ISSN: 0067-2904

Generating dynamic S-BOX based on Particle Swarm Optimization and Chaos Theory for AES

Alaa F. Kadhim, Zainab Ali Kamal*
Computer Sciences Department, University of Technology, Baghdad, Iraq

Abstract

Data security is a significant requirement in our time. As a result of the rapid development of unsecured computer networks, the personal data should be protected from unauthorized persons and as a result of exposure AES algorithm is subjected to theoretical attacks such as linear attacks, differential attacks, and practical attacks such as brute force attack these types of attacks are mainly directed at the S-BOX and since the S-BOX table in the algorithm is static and no dynamic so this is a major weakness for the S-BOX table, the algorithm should be improved to be impervious to future dialects that attempt to analyse and break the algorithm in order to remove these weakness points, Will be generated dynamic substitution box (S-BOX) base on the input key, shifting, chaotic theory (1D, 2D logistic map), and particle swarm algorithm. At the same time will be generated inverse of the table S-BOX through output of the S-BOX which will be generated from the suggestions above will return the values of the union of the row and the column for all the values generated for S-BOX. The S-BOX output is tested in several measurements represent (complexity, time, avalanche criterion, and balance) and the results show that any change in the input will change the output S-BOX also the proposed algorithm will be measured by the five statistical and NIST measurements all results will show a successful exception random excursions, random excursions variant (test not application). The time needed to implement it requires only milliseconds and is approximated to the time taken for the original algorithm.

Keywords*:* AES, key, shifting, chaos theory 1D and 2D logistic map, PS

توليد ديناميكية S–BOX على أساس الجسيمات سرب الطيور ونظرية الفوضى ل AES

علاء كاظم، زينب علي*
قسم علوم الحاسبات، الجامعة التكنولوجيه، بغداد، العراق

الخلاصة

امن البيانات مطلب رئيسي في عصرنا .و نتيجة للتطور السريع لشبكات الكمبيوتر الغير امنة يجب علينا حماية بياناتنا الشخصية من الاشخاص الغير مصرح لهم .و بسبب تعرض خوارزمية **AES** للعديد من الهجمات سواء كانت عملية مثل هجوم القوة الغاشمة او نظرية مثل الهجمات التفاضلية و الهجمات الخطية وواحدة من نقاط الضعف الموجودة في جدول ال **s–box** هو انه يكون ثابت لذلك يجب تحسين الخوارزمية لكي تكون منيعة ضد الهجمات المستقبلية التي تحاول كسر الخوارزمية و لازالة نقاط الضعف الموجودة في جدول ال **s–box** يتم انشاء **s–box dynamic** بالاعتماد على المفتاح المدخل و نظرية الفوضى و خوارزمية اسراب الطيور . و ايضا سوف يتم توليد معكوس **s–box dynamic** الناتج من مخرج ال **s–box** المتولد بالمقترحات اعلاه و الناتج من اتحاد السطر و العمود لكل قيمة موجودة

*Email: Zainab_albiaty@yahoo.com

بجدول ال **s−box** . و يتم اختبار مخرج ال **s−box** بعدة قياسات متمثلة بالهجوم و الوقت و التعقيد و معيار الانهيار و تظهر النتائج ان اي تغيير في المدخلات سوف يؤدي الى تغيير ناتج ال **s−box** و كذلك سوف يتم اختبار خوارزمية **AES** المحسنه بالاختبارات الاحصائية الخمسة و اختبارات ال **NIST** و تظهر جميع النتائج ناجحة واما الوقت فلا يستغرق سوى بعض الثواني و هو مقارب الى الوقت المستغرق للخوارزمية الاصليه .

1. INTRODUCTION

Textual information plays huge role in all aspects, both in our lives and personal life. Thanks to increase in utilizing intrusive programs in recent years, the need to improve encryption algorithms because most of the modern encryption algorithms have been expose to successive attacks[1]. In November 2009, AES algorithm exposure was attacked 8 rounds type attack (distinguish attack)[2],however the modern algorithm worked on a principle confusion and diffusion , confusion worked substitution to generate S-BOX e.g. AES and diffusion worked permutation to generate initial permutation e.g. DES because principle confusion used AES algorithm has resistance against this type of attack . Moreover in 2011, the algorithm came under attack (biclique attack) like a brute force attack [3], four successive attacks are projected simultaneously, and despite the many attacks, the algorithm has steadfastly resisted many of the attacks [4]. The length of the plaintext in AES algorithm is 128 bits and the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.Each round consists of four operations sub-byte, ShiftRow, MixColumn and AddRoundKey except the last round does not contain MixColumn operation [5]. The main purpose of this research is to use chaos theory to generate random S-BOX and utilize particle swarm optimization to update all values in state to generated S-BOX [6].

Related Work

There are many proposed algorithms that deal with data encryption based on chaotic

1- In 2011, Jolfaei, and Mirghadri [7]: The chaos is used to expand the diffusion and confusion in the images because it is sensitive to the initial conditions barker's maps are used to design dynamic permutation map and S-box, The proposed algorithm is measured through a series of tests, including visual testing, graph analysis, randomization test, information interruptions, coding quality, and correlation analysis. The results are very good for 10 round and data block 128 bits.

2- In 2013, Arrage,Hamdoun , Tragha and khamlich [8]: Generates S-box based on the master key (new $s-box_{xor}$ where each cell in S-box XOR with selected byte from master key),($s-box_{xor}[x,y] = S-box\,AES[x,y] xor\, key[i]$; in decode take the S-box result from encryption and made XOR with Selected same byte from key-key[i]. This work shows us new features of the S-BOX table that prevents the attack from parsing the encryption algorithm. In addition, the implementation only consumes a few milliseconds, and the encryption will be performed with high performance for confusion and diffusion due to the complexity of the algorithm.

3- In 2015 , Abulgader, Ismail, idbeaa, and Zainal[9]: A method is proposed to overcome the weaknesses in S-BOX and improve the performance of AES by replacing the Mix phase with the chaotic system , the XOR process to reduce the high computations of the Mix column and generated S-box base on the chaotic system . The results show that the proposed method enables to generate an efficiently encrypted image with very low correlation coefficients of the adjacent pixels and provide high speed of operation.

4- In 2016 , Alabaichi[10]: This work focuses on encrypting colored images using 3D maps Chebyshev to generate secret keys for the purpose of image diffusion and using 2D maps Arnold cat map to generate the S-box using XOR function with the old S-box. The proposed algorithm is tested using NPCR, UACI, information entropy, The results show that the algorithm resists a different kind of attack

5- In 2017, Kamel and Farhan [11]: A new block cipher is designed as a moving structure based on two of the proposed algorithms. The first algorithm uses the function of complementary DNA, shift, two layers of the s-box, In the second algorithm it consists of the degree of shift and the addition of DNA .In the implementation of the engine, it will generate a secret key through the chaotic generator agreed between the sender and the recipient and the text was tested encrypted

through several measurements and the five statistical analysis, NIST, where the advantages of the tests for all tests and high efficiency.

2. SUBSTITUTION (S-BOX)

In the AES algorithm, there is an array known as S-BOX whose size is a byte. The number of possibilities multiplied by 256 probabilities. Each value in the table represents a single byte of 8 bits, four of which are the value of the row (left most). The other four bits represent the column value(right most), For instance, the hexa-decimal value {45} references row 4, column 5 .S-box Is supported multiplication inverse for a certain number in GF (2^8) = GF (2)[x]/ ($x^8 + x^4 + x^3 + x^1 + 1$) with the used affine transformation ,As shown in Figure-1:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Figure 1- transformation affine [12]

AES defines a matrix of byte values, known as S-box, which includes a permutation of every possible 256, 8-bit values. Every distinct byte of State is mapped to a new byte. The leftmost 4 bits are utilized as a row value and the rightmost 4 bits are utilized as a column value. Those row and column values play the role of indexes into the S-box for selecting a unique 8-bit output value. For instance, the hexa-decimal value {95} references row 9, column 5

TABLE 1-S-BOX IN AES [12]

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

The Rijndael S-Box designed to be immune against differential and linear attacks this gives the strength of symmetric key algorithms. [12]

3. CHAOS THEORY

The chaos maps emphasize the investigation of nonlinear systems where they are sensitive to the initial conditions [13]. Changing the input will result in a change output, providing improved security levels of cryptographic algorithms and the emergence of cryptographic chaos theory in the 1980. Many researchers (Srividya & Nandakumar, 2011) have noted the appropriateness of chaos maps to develop cryptographic techniques because they generate almost random numbers, but in fact they are conducted under random organized behaviour. Where maps of chaos 1D and 2D are better because they provide a high degree of security and better random availability [14], and will be adopted one-dimensional and two-dimensional equations of Lorenz equations in this research.

4. PARTICALE SWARM OPTIMIZATION (PSO)

PSO can be described as one of the Meta-Heuristic algorithms developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. Algorithm attributed to the scientist Kennedy Eberhart is the first of the goal of social behavior, PSO which refers to the method of arithmetic improves the problem through a repeat attempt to improve the solution of a candidate it solves a problem through a number of good solutions . The algorithm has been used in many areas including training of neural networks, image recognition, and improvement of electrical power networks, biomechanics processes, and diagnosis of diseases. The concept PSO is utilizing a certain number of particles which determine a moving squadron in place to locate best solutions. Its advantages are simple to implement, contain a few parameters not sensitive to the expansion of variables and are very effective global algorithms [15] [16].

The parameters used for the particle swarm algorithm

Swarm: collection of particles (S)
Particle: possible solution
Position:
Velocity:
Every particle maintains
Individual best position (PBest)
Swarm maintains its global best (Gbest)
C1 is the importance of personal best value
C2 is the importance of neighborhood best value
Rand: random variable

1- Algorithm PSO[15]
Input: swarm complete, P particle ,V velocity , Bbest,Gbest,c1=2,c2=2, r1,r2
Output: Improved location detection to guide swarm movements

Begin
1- Forming the solution space by creating a complete swarm
2- For each P
Calculate fitness value
If the fitness value is better than the best fitness value (pBest)
set current value as the new pBest
end
3- Choose the particle with the best fitness value of all the particles as the GBest
4- Update the velocity and location of each particle in swarm
v = v + c1*rand*(pBest – p) + c2*rand*(GBest – p);
p = p + v;
5- go to step 2
End

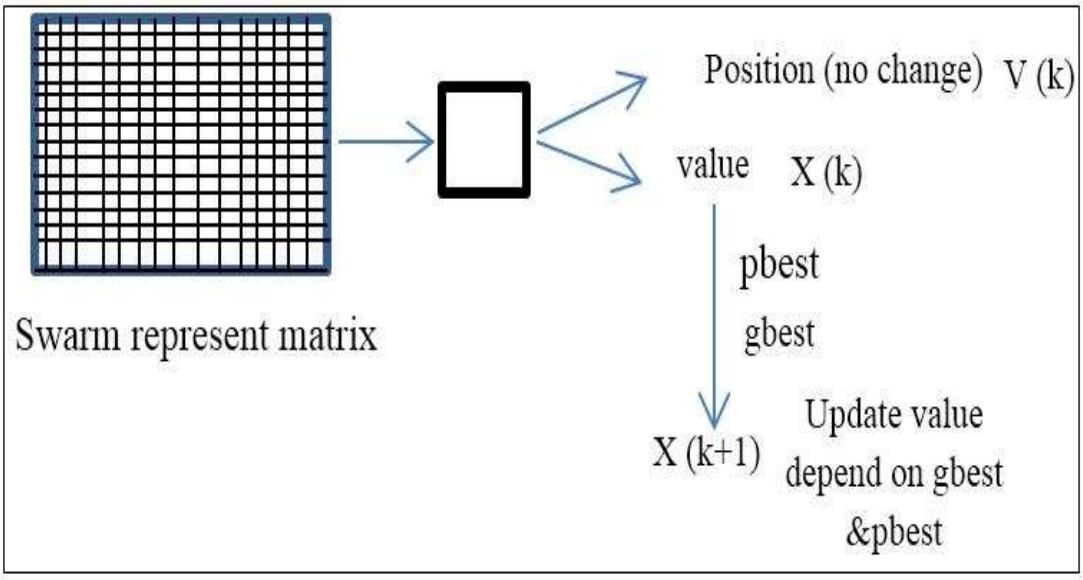


Figure 2-Represent Swarm in S-box

x(k) is a place in s-box contain value and position
xk+1 update value
vk position
vk+1 is not change
pbest summation (position and value)
gbes large value in s-box

5. PROPOSED ALGORITHM TO DYNAMIC S-BOX

A. S-Box Generation

Due to the exposure of modern algorithms to repeated attacks, it is necessary to reinforcement the strength of algorithms in order to be steadfast against any attacks that may be encountered. Since the S-BOX table is fixed, making it vulnerable to attack, the algorithm will be improved by making the dynamic S-BOX as shown below:

Step 1:insert key (8 char) and made expansion key to 32 char by take ASCII code key and convert to binary in the same time generate state contain 24 random number by 1D logistic map and using (XOR between key and random number) The Lorenz equation 1D used to generate the random numbers is the following :

$$x(i+1) = \mu * x(i) * \big(1 - x(i)\big) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

[8]

The parameters values initial x(i) ∈(0,1) and μ ∈(0,4) , These values have been identified by theorists of chaos theory .also convert to binary and made XOR between key binary with first 8 number randomness binary and the result made XOR with second 8 number randomness binary and result made XOR with third 8 randomness numbers binary logistic map

Step 2: Convert key 32 char to binary get state 2D values 256bites.

Step 3: Convert bites state above step to integer state by using shift and rotate.

Step 4: Generate state 2D contain random number by using chaotic logistic map 2D

$$x(i+1) = \Big(\mu 1 * x(i) * \big(1 - x(i)\big)\Big) + (y1 * Math.Pow(y(i), 2) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)[15]$$

$$y(i+1) = \mu 2 * y1 * \big(1 - y(i)\big) + y2 * \big(Math.Pow(x(i), 2) - x(i)\big) \dots\dots\dots\dots\dots\dots\dots\dots\dots (3)[15]$$

Where initial values (x, y ϵ 0, 1) and (μ 1=3.5, μ 2 =3.6) take randomly between (3.1to 3.6) take only two digits after the comma (Here the researcher withdraws one or two or three grades after the comma according to the needs of the bits here we need only two digits after the comma to generate 8 bits only)

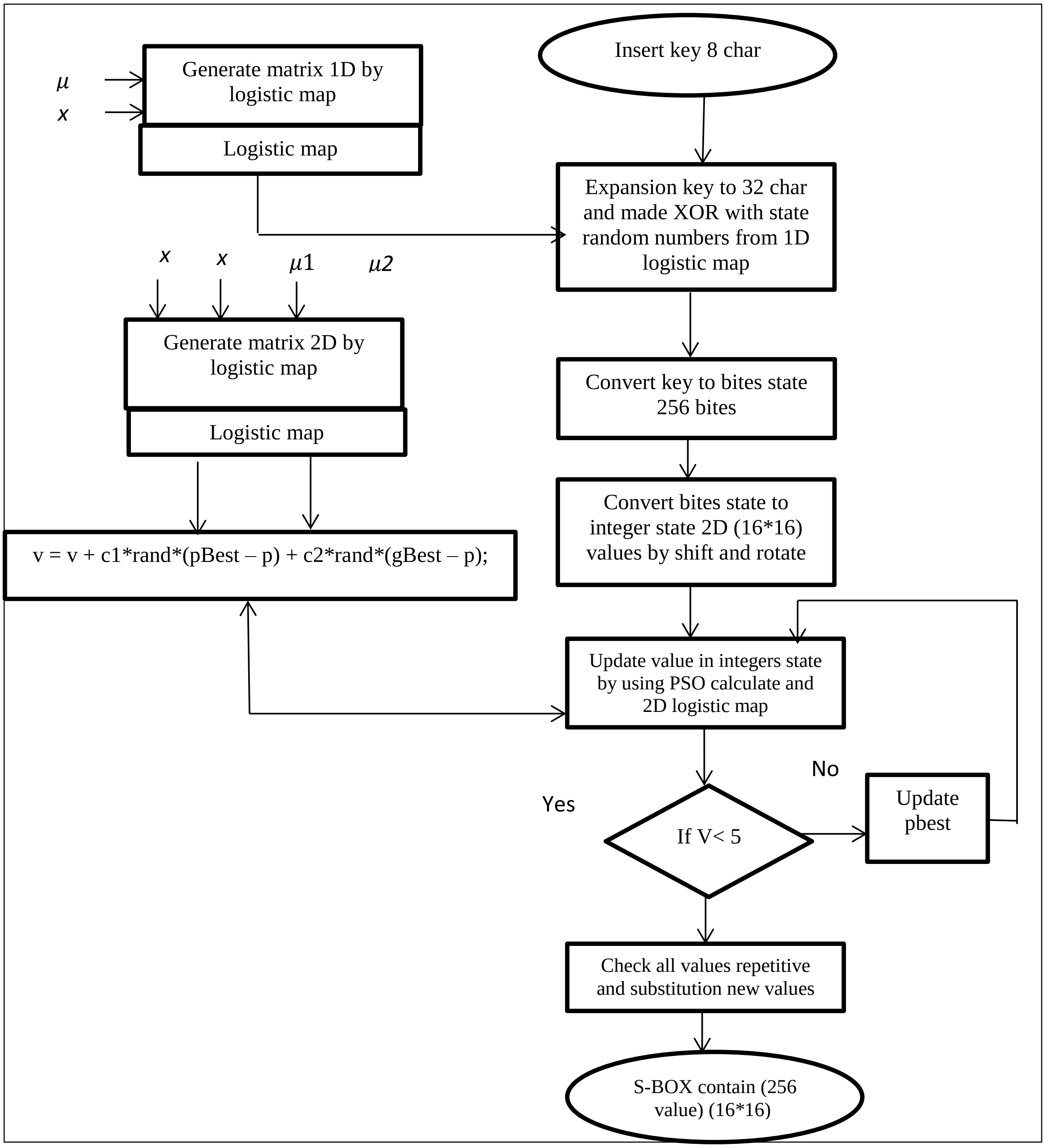
Step 5: The principle of the PSO algorithm is presented on the matrix resulting from the previous step the swarm will represent the resulting matrix. Each place of the matrix will represent the particle so that each place has a value and a position. In the proposed algorithm the values will be modified only and the position will remain unchanged by depend on the calculation used in the PSO algorithm

$$V = V + C1 * rand1 * (pBest - p) + C2 * rand2 * (gBest - p) \quad \ldots (4)$$

swarm represent matrix contain on V represent each place in matrix 2D, C1,C2 constant , gBest max value in matrix shift and rotate, R1, R2 random number result from 2D logistic map and pBest represent summation position P and value V and update depend on fitness if the result V < 5 adding V with pBest

With gBest represent pBest update to place and repeated step 5 and if V>=5 put value in matrix place without change. **(The value of the V is determined as a secret value)**

Step 6: All repeated values in the output of the previous step are replaced by a table whose numbers are arranged randomly and which consists of 256 numbers .as shown in Figure-3.



Example

1- Expansion Key:

- key= (**sciences**). Convert to ASCII code =(1159910510111099101115)
- convert ASCII code key to binary (01110011011000110110100101100101011011100110001101100101011100 11)Convert first 8 random number 1D logistic map (35,23,90,56,41,87,58,27) to binary =(00100011000101110101101000111000001010010101011100111010000110 11)
- And made XOR between binary key and binary logistic map random numbers the result = (**Pt3]G4_h**).

• Also take (Pt3]G4_h) binary made XOR with binary second random number (13,99,25,90,95,67,14,73) and the result ( ]┤*•↑wQ!)

• Last take binary (]┤*•↑wQ!) made XOR with binary third random number (96,61,29,82,74,42,76,98) and the result (=wJgx┤1A) the finally expansion key= (sciencesPt3]G4_h]┤*•↑wQ!=wJgx┤1A)

2-Convert expansion key (sciencesPt3]G4_h]┤*•↑wQ!=wJgx┤1A) to bites state

TABLE 2- CONVERT KEY TO BITS STAT

0	1	1	0	0	0	1	1	0	1	1	0	0	0	1	1
0	1	1	0	1	0	0	1	0	1	1	0	0	1	0	1
0	1	1	0	1	1	1	0	0	1	1	0	0	0	1	1
0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	1
0	1	0	1	0	0	0	0	0	1	1	1	0	1	0	0
0	0	1	1	0	0	1	1	0	1	0	1	1	1	0	1
0	1	0	0	0	1	1	1	0	0	1	1	0	1	0	0
0	1	0	1	1	1	1	1	0	1	1	0	1	0	0	0
0	1	0	1	1	1	0	1	0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	0	0	0	0	0	0	1	1	1
0	0	0	1	1	0	0	0	0	1	1	1	0	1	1	1
0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	1
0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	1
0	1	0	0	1	0	1	0	0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0	0	0	0	1	0	1	1	1
0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	1

3-Convert bits state to integer matrix 2D (16*16) contain 256 values by shift and rotate , take 16 bites from above state convert to integer and made shift 16 bites to left 15 times to get another fifteen integer numbers

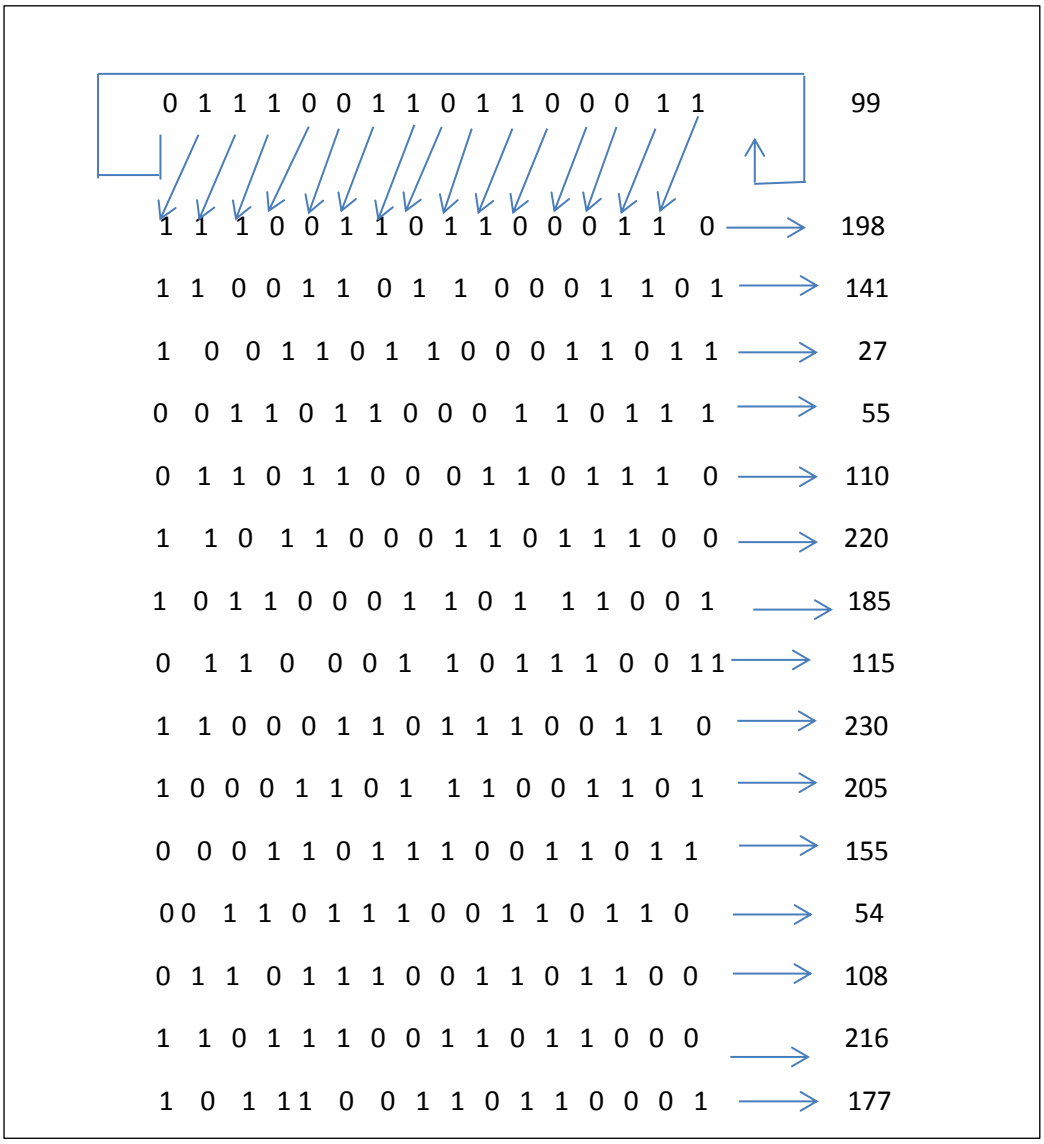


Figure 4-Generate 16 Numbers from 16 Bits by Shift and Rotates

TABLE 3-CONVERT TO INTEGER STATE

99	198	141	27	55	110	220	185	115	230	205	155	54	108	216	177
101	202	149	43	86	173	90	180	105	210	165	75	150	44	89	178
99	198	141	27	54	109	219	183	110	220	185	115	230	204	152	49
115	230	205	155	54	108	217	178	101	202	149	43	87	174	92	185
116	232	209	162	69	138	20	40	80	160	65	131	7	14	29	58
93	186	116	233	211	166	76	153	51	102	205	154	53	107	215	174
52	104	209	162	86	163	17	35	71	142	28	57	115	230	205	154
104	208	161	66	133	11	23	47	95	190	125	251	246	237	218	180
23	46	93	186	117	235	215	174	93	186	116	232	209	162	69	139
7	14	28	57	114	229	202	149	42	84	168	80	160	64	129	3
119	238	220	184	113	227	198	140	24	48	97	195	135	14	29	59
33	66	133	10	21	42	84	168	81	162	68	137	18	36	72	144
119	238	220	185	115	231	207	158	61	122	245	235	215	174	93	187
103	206	157	68	116	233	210	165	74	148	41	83	166	76	153	51
23	46	93	187	119	239	222	188	120	240	224	192	129	2	5	11

65	130	4	9	19	38	76	152	49	198	197	138	20	40	80	160

4-Generate matrix 2D contain 256 random numbers

$$x(i+1) = \Big(\mu 1 * x(i) * \big(1 - x(i)\big)\Big) + (y1 * Math.Pow(y(i), 2)$$

$$y(i+1) = \mu 2 * y1 * \big(1 - y(i)\big) + y2 * \big(Math.Pow(x(i), 2) - x(i)\big)$$

Where initial values (x, y ϵ 0, 1) and (μ 1=3.5, μ 2 =3.6) take only two numbers after the comma

5-Update each value result from step 3 by using PSO and random number results from chaotic 2D logistic map from step 4

$V = V + C1 * rand1 * (pBest - p) + C2 * rand2 * (gBest - p)$; If the update value is less and equal than 5 is it summation with gBest and represent update value pBest and are repeated step 5 else value largest than 5 put directly in state finally

TABLE 4- UPDATE VALUES BY PSO AND 2D LOGISTIC MAP

29	80	223	209	19	254	6	147	169	68	85	241	254	246	15	55
43	28	23	129	24	129	56	24	55	112	173	252	94	54	59	134
29	80	223	209	184	65	223	149	112	218	129	249	174	150	202	183
141	176	159	81	184	132	145	26	163	232	157	161	207	84	78	175
148	22	187	254	13	170	142	164	26	94	137	41	127	244	191	126
243	236	176	107	135	86	22	179	105	196	85	158	77	167	89	74
212	150	187	254	178	48	25	41	77	108	244	43	43	156	111	30
64	78	107	222	205	231	254	29	197	156	5	17	190	197	108	36
9	8	143	6	29	71	67	30	123	216	204	232	41	160	103	125
153	168	72	219	12	217	72	183	156	210	192	160	72	98	227	133
169	72	136	176	177	95	172	64	253	238	169	233	255	244	191	205
79	255	167	118	253	202	78	36	191	64	156	27	26	190	122	8
179	72	136	91	103	83	11	46	219	152	253	225	79	84	255	77
57	232	79	134	194	205	128	167	60	18	241	153	110	22	123	85
9	8	143	177	211	59	84	16	226	174	120	240	217	64	39	253
47	196	160	203	71	214	22	52	31	251	205	110	124	250	253	248

6-Delete all repetitive values in above step 5 and replace new values not previously present based on a secret table consisting of 256 values without repetition.

TABLE 5- NEW DYNAMIC S-BOX

1D	50	DF	D1	13	FE	6	93	A9	44	55	F1	66	F6	0A	37
2B	1C	17	81	18	F2	38	E0	EB	70	AD	FC	5E	36	3B	86
0E	4B	BA	B6	B8	41	E4	95	B9	DA	42	F9	AE	96	CA	B7
8D	B0	9F	51	DC	84	91	1A	A3	E8	9D	A1	CF	54	4E	AF
94	16	B8	1	0D	AA	8E	A4	97	E7	89	29	7F	F4	BF	7E
F3	EC	71	6B	87	56	75	B3	69	C4	DD	9E	4D	A7	59	4A
D4	58	A6	23	B2	30	19	AB	00	6C	C8	2A	52	9C	6F	1E
40	2C	ED	DE	CD	E7	5C	68	C5	4C	5	11	DE	B4	25	24
9	8	8F	20	64	47	43	3F	7B	D8	CC	6A	C3	A0	67	7D
99	A8	48	DB	0C	D9	46	3E	61	D2	C0	3	D5	62	E3	85
6D	31	88	8B	B1	5F	AC	60	FD	EE	D0	E9	FF	4	82	CE
4F	57	A2	76	7	28	21	92	3A	E6	C9	1B	A5	90	7A	33
77	14	5A	5B	53	E5	0B	2E	65	98	72	E1	2	26	B5	C7
39	C1	F5	73	C2	15	80	74	3C	12	35	8C	6E	C6	8A	9A
22	EF	32	45	D3	79	BD	10	E2	2D	78	F0	49	3D	27	D7
2F	0F	9B	CB	EA	D6	BC	34	1F	FB	5D	83	7C	FA	63	F8

B .INV S-Box Generation

The inverse S-BOX is generated from the union of each row and column for each value generated in the Table-6.

TABLE 6- INV S-BOX

68	43	CC	9B	AD	7A	06	B4	81	80	0E	C6	94	44	20	F1
E7	7B	D9	04	C1	D5	41	12	14	66	37	BB	11	00	6F	F8
83	B6	E0	63	7F	7E	CD	EE	B5	4B	6B	10	71	E9	C7	F0
65	A1	E2	BF	F7	DA	1D	0F	16	D0	B8	1E	D8	ED	97	87
70	25	2A	86	09	E3	96	85	92	EC	5F	21	79	5C	3E	B0
01	33	6C	C5	3D	0A	55	B1	61	5E	C2	C3	76	FA	1C	A5
A7	98	9D	FE	84	C8	0C	8E	77	58	8B	53	69	A0	DC	6E
19	52	CA	D3	D7	56	B3	C0	EA	E5	BE	88	FC	8F	4F	4C
D6	13	AE	FB	35	9F	1F	54	A2	4A	DE	A3	DB	30	46	82
BD	36	B7	07	40	27	2D	48	C9	90	DF	F2	6D	3A	5B	32
8D	3B	B2	38	47	BC	62	5D	91	08	45	67	A6	1A	2C	3F
31	A4	64	57	7D	CE	23	2F	24	28	22	42	F6	E6	7C	4E
9A	D1	D4	8C	59	78	DD	CF	6A	BA	2E	F3	8A	74	AF	3C
AA	03	99	E4	60	9C	F5	EF	89	95	29	93	34	5A	73	02
17	CB	E8	9E	26	C4	B9	75	39	AB	F4	18	51	72	A9	E1
EB	0B	15	50	4D	D2	0D	49	FF	2B	FD	F9	1B	A8	05	AC

5. Experimental Results

The proposal new S-BOX will be tested through several metrics including (Balanced, Avalanche Criterion,complexity , time , Statistical Tests and NIST)

1-Avalanche Criterion: Changing the amount of input will completely affect the amount of output as the value of the avalanche ranges from [0, 1], and are calculated from the following equation

$$\text{Avalancheeffect}=\frac{\text{number of flipped bits in (output)ciphertext}}{\text{number of all bits in (output)ciphertext}} \quad \ldots\ldots (5)[17]$$

By applying the equation note the signification difference between the output of Table-(5, 6) when change key and value parameters in chaos and PSO so that the difference reaches to 99%.

Number of all bites in (output) ciphertext: Is the number of bits generated from the table (S-box) and equal 256 value.

Number of flipped bites in (output) ciphertext: Is the difference in the order of bits generated from the proposed S-box table from the traditional s-box.

Table-4 shows the difference in the order of the resulting values for the proposed S-box for the traditional S-box is equal 254.

Avalanche effect= 254/256

=0.99

2-Balanced: Is to achieve the principle of balance as the number of ones is near to equal number of zeros

3-Complexity: Through the generated results we observe the complexity in S-BOX, Where the expansion phase of the key is very complex because it will depend on the equation (1) 1D logistic map .Where changing the key or changing input equation (1)1D logistic map will lead to change expansion key output. For example when used same key in above example (sciences) but change value parameters $x = 0.9$ and $\mu = 0.3$ in logistic map notice the difference expansion key output (sciences8K,T-V'Dysmmmog}93---/'=) and when change key and used the same value parameters

see the different expansion key (security|RLOTdKPF9zfbSx;V'#). as well as when changing the value
parameters equation 2D chaos theory and value parameters equation PSO lead to S-BOX result is different from in Table -5.

TABLE 7-NEW DYNAMIC S-BOX2

4B	5E	9F	39	25	30	B8	83	A3	D4	99	6F	EC	18	D3	A4
3B	FE	68	B9	FA	26	3C	8B	33	74	59	EF	93	E6	AF	7C
47	C6	49	EB	E5	B0	2E	D0	FC	52	C5	3E	43	9F	CA	9A
37	66	89	19	A5	C1	57	FD	4A	AA	6D	97	42	2D	6C	72
9E	B	BD	31	15	D5	55	7E	82	5A	0D	92	35	2A	FB	11
DA	98	95	D6	54	AD	E1	96	32	3A	5F	88	A7	F2	38	27
1C	64	65	B3	F4	F7	61	A1	CB	BC	C9	8A	85	2C	DF	CC
F3	56	71	6E	1A	58	5B	0A	A2	AB	C8	D2	E	C3	4E	E4
DE	B6	F1	76	BF	44	80	94	CE	62	E8	28	DC	ED	90	B1
77	4F	75	7A	3D	DB	E3	4D	48	36	A0	DD	40	B5	1E	12
41	CF	22	69	6	F5	17	7F	34	AE	BB	1	86	24	C7	84
8E	50	A6	8D	EA	46	9B	9	9D	F6	7D	67	79	23	5C	E9
14	87	B7	D8	78	B4	45	16	9C	3	1D	F0	70	D9	C0	91
F8	0C	E0	10	8	21	73	BA	2F	8C	29	8F	53	CD	E7	3F
E2	D1	20	1B	2	4	7B	1F	51	FF	A8	6A	C4	81	2B	7
00	B2	C2	13	EE	BE	6B	A9	0F	4C	5D	5	63	60	AC	D7

4-Time: The time taken for implementation is small and needs only a few millisecond. (The time was calculated through the timer for the Visual Basic .NET program)

TABLE 8-TIME TAKEN TO GENERATE S-BOX

User key	S-BOX Execution time
computer	0.133 millisecond
sciences	0.135 millisecond

5- Five Statistical Tests
The output of the ciphertext will be tested the five tests will be successful and shown in the table below

TABLE 9-STATISTICAL TEST OF PROPOSAL AES

Statistical Test			
Tests		**Freedom Degree**	**Proposal AES**
Frequency Test		**Must be <=3.84**	**Pass =0.141**
Run Test	**T0**	**Must be <=10.788**	**Pass= 8.158**
	T1		**Pass= 7.88**
Poker Test		**Must be <=11.1**	**Pass = 1.513**
Serial Test		**Must be <=7.81**	**Pass= 0.188**
Auto Correlation Test Must be <=3.84		**Shift No.1**	**Pass = 0.475**
		Shift No.2	**Pass = 2.268**
		Shift No.3	**Pass =1.142**

	Shift No.4	**Pass =1.286**
	Shift No.5	**Pass = 0.673**
	Shift No.6	**Pass =0.576**
	Shift No.7	**Pass =2.920**
	Shift No.8	**Pass =0.581**
	Shift No.9	**Pass=1.785**
	Shift No.10	**Pass=1.626**

6- NIST Test

The NIST Test Suite a statistical consisting 16 tests and running binary sequence. Testing encryption output text of encryption algorithms. Theses test based on randomizations can found sequence such as Linear Complexity, Discrete Fourier Transform …..Etc. The following Table-10 is show the result of the proposal TABLE-10

TABLE 10

NO	Test	Proposal AES
1	Frequency	Success
2	Block Frequency	Success
3	Commulative Sums	Success
4	Runs	Success
5	Longest Run	Success
6	Rank	Success
7	Discrete Fourier Transform	Success
8	Non-periodic Templates	Success
9	Overlapping	Success
10	Universal	Discard
11	Approximate Entropy	Success
12	Random Excursions	Test not application
13	Random Excursions Variant	Test not application
14	Serial	Success
15	Lampel –Ziv Compression	Success
16	Linear complexity	Discard

6. CONCLUSION

In this research, S-BOX will be designed based on the key, shifting, chaos theory, and particle swarm optimization artificial intelligence algorithm. We conclude from this research that changing one byte in the key will change the key expansion process, change the parameters of the logistics map (1D, 2D), Also, changing the fitness of the PSO algorithm will affect the output S-BOX this refers to the complexity in the generation of S-BOX and the most important point is the dynamic S-BOX is not static but depends on the input and this will indicate the security generated in the generation. Thus we conclude that AES improved algorithm, which was tested by the five statistical tests and the NIST program, has efficient results and the time it takes to generate the S-BOX only needs a few milliseconds to refer in table VIII It is almost the same time used for the original algorithm.

References

1. Elena, A, Andrey B. and Bart, M. **2013.** "*Towards Understanding the Known-Key Security of Block Ciphers*", IACR 2013, IACR, to Springer-Verlag on April 29.
2. Englund, H.**. 2007.** "Some Results on Distinguishing Attacks on Stream Ciphers" Ph.D. Thesis, December 14, 2007, Department of Electrical and Information Technology Lund University, ISBN: 91-7167-046-7
3. Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, Adi Shamir. **2009**-08-19. "Key Recovery Attacks of Practical Complexity on AES Variants with up To 10 Rounds", University of Luxembourg, Luxembourg, Ecole Normale Superieure, Einstein Institute of Mathematics, Hebrew University, Faculty of Mathematics and Computer Science the Weizmann Institute Rehovot, LNCS 6110, pp. 299–319, Archived from the original on 28 January 2010. Retrieved 2010-03-11.
4. Andrey Bogdanov; Dmitry Khovratovich & Christian Rechberger. **2011**. "Biclique Cryptanalysis of the Full AES" (PDF), Microsoft Research Redmond, USA; ENS Paris and Chaire France Telecom, France. Archived from the original (PDF) on August 31, 2011.
5. Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197, 2001
6. Zhan, Z-H. And Zhang, J. and Li, Y. and Chung, H.S-H. **2009.** "Adaptive particle Swarm Optimization". *IEEE Transactions on Systems Man, and Cybernetics — Part B: Cybernetics*, **39** (6). pp. 1362-1381. ISSN 0018- 9472, (2009)
7. Alireza Jolfaei, Abdolrasoul Mirghadri. **2011.** "Image Encryption Using Chaos and Block Cipher", *Computer and Information Science*, **4**(1): 172-185.
8. Sliman Arrag,Abdllatif Hamdoun , Abderrhim Traga and Salah eddine. **2013**. "Implementation of Stronger AES by Using Dynamic S-box dependent of Master Key ", E-ISSN: 1817-3195 , **53**(2), 20th July 2013.
9. Ali Abdulgader. Mahamod Ismail, Nasharuddin Zainal, Tarik Idbeaa. **2015**. Enhancement of AES Algorithm based on Chaotic Maps and Shift Operation for Image Encryption", E-ISSN:, *Journal of Theoretical and Applied Information Technology*, **71**(1): 1817-3195, 10th January 2015.
10. Ashwaq Mahmood Alabaichi **2016. "**Color Image Encryption using 3D Chaotic Map with AES key Dependent S-Box" , *IJCSNS International Journal of Computer Science and Network Security*, **16**(10).
11. Kamel sh.H. And Farhan a.k.**2017**."Proposal Dynamic Block Cipher Structure Depend on Secret Map ", Department Computer Sciences, University of Technology, 2017.
12. William Stallings **2011.** "*Cryptography and Network Security Principles and Practice*", FIFTH EDITION, prentice Hall.
13. Chittaranjan Pradhan1 and Ajay Kumar Bisoi **2013.** "Chaotic Variations of AES Algorithm ", *International journal of chaos, control modeling and simulation (IJCCMS*), **2**(2), June.
14. Naif B. Abdulwahed **2013.** "Chaos –Based Advance Encryption Standard", King Abdullah University of Science and Technology, May.
15. Kennedy, J. and Eberhart, R, **1995.** "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948. Doi: 10.1109/ICNN.1995.488968.
16. Poli, R. **2007.** "An Analysis of Publications on Particle Swarm Optimisation Applications" (PDF). Technical Report CSM-469. Department of Computer Science, University of Essex, UK.
17. Vikas Kaula*, Bhushan Nemadeb, Vinayak Bharadic, S. K. Narayan khedkard, **2016.** "Next Generation Encryption using Security Enhancement Algorithms for End to End Data Transmission in 3G/4G Networks" 7th International Conference on Communication, Computing and Virtualization, 2016.