Proposal New S-box for AES Algorithm depend on A.I Bee Colony

Dr. Alaa Kadhim
Computer Science Department, University of Technology/ Baghdad.
Sura Khalaf
Computer Science Department, University of Technology/ Baghdad.
E-mail:Surakhallaf13@gmail.com

Received on:8/5/2014 & Accepted on: 4/12/20 14

ABSTRACT

The AES algorithm, also called the Rijndael algorithm, is asymmetric block cipher, where the data is encrypted/ decrypted in blocks of 128 bits. Each data block is modified by several rounds of processing, where each round involves four steps. Three different key sizes are allowed: 128 bits, 192 bits, or 256 bits, and the corresponding number of rounds for each is 10 rounds, 12 rounds, or 14 rounds, respectively. From the original key, a different "round key" is computed for each of these rounds. The single nonlinear step is the Sub Bytes step, where each byte of the input is replaced by the result of applying the "S-box" function to that byte. This nonlinear function involves finding the inverse of the 8-bit number, considered as an element of the Galois field GF (2^{16}). The Galois inverse is not a simple calculation, and so many current implementations use a table of the S-box function output. This table look-up method is fast and easy to implement. S-box is influenced by linear and differential cryptanalysis and also interpolation attacks. In this paper intended a new approach for the design of s-box based on bee colony algorithm to increase the power of s-box and enhanced resistance against attacks through the use of artificial intelligence algorithms.

أقتراح S-Box جديد لخوارزمية AES بالاعتماد على مستعمرات النحل في الذكاء الاصطناعي

الخلاصة

خوارزمية AES ، وتسمى أيضا خوارزمية Rijndael ، يتم تشفير كتلة متماثلة، حيث البيانات يتم تشفير/فك تشفير في كتل من128 بت. كل كتلة بيانات هو تعديل بعدة جولات للمعالجة، حيث تتضمن كل جولة أربع خطوات. مسموح بثلاثة اطوال مختلفة للمفاتيح : 128 بت،192 بت، أو256 بت، وعدد الدورات تتطابق مع طول المفاتيح وهو 10 ,12,14 دوره ، على التوالي. المفتاح الأصلي، يتم احتساب "round key" مختلف لكل واحدة من هذه الجولات. الخطوة غير الخطية هي خطوة 'sub byte'، حيث يتم استبدال كل بايت من الإدخال بواسطة تطبيق الدالة 'S-Box' الى بايت اخر. هذه الدالة غير الخطية تتضمن معكوس العدد 16-بت، يعتبر عنصرا من عناصر GF (2^{16}). Galois inverse ليست عملية حسابية بسيطة، واستخدام العديد من التطبيقات الحالية لاخراج جدول دالة S-Box. حيث ان هذا الأسلوب سريع وسهل التنفيذ. S-Boxيتأثر بتحليل الشفرات الخطية والتفاضلية، وأيضا هجمات الاستيفاء. في هذه البحث يقصد اتباع نهج جديد لتصميم S-Box استناداً إلى

https://doi.org/10.30684/etj.33.1B.2
2412-0758/University of Technology-Iraq, Baghdad, Iraq

خوارزمية مستعمرة النحل. لزيادة قوة S-Box وتعزيز المقاومة ضد الهجمات عن طريق استخدام خوارزمية الذكاء الاصطناعي.

Keywords: Encryption, Encryption, Artificial Bee Colony, S-Box, Polynomial, Galois field.

INTRODUCTION

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. While cryptography is the science of securing data, cryptanalysis's the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called attackers. A block cipher is a method of encrypting text (to produce cipher text) in which a cryptographic key and algorithm are applied to a block of data (for example, 64 contiguous bits) at once as a group rather than to one bit at a time. The main alternative method, used much less frequently, is called the stream cipher.

So that identical blocks of text do not get encrypted the same way in a message (which might make it easier to decipher the cipher text), it is common to apply the cipher text from the previous encrypted block to the next block in a sequence. So that identical messages encrypted on the same day do not produce identical cipher text, an initialization vector derived from a random number generator is combined with the text in the first block and the key. This ensures that all subsequent blocks result in cipher text that doesn't match that of the first encrypting [1].

Advanced Encryption Standard

The Advanced Encryption Standard (AES) was specified in 2001 by the National Institute of Standards and Technology [2]. The purpose is to provide a standard algorithm for encryption, strong enough to keep U.S. government documents secure for at least the next 20 years. The earlier Data Encryption Standard (DES) had been rendered insecure by advances in computing power, and was effectively replaced by triple-DES. Now AES will largely replace triple-DES for government use, and will likely become widely adopted for a variety of encryption needs, such as secure transactions via the Internet. A wide variety of approaches to implementing AES have appeared, to satisfy the varying criteria of different applications. Some approaches seek to maximize throughput others minimize power consumption and yet others minimize circuitry. For the latter goal, Rijmen suggested using subfield arithmetic in the crucial step of computing an inverse in the Galois Field of 256 elements—reducing an 8-bit calculation to several 4-bit ones. Satoh et al. [3] further extended this idea, using the "tower field" approach of Paar, breaking up the 4-bit calculations into 2-bit ones, which resulted in the smallest AES circuit to date. Rijndael round consists of four different stages

Sub Byte transformation: (S-box substitution) provides non linearity and confusion, constructed by multiplicative inverse and affine transformation.
Shift Row: (rotations) provides inter-column diffusion where the bytes in the last three rows of the states are cyclically shifted.

Mix Column: (linear combination) provides inter-byte diffusion where each column vector is multiplied by a fixed matrix. The bytes will be treated as polynomials rather than numbers

Add Round Key: (round key bytes XOR with each byte of the state and the round key) provides confusion .

The encryption process begins with an Addroundkey stage, and followed by nine rounds of Sub Bytes, Shift Rows, MixColums and Addroundkey transformation. The transformation will be performed respectively and iteratively (Nr times) depending on the key length. The final round will only include 3 stages; Sub Byte, Shift Rows and Addroundkey. All of the operations are byte- oriented. The encryption and decryption structure consists of several transformation stages as shown in Fig. 1. The decryption is essentially the same structure as encryption, but Sub Byte, Shift Row and MixColums are replaced by their inverses; InvSubBytes, InvShiftRows, InvMixColumns, and Addroundkey. It is the reverse order of the encryption structure.

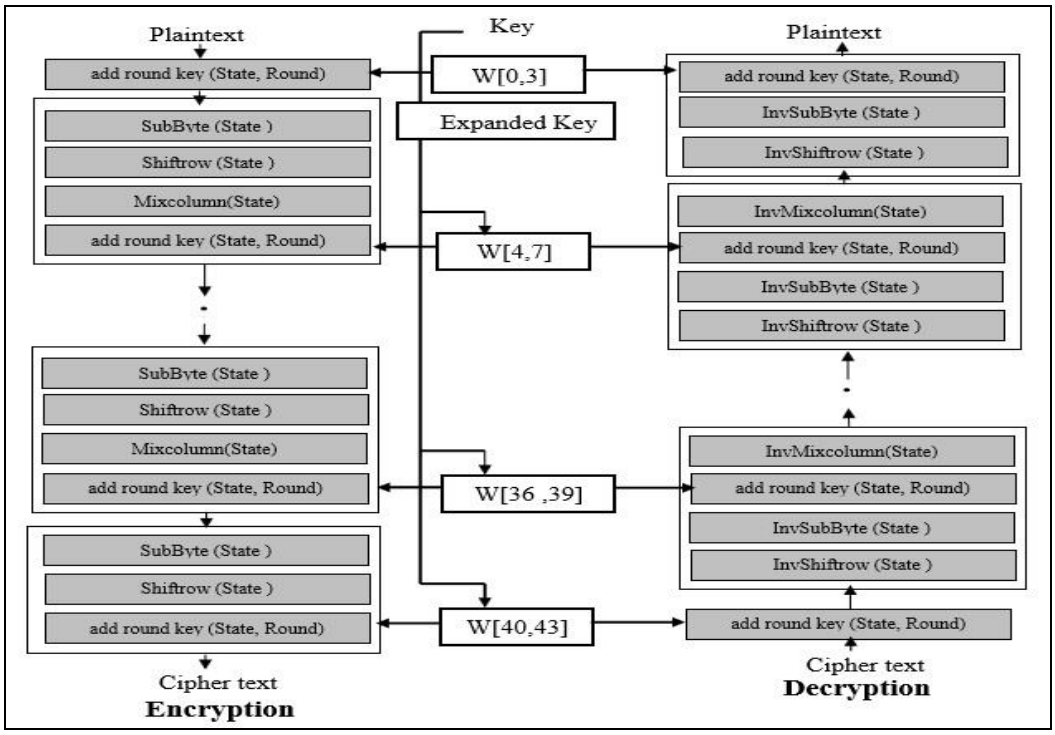


Figure (1): AES Encryption and Decryption

Attacks of Block Cipher

Cryptographic attacks are designed to subvert the security of cryptographic algorithms, and they are used to attempt to decrypt data without prior access to a key. They are part of Cryptanalysis, which is the art of deciphering encrypted data. Cryptanalysis and Cryptography (the art of creating hidden writing, or ciphers) form the science of Cryptology. Differential cryptanalysis and linear cryptanalysis are related attacks used primarily against iterative symmetric key block ciphers. An iterative cipher (also called a product cipher) conducts multiple rounds of encryption using a sub key for each round. Examples include the Feistel Network used in DES and the State rounds

used in AES. In both attacks, a cryptanalyst studies changes to the intermediate cipher text between rounds of encryption. The attacks can be combined, which is called differential linear cryptanalysis.

- One of the most powerful attacks against block ciphers differential cryptanalysis is a potent cryptanalysis technique invented in 1990, by Eli Biham and Adi Shamir based on choosing plaintext attack against DES that was more efficient than brute force search. DC looks for characteristics which are patterns of differences between two chosen plaintext messages that result in specific differences in the corresponding cipher text messages with a high or low probability of occurrence. The main idea in DC is comparing the XOR of two plaintexts with the XOR of corresponding two cipher texts. In DC the input and output difference of the S-boxes are considered to determine a high probability difference which leads to some information about the plaintext difference and the difference of the input to the last round [4].

$$P_1 \oplus P_3 \oplus C_1 = K_2.$$

- The second powerful technique developed in the early in 1993 by Matsui invented linear cryptanalysis presented the technique to create a known plaintext attack to break the FEAL cipher. In 1994 Matsui presented a similar attack on DES. LC studies statistical linear relations between bits of plaintext, cipher text and keys that are encrypted under. These relations are used to predict values of bits of the key when many plaintexts and their corresponding cipher text are known [5]. LC tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, cipher text bits, and it collects algebraic relationships between input, output, and key bits for each round and combining them to form a linear approximation that shows a maximal bias in probability from the value 1/2 allowing to distinguish the cipher from a random permutation. In DES the only non-linear stage is the S-boxes one. All the other operations are linear and can be easily analyzed. In fact, the S-boxes have more features in common with a linear transformation the analyzed. In fact, the S-boxes have more features in common with a linear transformation than one would expect if they were chosen completely at random. Thus, one can be convinced that the DES S-boxes are not optimized against LC n one would expect if they were chosen completely at random. Thus, one can be convinced that the DES S-boxes are not optimized against LC [6]**.**

- The interpolation attacks depend only on the number of S-boxes and number of rounds in the cipher. This attack is independent of the sizes of the S-boxes. Interpolation attacks were the first demonstration of successful polynomial-based algebraic attacks against block ciphers. Interpolation attacks work by expressing the relationship between the plaintext and cipher text for a fixed key as either one or as a vector of polynomials. If the degree of these polynomials is low enough, the coefficients of the polynomials can be interpolated from a number of plaintext/cipher text pairs. A key-dependent equivalent of the encryption or the decryption algorithm has then been determined. In [4] upper bounds on the data complexity-the number of required pairs for known-plaintext interpolation attacks-are given for selected examples. In general, this number increases exponentially with the degree of the polynomial function describing the S-Box, the number of rounds and the number of

elements in the internal state.

Artificial Intelligence Search Algorithms

Artificial intelligence (AI) is the intelligence of machines and the branch of computer science that aims to create it. AI textbooks define the field as 'the study and design of intelligent agents'[7]where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success.[8] John McCarthy, who coined the term in 1955 defines it as 'the science and engineering of making intelligent machines.[9]

The ABC algorithm was firstly introduced by Karaboga [10] for numerical optimization problems based on the foraging behavior of honeybee swarm. Further improvements of the ABC algorithm have been carried out by Karaboga and Basturk [11-12]. In this model, the foraging bees are classified into three different types: employed bees, onlookers and scouts. A bee which has found a food source to exploit is called an employed bee. Onlookers are those waiting in the hive to receive the information about the food sources from the employed bees and Scouts are the bees which are randomly searching for new food sources around the hive. After exploiting a food source, an employed bee returns to the hive and shares the information about the nectar amount of the food source with other bees by dancing in the dance area of the hive. The duration of the dance is proportional to the profitability of the food source. As the quality of a food source enhances, the dancing duration related to this food source increases, making it more probable for an onlooker to choose this source. After watching several dances, an onlooker bee selects a food source and becomes employed. In a similar way, a scout is called employed when it finds a food source. After completely exploiting a food source, the food source is abandoned and all the employed bees change into onlookers or scouts. Karaboga [10] proposed the ABC algorithm inspired by this foraging behavior of honeybees. In this algorithm a food source position is considered as a candidate solution for the optimization problem and the fitness of the solution is represented by the nectar amount of the food source. Similar to the real bee colony, the colony of artificial bees is composed of employed bees, Onlookers and scouts. In the ABC algorithm, half of the colony are employed bees while the other half consists of onlookers, also it is assumed that the number of food sources is equal to the number of employed bees. After abandoning a food source, the employed bee of that food source becomes a scout and carries out a random search. Same as other swarm intelligence based algorithms, the ABC algorithm has an iterative process.

By assuming the number of food sources as NS which is equal to the number of employed or onlooker bees, and D as the dimension of each solution vector, the main steps of an ABC algorithm can be defined as follows:

Step 1: A random population ($X_1, \ldots, X_{NS}$) is initialized, where $X_i = \{x_{1i}, x_{2i}, \ldots, x_{iD}\}$ and each solution vector is generated using :

$$x_{ij} = x_{j\min} + \text{rand}[0, 1] * (x_{j\max} - x_{j\min}) \quad \ldots (1)$$

for $j = 1, 2, \ldots, D$ and $i = 1, 2, \ldots, NS$

Where

x_j max and x_j min respectively represent the upper and lower bounds for the dimension j. After Initialization of the population, the fitness of each food source is evaluated .

Step 2: Each employed bee searches the neighborhood of its current food source to determine a new food source using:

$$v_{ij} = x_{ij} + \phi_{ij} * (x_{ij} - x_{kj}) \quad \dots (2)$$

Where

$k \in \{1, 2, \dots, NS\}$ and $j \in \{1, 2, \dots, D\}$ are randomly chosen indexes. It must be noted that k has to be different from i. ϕ_{ij} is a random number between [-1, 1]. Parameter values produced by Eq. (2) which exceed their boundary values are set to their boundary values [13].

Step 3: After generating the new food source, the nectar amount of it will be evaluated and a greedy selection will be performed. If the quality of the new food source is better than the current position, the employed bee leaves its position and moves to the new food source; in other words, If the fitness of the new food source is equal or better than that of X i, the new food source takes the place of X i in the population and becomes a new member .

Step 4: First an onlooker bee selects a food source by evaluating the information received from all of the employed bees. The probability pi of selecting the food source i is determined by:

$$P_i = \frac{F_i}{\sum_{i=1}^{NS} F_i} \quad \dots (3)$$

Where

f_i is the fitness value of the food source X_i . After selecting a food source, the onlooker generates a new food source using Eq. (2). Once the new food source is generated, it will be evaluated and a greedy selection will be applied, same as the case of employed bees.

Step 5: If a candidate solution, represented by a food source cannot be further improved by a predetermined number of trials, the food source is considered abandoned and the employed bee associated with that food source becomes a scout. The scout randomly generates a new food source using:

$$V_{ij} = x_j \min + \text{rand}[0, 1] * (x_j \max - x_j \min) \quad \dots (4)$$

For j = 1, 2… D

The abandoned food source is replaced by the randomly generated food source. In the ABC algorithm, the predetermined number of trials for abandoning a food source is called limit, also in this algorithm at most one employed bee at each cycle can become a scout

Step 6: If a termination condition is met, the process is stopped and the best food source is reported; otherwise the algorithm returns to step 2.

Proposal New Approach For design S-Box

In this paper new method for design new S-box by using one of the best artificial intelligence searches algorithms. Artificial Bee Colony (ABC) algorithm which is one of the most recently introduced optimization algorithms simulates the intelligent foraging behavior of a honey bee swarm. Clustering analysis, used in many disciplines and applications, is an important tool and a descriptive task seeking to identify homogeneous groups of objects based on the values of their attributes. In this work, ABC is used for data clustering on benchmark problems and the performance of ABC algorithm is compared with Particle Swarm Optimization (PSO) algorithm and other nine classification techniques from the literature. The simulation results indicate that ABC algorithm can efficiently be used for multivariate data clustering. Used ABC algorithm for design new S-Box will be an explanation in the following flowchart

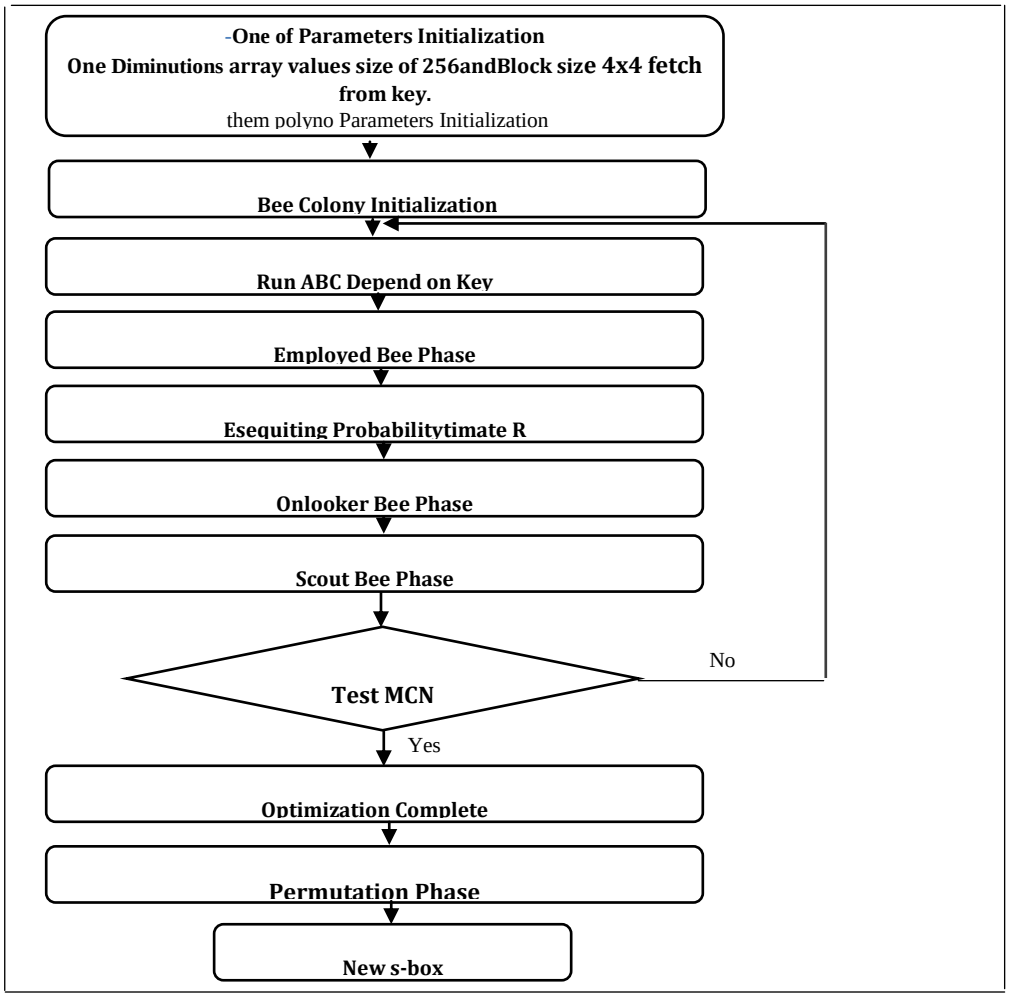


Figure [2]: Structure proposal New S-box

Algorithm of Colony Byte

The procedure described in the following seven steps:

Input:

- Block size 4x4 fetch from key.
- One Diminutions array values size of 256.

Output:

- New S-Box two diminution array 16 x 16

Begin

1.Initialization parameters by apply function

$$F(x) = F(x) = x^{16} + x^5 + x^3 + x + 1$$

Where

$X = \{X_1, X_{2,\ldots}, X_N\}$ and LBi and UBi are the lower and upper bound values for the variable xi. N is equal of 256.

2.Initialization of the Food Source Memory (FSM) The Food Source Memory (FSM) is an augmented matrix of size SN × N comprised in each row.

$$Xj(i) = LBi + (UBi - LBi) \times r \quad \ldots (1)$$

$\forall j \in (1, 2\ldots SN), \forall i \in (1, 2\ldots N)$

Where

SN equal of 256 Note that r ~ (0, 1) generates from key.

3.Assigning employed bees to the food sources using following equation :

$$X'(i) = X_j(i) + r(X_j(i) - X_K \quad \ldots (2)$$

$\forall K \in (1, 2\ldots SN)\ K \neq j$

4.Sending the onlooker bees the process of selection at the onlooker phase works as follows:

- assign for each employed bee a selection probability P_j as follow:
-

$$Pj = \frac{F(xj)}{\sum_{K=1}^{NS} F(xk)}$$

- The food source of the employed bee with the highest fitness is selected by the onlooker bee, based on its selection probability and adjusted.

5.Sending the Scout to search for possible new food sources using equation (1)

- For i=1 to SN do
- If (scout (i) = limit) then
- Generate X_j using equation (1)
- Where limit =256.
- End if
- End for

6.Memorizing the best food source

This involves memorizing the fitness and position of the best food source, x^{best} found so far in FSM.

7.Stop condition

Steps 3 to 6 are repeated until a stop criterion is met. This is originally determined by the maximum cycle number (MCN) value where MCN is equal of 2500 cycle.

8.Permutation

The permutation portion is simply the transposition of the bits or the permutation of the bit positions

End

Experiment

This procedure is performed using visual csharp .net 2010 and get the following results.

Table (1): New S-Box

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

Algorithm of INV Colony Byte

A new way proposed for computing the Inverse of S-Box and this way was tests on the original S-Box and many others Pre-defined S-Boxes and it give a correct result by comparing the results obtained from it with the Pre-defined Inverse S-Boxes, and also

by the decrypting operation. This new way satisfy the simplest, efficiency and speed in computing the Inverse of S-Box.
The following steps will explain its work:

Input:

- S-Box matrix size 16x16

Output:

- Inverse S-Box matrix size 16x16

Begin

In every S-Box and its inverse each number in the matrix of S-Box representing the address of column and row in its inverse Box and vice versa. According to this, the new way built.

1. Loop to all numbers in the S-box.

2. Extract each number from S-Box and speared it into two digits High and Low digit. I.e. the '7C' from Table (1) which found at Row = 1, Column =2 will spreading to '7' & 'C'.

3. The address of the spreader number ('7C') also will extract and will combine to represent new number that will store in the Inverse S-Box. It should be mention that the row address represent the high digit and the column address represent the low digit for new number. I.e. in the previous step the address is Row =1, Column =2. Therefore, the combined number will be '12'.

4. The combined number (i.e. '12') it should store in the Inverse S-Box at location specified by the extracted number in step 2. In addition, the high digit will represent the row address in the new inverse S-Box and the low digit will represent the column address. So that mean the combined '12' will be store in Row=7 and Column=C in the inverse S-Box as shown in Table (2).

5. The previous step will repeated to all S-Box elements individually. Until the Inverse of S-Box computed successfully.

It should be mention in this new way; ones can also compute the S-Box from its Inverse and vice versa.

End

Table (2): New InvS-Box

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

Description new S-box and its resistance to attacks

Proposed S-box it is design using complexity of its algebraic expression in GF (2^{16}), does not contain any Xored operation that is mean resistance to differential attack and does not contain linear combinations of input bits and linear combination of output bits that is mean resistance to linear attack. Larger values resulting from new S-box an attempt to make it difficult for a guess.

Use new S-Box in AES

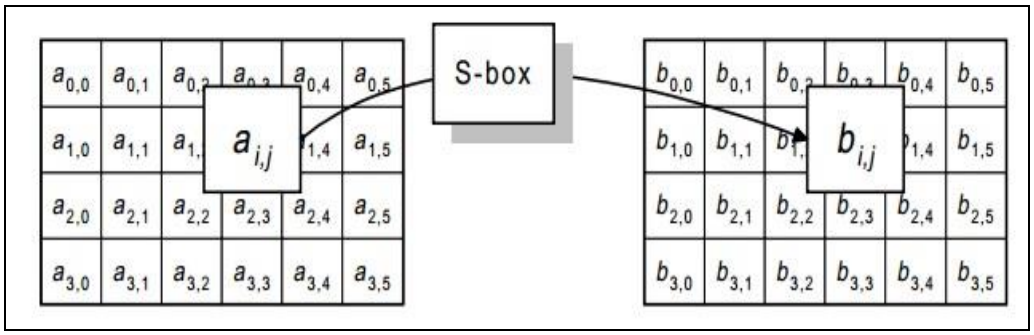
Uses proposal S-box same uses original S-box operating on each of the State bytes independently. The substitution table (or S-box) is invertible and is constructed by the composition of two transformations:

1. First, taking the multiplicative inverse in GF (2^8), with the representation defined in Section 2.1[2]. '00' is mapped onto itself.
2. Then, applying an affine (over GF (2)) transformation defined by

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

3. By using the vector from **(step 2)** as a (x, y) index for new S-Box and return new values

The application of the described S-box to all bytes of the State is denoted by:



The inverse of Byte Sub is the byte substitution where the inverse table is applied. This is obtained by the inverse of the affine mapping followed by taking the multiplicative inverse in GF (2^8), can show how to used new s-box and inverse s-box in the following example.

Plain text= In the beginning praise be to Allah Lord of the worlds.

Cipher text= ?BpY??؟<ى?'Rشه)?<ظذب??m?خh??2?7"U?pشُHي?Bط!ء"?>A

CONCLUSION

1. Proposal new s-box depend on bee colony algorithm to increase resistant linear and differential attack where new s-box doesn't depend on Xoring operation and multiplicative inverse.
2. Used bee colony algorithm because it is one of the best artificial intelligence algorithms to contain a variable taking value at random every time as well as to increase the random new values are entered 256 S-box matrices then pass the result to permutation box to transaction position of bit.

3. The randomly ABC dependent on key (**secret key enter by user 16-bits**) to addition random and dynamic to proposed s-box unlike static original s-box.
4. Proposal new inverse s-box, the method used in the inverse s-box can find inverse any matrix without depend on multiplicative inverse.

REFERENCES

[1] .From web site:
SearchSecurity.com offers security-specific information about cryptography, 2011.
[2].NIST: Specification for the ADVANCED ENCRYPTION STANDARD (AES).Technical Report FIPS PUB 197, National Institute of Standards and Technology ,2011
[3].Morioka S., "Satoh, A.: A 10 Gbps full-AES crypto design with a twisted-BDD S-box architecture", IEEE International Conference on Computer Design, 2002.
[4].Jay Ramachandran, "Designing Security Architecture Solutions", John Wiley & Sons, Inc. 2003.
[5].Eli Biham, "On Matsoi's Linear Cryptanalysis", Compute Science Department Technion, Haifa Institute Technology, Springer-veriag, 1998.
[6].Benjamin Toft Jakobsen, Mehdi Abyar and Peter Sebastian Nordholt "Linear and Differential Cryptanalysis", University of Aarhus, Denmark December 15, 2006.
[7].Alex B. and Dmitry K., "Related-Key Cryptanalysis of the Full AES-192 and AES-256" ,In ASIACRYPT'09, volume 5912 of Lecture Notes in Computer Science, pages 1–18. Springer, 2009.
[8].From web site:
http://www.hutter1.net/ai/uaibook.html.
[9].Elaine R., Kevin K., " Artificial Intelligence", Second Edition, page no.3.
[10].B. Basturk and D. Karaboga, An artificial bee colony (abc) algorithm for numeric function optimization, IEEE Swarm Intelligence Symposium 2006, Indianapolis, Indiana, USA, May 2006.
[11].From web site:
http://en.wikipedia.org/wiki/Artificial_intelligence.
[12].D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.
[13].D. Karaboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm", Applied Soft Computing, 8 (1), 687-697, 2008.