X_0 (0,1) and the control parameter value r (0,4) [18].

This map creates chaotic orbits for r >3.83. Figure 1 shows the plot of X_i VS i for X_0=0.5 and r=3.9999 after 500 iteration.

Proposed Method: A dynamic S-box is generated by iterating chaotic logistic map in our proposed method. This method doesn't use the system parameters or initial condition as the secret key. To achieve this goal, we modify Pareek's method [15] and add a new parameter named keysize which determines the length of the secret key in binary format for compatibility with three different AES key lengths.

We use cipher key to generate initial value of logistic map (X_0) and outputs of logistic map as the values of dynamic S-box.

As mentioned before, logistic map produces real numbers ranging from 0 to 1 and the S-box elements are integer numbers ranging from 0 to 255. To convert the outputs of chaotic map to the inputs of S-box, we use Baptista's method in [16].

Baptista proposed a method for message encryption (message is an alphabetic text) using chaotic logistic equation. The basic idea of this method is to encrypt each character of the message as the integer number of iterations which should be run in the logistic map. To achieve this goal, he divide the range of [0,1] into S intervals with the size of 1/256 (S is the number of alphabet units), then associate S units alphabet with the S intervals.

The encryption of a unit is the number of iterations of Eq. (1) needed to reach the aimed intervals associated with that character.

In our proposed method, we choose S=256 because of the number of S-box elements and associate each integer number in the range of 0 to 255 with one of the 256 intervals. In the next step, we calculate the state value of logistic map (X_i) using Eq. (1) and then use the index of the interval containing the state value as an integer input of S-box instead of using the number of iterations in Baptista's method.

Because the S-box is required to be a one-to-one map, we tag the intervals that have been met and discard them, so if next X_i is in those intervals, we iterate the logistic map to get another X_i.

Repeat this operation until all of the 256 inputs of S-box are created.

The applied algorithm is explained step by step:

Step1: The method uses AES cipher key, *K,* that has 128/192/256 bits length as

$$K = K_0, K_1, \ldots, K_{31} \text{ (Hexadecimal) for 128 bit key} \quad (2)$$
$$K = K_0, K_1, \ldots, K_{47} \text{ (Hexadecimal) for 192 bit key} \quad (3)$$
$$K = K_0, K_1, \ldots, K_{63} \text{ (Hexadecimal) for 256 bit key} \quad (4)$$

and converts these bits to a binary form to define initial value X_0. Here, K_i is part of secret key in hexadecimal mode.

Step2: Next the real numbers X_{01} and X_{02} values are calculated as follows

$$X_{01} = K_1 \times 2^{keysize/2-1} + K_3 \times 2^{keysize/2-2} + \ldots + K_{keysize-1} \times 2^0 \quad (5)$$
$$X_{02} = K_0 \times 2^{keysize/2-1} + K_2 \times 2^{keysize/2-2} + \ldots + K_{keysize} \times 2^0 \quad (6)$$

Which K_i defines the i'th bit of cipher key and *keysize* is the size of key in binary format.

Then the initial value X_0 for logistic map can be calculated as:

$$X_0=(X_{01}+X_{02}) \quad (7)$$

Step3: As can be seen in Figure 1, the map defined in [0,1], so we divide this interval into 256 subintervals with the size of 1/256 as [0,1/256), [1/256,2/256),…,[255/256,1] (8)
and assign the numbers 0 to 255 to the intervals so that one number is assigned to exactly one region.

Step 4: Then state value of logistic map X_i will be calculated using (1) and take the index of the interval which meets X_i as an input of S-box.

Tables (1) gives an example of proposed S-box when key equal to (00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F) is applied.

Security Analysis: In this section, tests such as, Avalanche effect, Strict Avalanche Criteria (SAC), Bit Independence Criteria (BIC), Nonlinearity, Equiprobable Input/Output XOR Distribution and Key sensitivity are applied to test the strength of the S-box. At the same time, in order to compare, all these properties of the S-box generated by the method proposed by Jakimoski and Kocarev (Table 1 of [5]) is computed too. All these data are given as follows.

Avalanche Effect: AVAL Criteria is an important cryptographic property of block ciphers which says that small number of bit differences in the input plaintext leads to an "avalanche" of changes, that is, results in a large number of cipher text bit differences. More formally, a function f: $\{0,1\}^n \rightarrow \{0,1\}^n$ satisfies AVAL criterion if whenever one input bit is changes, on the average half of the output bits change, where i and $j \square (1, 2,\ldots, n)$ are input and output bits respectively [19]. The avalanche effect of our proposed S-box and Jakimoski's one for key = "00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f" is shown in Table 2 and 3.

To perform the test we change plaintext bit to"10"instead of"00"and "01"instead of "11".

The result obtained is 0.5312 and 0.5468 for Proposed S-box and 0.4921 and 0.4687 for Jakimoski's method which proves that our proposed S-box is better than Jakimoski's.

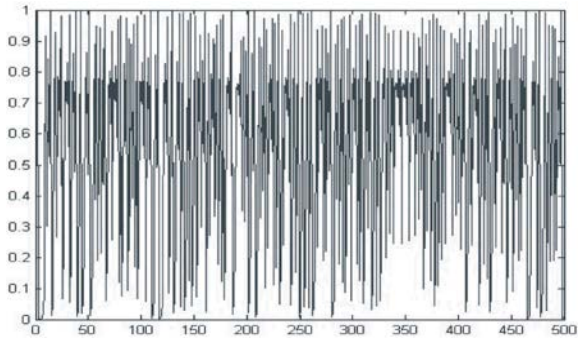


Fig. 1: Chaotic behavior of logistic map with $X_0 = 0.5$ and $r = 3.9999$

Table 1: Proposed AES S-box (in hexadecimal format)

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

Table 2: Avalanche test for our proposed method

S.No	Input	Output	Avalanche test
1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	B9 64 07 cc cc b2 7a 1c d4 e0 8b 53 99 4c c4 0e	0.5312
	10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	ad 60 d4 91 9f e6 b7 d7 62 b2 7c 44 72 d7 43 40	
2	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	30 7a df 31 90 f1 68 48 3b 1c 75 a6 c7 be e2 4d	0.5468
	01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	a6 2b 88 d1 1a ed 16 87 ee 01 81 4c fa da 84 4d	

Table 3: Avalanche test for Jakimoski's method

S.No	Input	Output	Avalanche test
1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	16 34 b9 af 01 04 79 d6 9f b8 c8 5a 4b 90 4f 44	0.4921
	10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	fc b5 41 97 1e 01 83 a7 56 0e a5 3f 2e c1 cb 84	
2	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	b3 3f 13 14 dd c3 44 6b 16 6d 1e e8 61 8a 79 1f	0.4687
	01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	8f 5d f6 4e f9 6a 02 1a 38 6f b0 6f 75 41 64 a4	

Table 4: Dependence matrix of Proposed S-box

	1	2	3	4	5	6	7	8
1	0.51	0.59	0.57	0.44	0.47	0.51	0.42	0.55
2	0.57	0.53	0.55	0.49	0.42	0.51	0.40	0.50
3	0.50	0.59	0.47	0.42	0.50	0.45	0.48	0.60
4	0.46	0.52	0.40	0.52	0.61	0.48	0.53	0.46
5	0.51	0.48	0.52	0.54	0.45	0.54	0.56	0.56
6	0.56	0.36	0.50	0.58	0.42	0.51	0.63	0.50
7	0.55	0.56	0.54	0.56	0.57	0.53	0.41	0.50
8	0.39	0.50	0.66	0.62	0.56	0.52	0.54	0.44
Mean	0.51078							

Table 5: Dependence matrix of Jakimoski's S-box

	1	2	3	4	5	6	7	8
1	0.49	0.48	0.41	0.38	0.65	0.44	0.53	0.51
2	0.49	0.50	0.60	0.41	0.55	0.53	0.58	0.53
3	0.40	0.49	0.60	0.58	0.47	0.52	0.50	0.39
4	0.58	0.54	0.53	0.57	0.51	0.59	0.58	0.51
5	0.59	0.59	0.53	0.47	0.44	0.41	0.50	0.51
6	0.50	0.45	0.52	0.66	0.56	0.51	0.51	0.53
7	0.41	0.39	0.54	0.54	0.35	0.43	0.62	0.44
8	0.56	0.45	0.57	0.52	0.45	0.48	0.45	0.58
Mean	0.5078							

Strict Avalanche Criteria (SAC): If a function satisfies this criterion, each of its output bits should change with a probability of a half whenever a single input bit x is complemented. In general, the dependence matrix is used to test 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 fulfills the SAC. The details of calculating the dependence matrix can be found in Ref. [19].

The dependence matrix of our proposed S-box and Jakimoski's one and mean values of them when key equal to (00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F) can be found in Table 4 and 5.

The average value for our proposed S-box is calculated as 0.51078 while it is calculated as 0.5078 for Jakimoski. It is obvious that the calculated average values are very close to the ideal average value 0.5000.

The Output Bits Independence Criterion (BIC): This is another criterion which states that the output bits j and k should change independently when any single input bit i is inverted, for all i, j and k. The avalanche and BIC criteria appear to strengthen the effectiveness of the confusion function.

To measure the bit independence concept, one needs the correlation coefficient between the j 'th and k'th components of the output difference string, which is called the avalanche vector A^{ei}. A bit independence parameter corresponding to the effect of the i 'th input bit change on the j 'th and k'th bits of A^{ei} is defined as:

$$BIC(a_j,a_k) = max|Corr(a_j^{ei},a_k^{ei})|\ 1 \le i \le n \quad (9)$$

Overall, the Bit Independence Criterion (BIC) parameter for the s-box function f is then found as:

$$BIC(f) = max\ BIC(a_j,a_k)\ 1 \le j,k \le n\ ,\ j \ne k \quad (10)$$

which demonstrates how close f is to satisfying the BIC. BIC (f) takes values in [0, 1]. It is ideally equal to 0 and, in the worst case, it is equal to1 [20].

Table 6: A comparison of Lp.

S-box	Lp
The proposed S-box	0.0625
The Jakimoski's S-box	0.0625

The calculated result for our proposed method is 0.4993 and for Jakimoski's method is 0.5018 which shows that bit independency of our proposed S-box is less than Jakimoski's S-box. So our proposed method has better performance.

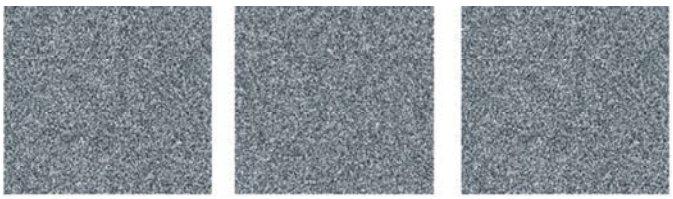
Nonlinearity: Linear cryptanalysis studies linear approximations of the cryptosystem. The purpose is to construct linear equations between input plaintext and output cipher text and enumerate all linear approximations of the S-box in a linear approximation table. Otherwise, the nonlinearity is measured by calculating the linear probability approximation (Lp):

$$Lp = \max_{a,b\neq 0} \left(\frac{\#\{x \in X \mid x \cdot a = f(x) \cdot b\} - 2^{n-1}}{2^{n-1}} \right)^2 \quad (11)$$

where $a \bullet b$ denotes the parity of bit-wise product of a and b, X is the set of all possible inputs and 2^n is the number of its elements. Decreasing the Lp yields to increasing the complexity of the linear cryptanalysis attack [5].

The Lps of proposed S-box and Jakimoski's one are computed according to Eq. (11) and the values are listed in Table 6. As you see, two methods are at the same level of Lp performance.

The Equiprobable Input/Output XOR Distribution: Biham and Shamir introduced differential cryptanalysis, 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, the S-box could be immune to the differential attack. The differential approximation probability (Dp) is a measure for differential uniformity and is defined as:



A. Encrypted with key $K1$ B. Encrypted with key $K2$ C. Difference of images in A & B

Fig. 2: Key sensitivity Analysis

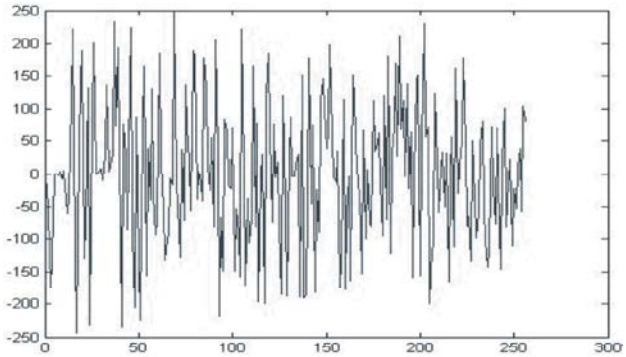


Fig. 3: Plot of the difference of the S-boxes elements

Table 7: A comparison of *Dp*.

S-box	*Dp*
The proposed S-box	0.0390
The Jakimoski's S-box	0.0468

$$Dp = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X \mid f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \right) \quad (12)$$

where X is the set of all possible input values and 2^n is the number of its elements. Actually Dp is the maximum probability of output difference Δy, when the input difference is Δx [7].

The *Dp*s of proposed S-box and Jakimoski's one are computed according to Eq. (12) and the values are listed in Table 7. In comparison with Jakimoski's method, our proposed method has better *Dp* performance.

Key Sensitivity Test: This test has been performed according to the following steps:

First, an original image is encrypted by using the test key, $K1$= "00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f "(Hex).

Then, one bit of test key is changed, so that the original key becomes, say $K2$= "01 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f "(Hex) which is used to encrypt the same image.

Finally, the above two ciphered images, encrypted by the two slightly different keys, are compared. Figure 2 shows the test result.

This clearly shows that even when one bit of key is changed, it has its influence on all the pixels.

Moreover, we found 256 different elements between S-box1 and S-box2 by changing only 1 bit of the key, thus 100% of second S-box is changed. The difference of S-box1 and S-box2 elements is illustrated in Figure 3.

CONCLUSION

In this paper, we have introduced a method for constructing key dependent S-box. This method will lead to more secure block ciphers by solving the problem of the fixed structure S-box and will generate key-dependent flexible S-box to enhance its most important security parameters. So the main advantage of the proposed algorithm is that many S-boxes can be generated by changing cipher key.

The proposed S-box passes all of the Avalanche effect, Strict Avalanche Criteria (SAC), Bit Independence Criteria (BIC), Nonlinearity, Equiprobable Input/Output XOR Distribution and Key sensitivity Analysis which are important criterions for strong S-boxes to produce more confusion to the encryption process. The results show that all of the criteria are approximately fulfilled and the comparisons show that our proposed method has better performance than the one in Ref [5].

REFERENCES

1. Xiang Li, Junli Chen, Dinghu Qin and Wanggen Wan, 2010. Research and Realization based on hybrid encryption algorithm of improved AES and ECC, International Conference On Audio Language and Image Processing (ICALIP), pp: 396-400.
2. Shivkumar, S. and G. Umamaheswari, 2011. Performance Comparison of Advanced Encryption Standard (AES) and AES key dependent S-box Simulation using MATLAB, International Conference on Process Automation, Control and Computing (PACC), pp: 1-6.
3. Kazlauskas, K. and J. Kazlauskas, 2009. Key-Dependent S-Box Generation in AES Block Cipher System, Informatica, 20(1): 23-34.
4. Keliher, L., 1997. Substitution-Permutation Network Cryptosystems Using Key-Dependent S-Boxes, M.S. Thesis, Queen's University, Kingston, Canada.
5. Jakimoski, G. and L. Kocarev, 2001. Chaos and Cryptography: block encryption ciphers, IEEE Trans Circuits Syst-I, 48(2): 163-70.
6. Tang, G., X. Liao and Y. Chen, 2005. A novel method for designing S-boxes based on chaotic maps, Chaos Solitons Fractals, 23(2): 413-419.
7. Tang, G. and X. Liao, 2005. A method for designing dynamical S-boxes based on discretized chaotic map, Chaos Solitons Fractals, 23(5): 1901-1909.
8. Chen, G., Y. Chen and X. Liao, 2007. An extended method for obtaining S-boxes based on three-dimensional chaotic baker maps, Chaos Solitons Fractals, 31(3): 571-579.
9. Özkanak, F. and A.B. Özer, 2010. A method for designing strong S-boxes based on chaotic Lorenz system, Phys. Lett. A, 374(36): 3733-3738.
10. Wang, Y., K. Wong, C. Li and Y. Li, 2012. A novel method to design S-box based on chaotic map and genetic algorithm, Phys. Lett. A, 376(6-7): 827-833.
11. Alvarez, G., F. Montoya, M. Romera, G. Pastor, 2000. Cryptanalysis of a chaotic encryption system, Phys. Lett. A., 276(1): 191-196.
12. Alvarez, G., F. Montoya, M. Romera, G. Pastor, 2003. Cryptanalysis of an ergodic chaotic cipher, Phys. Lett. A, 311(2): 172-179.
13. Alvarez, G., F. Montoya, M. Romera, G. Pastor, 2003. Cryptanalysis of a chaotic secure communication system, Phys. Lett. A, 306(4): 200-205.
14. Pareek, N.K., V. Patidar and K.K. Sud, 2003. Discrete chaotic cryptography using external key, Phys. Lett. A, 309(1-2): 75-82.
15. Pareek, N.K., V. Patidar and K.K. Sud, 2006. Image encryption using chaotic logistic map, Image and Vision Computing, 24(9): 926-934.
16. Baptista, M.S., 1998. Cryptography with chaos, Phys. Lett. A, 240(1-2): 50-54.
17. Behnia, S., A. Akhshani, A. Akhavan, H. Mahmodi, 2009. Applications of tripled chaotic maps in cryptography, Chaos Solitons Fractals, 40(1): 505-519.
18. Stavroulakis, P. and M. Stamp, 2010. Handbook of Information and Communication Security, Springer, New York.
19. Webster, A.F. and S.E. Tavares, 1998. On the design of S-boxes, Advances in Cryptology: Proceedings of CRYPTO '85, Springer-Verlag, pp: 523-534.
20. Vergili, I. and M.D. Yücel, 2001. Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen n× n S-boxes, J. Elec. Engin, 9(2): 137-145.