Tikrit Journal of Pure Science
ISSN: 1813 – 1662 (Print) --- E-ISSN: 2415 – 1726 (Online)
Journal Homepage: http://tjps.tu.edu.iq/index.php/j

A NOVEL SERPENT ALGORITHM IMPROVEMENT BY THE KEY SCHEDULE INCREASE SECURITY

Hassan Rahmah Zagi , Abeer Tariq Maolood
Computer Science Department , University of Technology

https://doi.org/10.25130/tjps.v25i6.320

ARTICLE INFO.

Article history:
-Received: 2 / 7 / 2020
-Accepted: 30 / 8 / 2020
-Available online: / / 2020

Keywords: Encryption, Decryption, Serpent, Hash Function, AES -Key Schedule, Complexity , Time, Statistical Tests.

Corresponding Author:
Name: Hassan Rahmah Zagi
E-mail:
hassanzajee55@gmail.com
110032@uotechnology.edu.iq
Tel:

ABSTRACT

Block encryption algorithms rely on the two most important features of their complexity and ease of use to support security requirements (confidentiality, data integrity, and non-repudiation) to prevent unauthorized users from entering the system and tampering with centralized data, disrupting it or disclosing it. The data encryption and decryption process is done using the (Serpent) algorithm, which is one of the most important of these operations. AES Algorithm Proposals. In this paper, a new proposal is presented to improve and support the confidentiality of data while adhering to the external structure of the standard algorithm, relying on designing a new approach to the key generation function because the sobriety of block cipher relies on the use of a strong and unique key. Where several functions were used (Gost external structure) with a combination of (Shift <<<), (AES -Key Schedule), (MD5)). The results of the proposed method were examined using statistical measures, yielding good results, and overcoming the weakness of the key generation function of the original algorithm, in addition to enhancing the most important cryptographic features "confusion", "diffusion" and "increased randomness".

1- Introduction

Encryption technology is an ancient technology for securely transmitting and receiving data. This technology depends on two main processes, the first of which is the encryption process: It is the process of mixing the sender's data with the agreed key, which is called the encrypted text. Second: The decryption process is the process of mixing the ciphertext with the key and called the original text[1].

In block encryption algorithms, the same algorithm structure is used in the encryption and decryption process, and the key used in both processes is chosen relying on Strong foundations to form a strong algorithm that is difficult to crack by attackers[2]. Nowadays, much has been added from mathematical technologies or artificial intelligence to support and improve algorithm construction. The encryption brings strong principles (Data confidentiality, anonymity and authentication, Integrity) to secure data from unauthorized attackers. The coding process flow is illustrated in Figure(1) [1] .

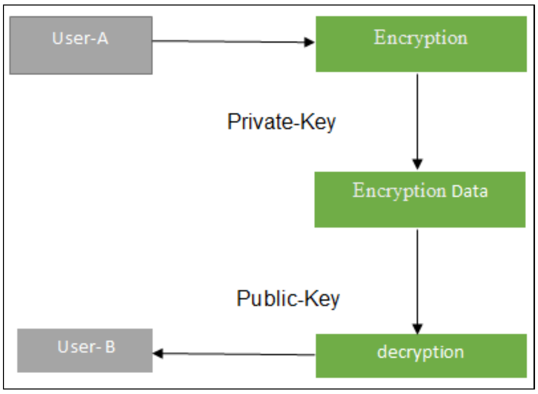


Fig. 1:Process flow of Cryptography Block Diagram [1].

One of the block cipher algorithms (Serpent) that manipulate data by switching 128 bits through a table (Initial permutation (IP)) and then divide the words 128 bits into four sections 32 bits to pass to a table (Substitution-box). After the algorithm is entered, the data processing stage begins by passing it on a function (Ip) that changes the order of the data and then enters with round positions that last to 31 shared with 32 keys that are generated by the key generation function. [3]

This paper presents a proposed improvement to the algorithm Serpent by constructing a generating function in a new way. It requires 256 entries to the function to generate 32 keys that are used within the original algorithm .

2-Related Work

Enhanced Serpent algorithm that is a Feistel network where encryption is 32 times repeated. Where the input block length of the encoder is 265 bits while the key length is up to 256 bits. This proposal carefully takes into account the advantages and strengths of the original algorithm and prevents disadvantages by providing an improved version of the standard algorithm[4].

In this paper presents an improvement of the serpent algorithm based on the Virtex XCVlOOO FPGA, using a partial assessment technique. In this implementation, partial reconfiguration is used. In comparison with other implementation, high performance is the main effect of using partial reconfiguration in the medium reduced required. The architecture is built into each cipher's interior and the results reveal a higher performance of the encryption/deception than other implementations reported recently[5].

proposed Implementation for Secured Data Transmission of SERPENT Cryptographic Algorithm. Using the programming tool for graphics. This algorithm uses a round function process that Contains(The Key Schedule, substitution(S-box), Linear Transformation(LT)). This is a substitution permutation network method. The algorithm requires little memory, and is fast and simple to upgrade[1].

In this paper, a new approach to the Serpent algorithm is proposed that is done by modifying Functions and rounds because the traditional functions in the original algorithm structure need a high execution time and a degree of complexity. Whereas, a new exoskeleton was proposed relying on several techniques (Process Multiply the columns, compensation box, and permutations (initial and final) to increase the complexity of the algorithm's work[2].

In this proposal, the enhanced Serpent algorithm it more compatible in terms of working with the algorithm(AES). by the loop terminated by (4 × 4) S box. The method uses 4 in 4 S squares, created by multiplying the class by the hopping chain. Final (containing bytes instead of bites). All operations in this function correspond to the operations of the hopping chain link. The search results clearly show the effectiveness of the improved python machine's algorithm over the regular python algorithm[3].

In this proposal, dynamic methods are presented to improve the algorithm serpent in the functions of switching and substitution and generating keys for the algorithm based on chaotic maps in order to provide security requirements for data, and the results of the proposed method were good to overcome a lot of misfortune as well as the possibility of reducing the number of rounds to achieve shortening at the time of implementation[6].

3- Theoretical Background

3.1- Hash Function

Hash function also called a (Message Digest or the fingerprint). Allows objects in the work area to be compact. The key feature of this (hash function) is that, given the element in the field, a specific field element that conflicts with x is mathematically difficult to find. It performs a process whose entry is a variable-length string and converts it to a fixed-length binary chain. Beside that. Besides, the hash function is built to make it impossible to reverse the procedure, i.e. to locate a string that is separated into a certain value (i.e. a single name)[7] . A strong hash function also makes finding the same hash value for two series difficult. Even a small alteration in an input string can cause a drastic change in the hash value. And if a bit has been reversed in the input string, at least half the bit in the hash value must be reversed. This is a data breakdown product [8] .

3.1.1-MD5 Algorithm

The MD5 is designed to operate at 32-bit machines very easily. Moreover, the MD5 no large replacement tables are required; the algorithm is very compact. The hash function (MD5) is a developer extension that is somewhat lighter the MD4) hash function for the more "protective" MD5 message in nature. The algorithm MD5 entry has different random lengths, which in turn generates a digest message 128 with a fixed-length. A Message digest entire message data used for Authentication is determined using the authentication algorithm. The message digest is usually registered with a trusted third party or otherwise encrypted. To verify a message, the digester is used by the recipient. It can also be used to encrypt the contents of a message through another algorithm via a second pass over the data. MD5 requires that the sender and recipient measure the digest of a message throughout the body [9] . One of the most important features of hash function(MD5) is that it produces a unique output in every execution, as will be illustrated in the figure2 [10].

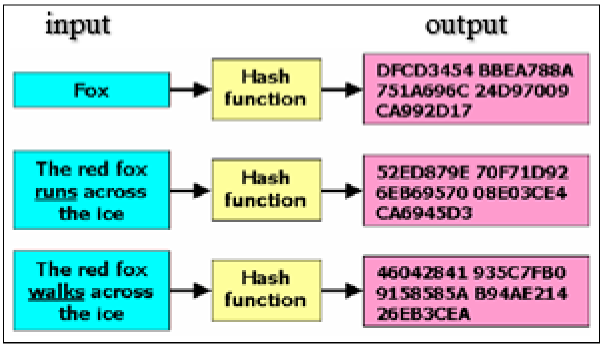

input
output
Fox
Hash function
DFCD3454 BBEA788A 751A696C 24D97009 CA992D17
The red fox runs across the ice
Hash function
52ED879E 70F71D92 6EB69570 08E03CE4 CA6945D3
The red fox walks across the ice
Hash function
46042841 935C7FB0 9158585A B94AE214 26EB3CEA

Fig. 2: hash function (MD5) Encryption[10].

Steps for working a cipher (Md5) algorithm:-

1- Insert a set of bits called (the fill bits), which in turn match the bits to be encrypted with the length of the block used by the algorithm adding the first bit (1) followed by many(0).

2- The last block is reserved to insert the length of the original message.

3- 3-Provide the algorithm(MD5) 128 bit Buffer , divided by four with bit length 32(A,B,C,D):A= 0x01234567; B = 0 x89abcdef; C = 0 xfedcba98; D = 0x76543210.

4- The process of transforming the input data (with a length of 32 bits) is the basis for the work of the algorithm by compressing a "cycle" by logical functions and through which 32 bits are generated, as will be illustrated in Figure 3. The logical functions used for each session: F(X,Y,Z) = (XY v not(X) Z) , G(X,Y,Z) = (XZ v Y not(Z) . H(X,Y,Z) = (X xor Y xor Z) ,I(X,Y,Z) =(Y xor (X v not(Z))) bit (X, Y, Z), G(X, Y), H(X, Y, Z), and I(X, Y, Z) is neutral and objective if the bits of X, Y and Z are objective and unbiased. Steps.[11].

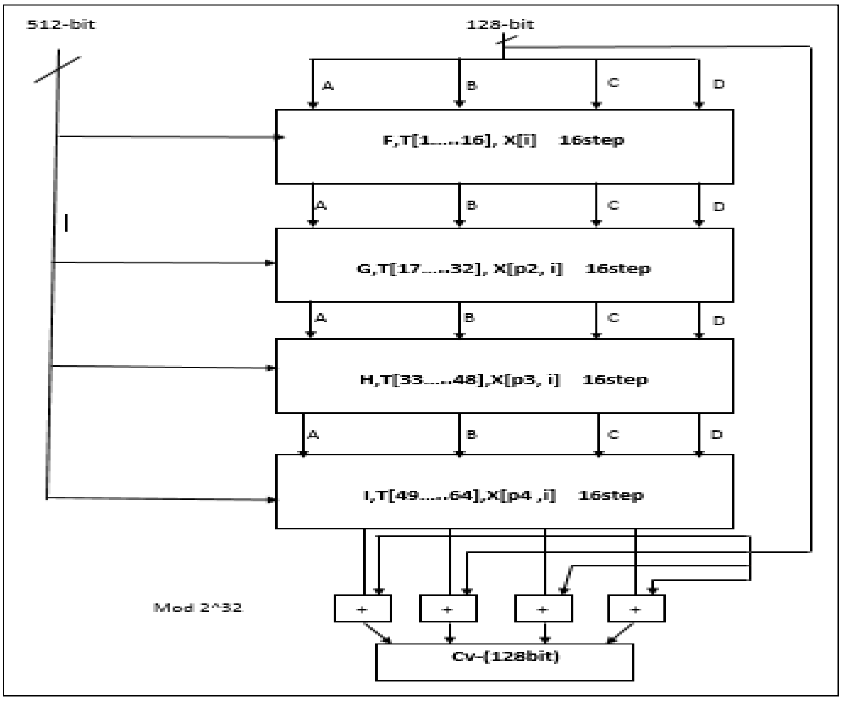

512-bit
128-bit
A
B
C
D
F,T[1.....16], X[i] 16step
G,T[17.....32], X[p2, i] 16step
H,T[33.....48],X[p3, i] 16step
I,T[49.....64],X[p4 ,i] 16step
I
Mod 2^32
+
+
+
+
Cv-(128bit)

Fig. 3 :Rounds algorithms MD5[11].

5- The output is generated at a fixed length.

3.2-Serpent Block Cipher Algorithms

In the data processing stage by the serpent algorithm, the algorithm entry is 128 bit passing through around as shown in Figure (2), before entering data in the circular processing function, 128 is passed through the (Ip) function, which mixes the data while maintaining On the statistical properties then begins the stage of circular data processing which in turn needs to use 32 keys generated by the mechanism to generate the keys. After that, the actual processing is done using many mathematical basis functions where the XOR function is used between 128 resulting from the function (Ip) and the first key that was created and then after that. The result of the previous job is divided into a length of 128 into 4 parts with a length of 32 bits that are passed to a function (alternatives) in which data is processed using tables (S-box) and then the condition of the round guest is activated by which it is determined in around. The diagram in Figure 4 shows the method for Serpent encryption, Figure 5 hows the method for Serpent decryption[1].

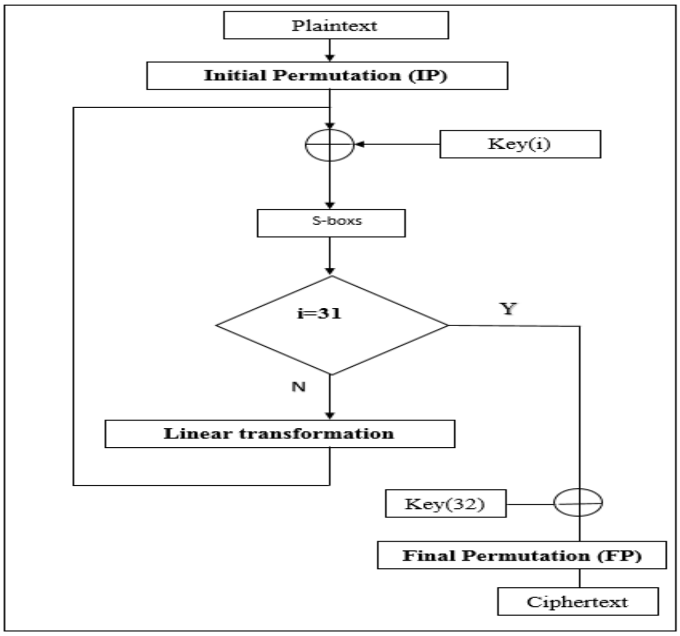


Fig. 4: Serpent encryption block diagram [1].

Figure 5: Serpent decryption block diagram [1].

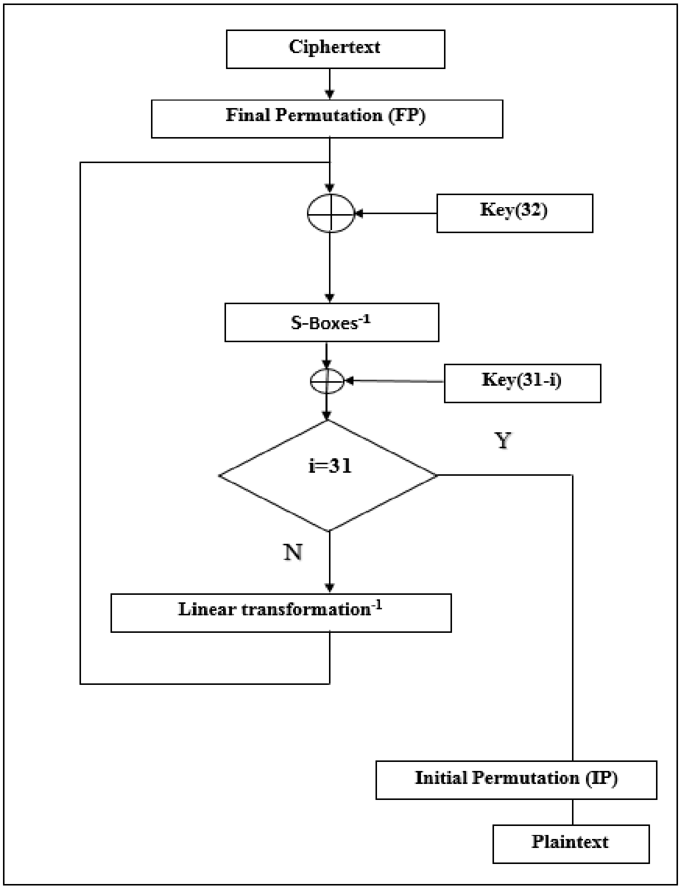


Fig. 5: Serpent decryption block diagram [1].

During the encryption process by the Serpent algorithm, data is processed by four main functions as follows[1]

1- Initial permutation (IP) & Final Permutation (FP)
2- The S-box Functions.
3- Linear transformation
4- Key Schedule

1 –The Initial Permutation (IP) Functions & The Final Permutation (FP) Functions.

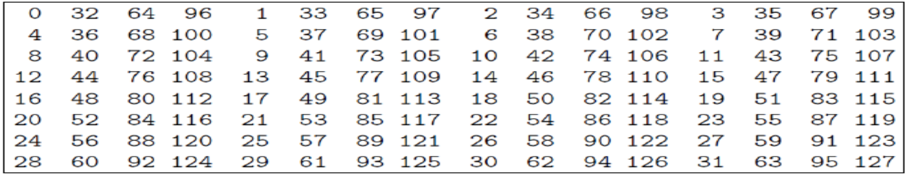
The functions (Ip) and (FP) are applied to the plain text (P) to produce (B) that enters the first round(R) of the algorithm, where the (Ip) function mixes the data before passing it to the algorithm's functions, which in turn maintains the statistical properties of the data and increases its dispersion., Thus enhancing the 'confusion' feature, the initial permutation, and final switching that are used to simplify the optimization of cryptographic implementation. Linear flipping and shifting add more complexity to the algorithm. The base permutations are computed by [(i x 32) mod 127] and the final substitution with [(i x 4) mod 127)]. As shown in the table (1) function (Ip), and table (2) function (FP)[6].. The equations also explain how it works[5] .

B_0 = IP(P)

B_{i+1} =R(B_i)

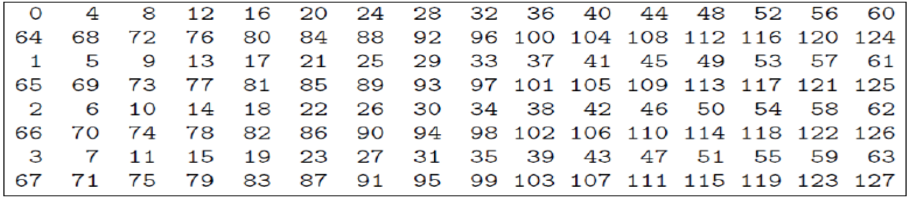
C =FP(B_{32})

Table 1: IP of Serpent algorithm [6]



0	32	64	96	1	33	65	97	2	34	66	98	3	35	67	99
4	36	68	100	5	37	69	101	6	38	70	102	7	39	71	103
8	40	72	104	9	41	73	105	10	42	74	106	11	43	75	107
12	44	76	108	13	45	77	109	14	46	78	110	15	47	79	111
16	48	80	112	17	49	81	113	18	50	82	114	19	51	83	115
20	52	84	116	21	53	85	117	22	54	86	118	23	55	87	119
24	56	88	120	25	57	89	121	26	58	90	122	27	59	91	123
28	60	92	124	29	61	93	125	30	62	94	126	31	63	95	127

Table 2: FP of Serpent algorithm [6]



0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
64	68	72	76	80	84	88	92	96	100	104	108	112	116	120	124
1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
65	69	73	77	81	85	89	93	97	101	105	109	113	117	121	125
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
66	70	74	78	82	86	90	94	98	102	106	110	114	118	122	126
3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63
67	71	75	79	83	87	91	95	99	103	107	111	115	119	123	127

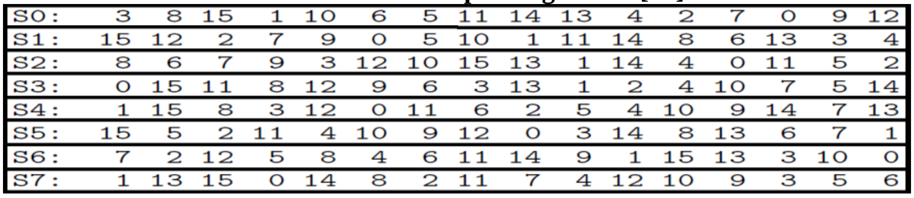
2-The S-box Function

The (S-box) Function helps raise the ambiguity of the ciphertext. The S- box focuses on uncertainty and scattering of data, which enhances its "confusion" feature. the serpent algorithm uses eight S- box containing 16 elements through which the block is encrypt. there are illustrated in Table (3) and Table(4)[1]. The pseudo-code generating function (S-box) use in serpent algorithm[12].

```
i = 0
repeat
CurrentS-box = i modulo 32
For i=0 to 15 do
j = Sbox[(currentS-box+1) modulo 32][serpent[i]];
swapentries (Sbox[currentS-box][i],Sbox[currentS-box] [j])
if Sbox[currentS-box] [.] has the desired properties, save it
i = i + 1;
```

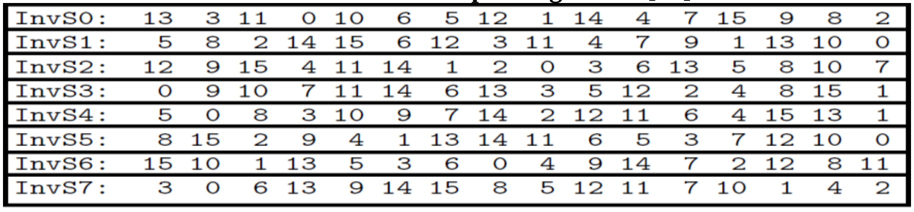
To create eight (S-box) each containing 16 bit.

Table 3: S-box of Serpent algorithm [12].



S0:	3	8	15	1	10	6	5	11	14	13	4	2	7	0	9	12
S1:	15	12	2	7	9	0	5	10	1	11	14	8	6	13	3	4
S2:	8	6	7	9	3	12	10	15	13	1	14	4	0	11	5	2
S3:	0	15	11	8	12	9	6	3	13	1	2	4	10	7	5	14
S4:	1	15	8	3	12	0	11	6	2	5	4	10	9	14	7	13
S5:	15	5	2	11	4	10	9	12	0	3	14	8	13	6	7	1
S6:	7	2	12	5	8	4	6	11	14	9	1	15	13	3	10	0
S7:	1	13	15	0	14	8	2	11	7	4	12	10	9	3	5	6

Table 4: S-box^{-1} of Serpent algorithm [12].



InvS0:	13	3	11	0	10	6	5	12	1	14	4	7	15	9	8	2
InvS1:	5	8	2	14	15	6	12	3	11	4	7	9	1	13	10	0
InvS2:	12	9	15	4	11	14	1	2	0	3	6	13	5	8	10	7
InvS3:	0	9	10	7	11	14	6	13	3	5	12	2	4	8	15	1
InvS4:	5	0	8	3	10	9	7	14	2	12	11	6	4	15	13	1
InvS5:	8	15	2	9	4	1	13	14	11	6	5	3	7	12	10	0
InvS6:	15	10	1	13	5	3	6	0	4	9	14	7	2	12	8	11
InvS7:	3	0	6	13	9	14	15	8	5	12	11	7	10	1	4	2

3-Linear Transformation(LT) Function

The Linear Transformation Function because of the possibility of a 1-bit change in the input will lead to a change in several bits at the output, which leads to a change in the statistical characteristics and the increase in randomness and complexity because it enhances the advantage of "diffusion" [13]. Linear transformation depends on a linear function through which 32 bits are mixed in each of the output words, as in the equation[5]. Figure 6 Shows the (LT) Function[6] .

X0,X1,X2,X3= Si(Bi xor Ki)

X0 = X0 <<< 13

X2 = X2 <<< 3

X1 = X1 xor X0 xor X2

X3 = X3 xor X2 xor (X0 << 3)

X1 = X1 <<< 1

X3 = X3 <<< 7

X0 = X0 xor X1 xor X3

X2 = X2 xor X3 xor (X1 << 7)
X0 = X0 <<< 5
X2 = X2 <<< 22
Bi+1 := X0;X1;X2;X3

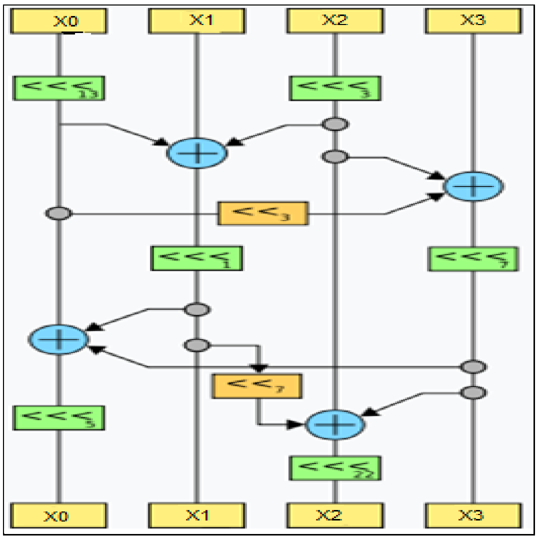
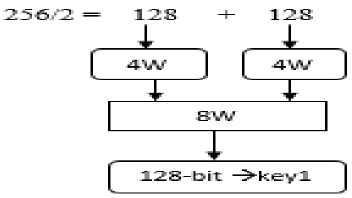


Fig. 6: Linear Transformation(LT) block diagram [6].

In the stage of data processing in a function (LT), a bit is divided into 4 parts 4 in order (X_0 , x_1 , x_2, x_3). The data is scattered by a linear approach by using the << <left rotation process, << left shift process, and the process (XOR) and through this stage a new 128 is produced that shows Figure (6) Linear Transformation (LT) block diagram [13].

4 - Key Schedule

The algorithm requires 32 keys to produce where the 256-bit key generation function is entered based on the formula:



The remaining keys are generated by the following equation:

$$W_i = (W_{i-8} \oplus W_{i-5} \oplus W_{i-3} \oplus W_{i-1} \oplus \phi \oplus$$

where ϕ is a proportion of a golden ratio, the 0 * 9e3779b9 suggesting < < < is a fractional component of that which is the left shift , generated by using the primitive polynomial (X^9+X^7+ X^5+ X^3+1) [4 ,12]. All 31 sub-keys are generated from the key generation algorithm. It is difficult to implement the key formation algorithm in a way that calculates the sub key in a parallel way with the output of the corresponding round. The key generation algorithm has been implemented as a separate unit for the circuit and the sub-key is stored in a 4096-cell memory block. It will also be illustrated by the Figure (7) [14] ..

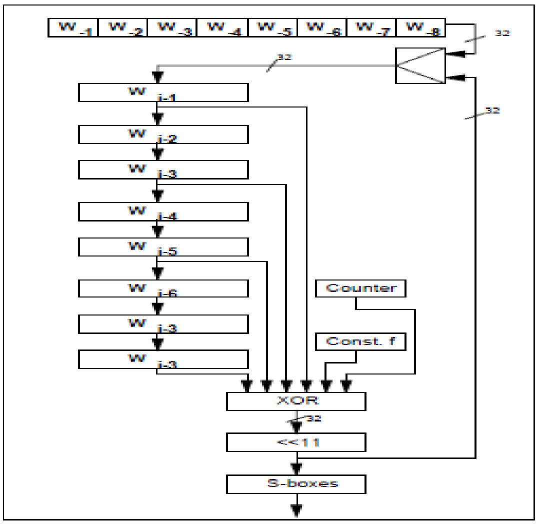


Fig. 7 :Serpent -Key Schedule [14].

3.4-AES -Key Schedule

The AES algorithm relies on the production of strong and secret keys used for encryption and decryption. Through this algorithm, an approach to generate the encryption key (Key) is adopted to produce a basic expansion sequence. The key expansion determines how the extended key from the key is extracted, as, in the process of the cipher, one round key is required for the initial key addition . It will also be illustrated by the Figure(8) [13].

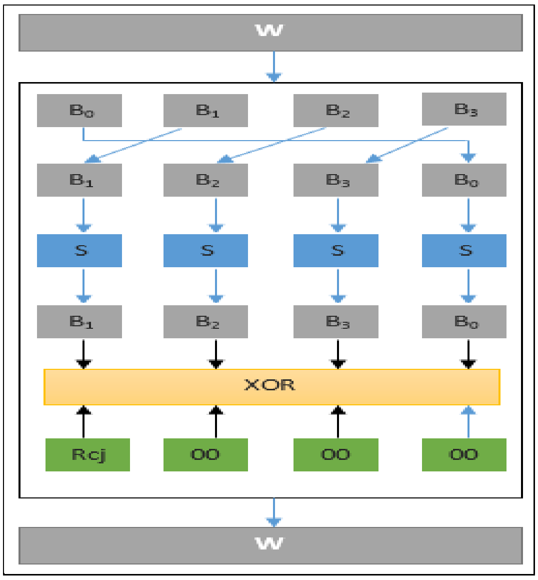


Fig. 8: AES-Key Schedule [13].

3.4.1-Function (g)

Function (g) contains 3 primary expansion subfunctions:

1- Left circular shift (RotWord): As an entry to a cyclic exchange, use [X_i, X_{i+1}, Xi_{+2}, X_{i+3}] and return [X_{i+1}, X_{i+2}, X_{i+3}, X_i] script.

2- AES S-box (SubWord): It is a method that uses four-byte words to produce output by taking a substitution (S-box) for every byte of its word.

3- Round Consistent Word Array: Rcon[i] is the list of round and consists of values supplied with [Xi-1, {00}, {00}] and Xi-1 start at 1 .The round constant for each round is different. Three of the circular constant's rightmost bytes are 0. The XOR with Rcon is the left byte of the term.. Figure (9): Function (g),[13].

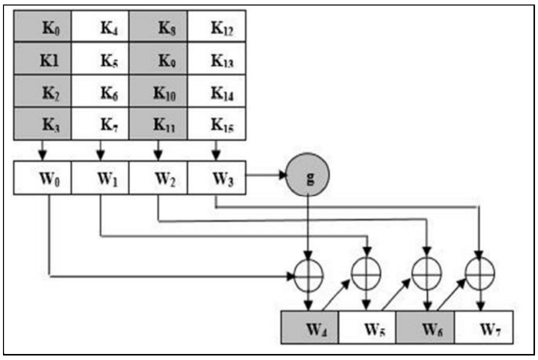


Fig. 9: key-AES Function (g)[13].

4-Proposed Modifying Serpent Algorithm

The proposal to develop the serpent algorithm aims to preserve the general structure and main layers of the standard algorithm.. Figure 11 shows the general evolution of the algorithm. The results of the modified algorithm appear more complicated and random, and enhance the (spread) and (confusion) properties of the data, as will be illustrated by the statistical tests used at a later time. The algorithm is modified by this by using a different approach in the way of generating strong keys that enter in the process of encryption and decryption, and the keys were created using several functions (General structure-Gost), (shift<<<), (MD5), (AES-Key Schedule) for the algorithm. The generated key for each session will be used within the algorithm. As shown in Figure (10).

Steps to modify the algorithm :-

1-256- bits enter for the proposed approach, which in turn is divided into two parts.

2 -128-bits from the right side are converted to a state matrix [4 x4], after which the state matrix [4x4] is passed to a function (AES -Key Schedule) that in turn produces 11 matrices to the first round, 10 matrices to the second round, and 11 matrices to the third round.

3- The array [4x4] resulting from (AES - Key Schedule) is converted into a 128-bit binary system in order to pass it to a function (MD5). The hash result will be the key that will be used in the proposed algorithm.

4- The last key is generated by each round after it is converted to a binary and entered by the hash function, so it is also done for it (XoR) with the left side

5- After completing the first round, the left side is entered into a function (Shift(<<<)), and then the switch between the right and left ends

In the algorithm for generating the proposed keys the entry is F-Function 128 -bit and then passes by 3 consecutive rounds. In each round, data processing begins with methods (AES-Key) that increase its property (confusion) and (diffusion) and it is known for its strength against attacks and after processing data with (AES- Key Schedule) enter to (MD5) also achieves two properties (confusion) and (diffusion) an increase in reliability. 32 keys were generated that are ready for use within its algorithm (serpent). Following the algorithm (1) to demonstrate the proposed modification of the key generation function.

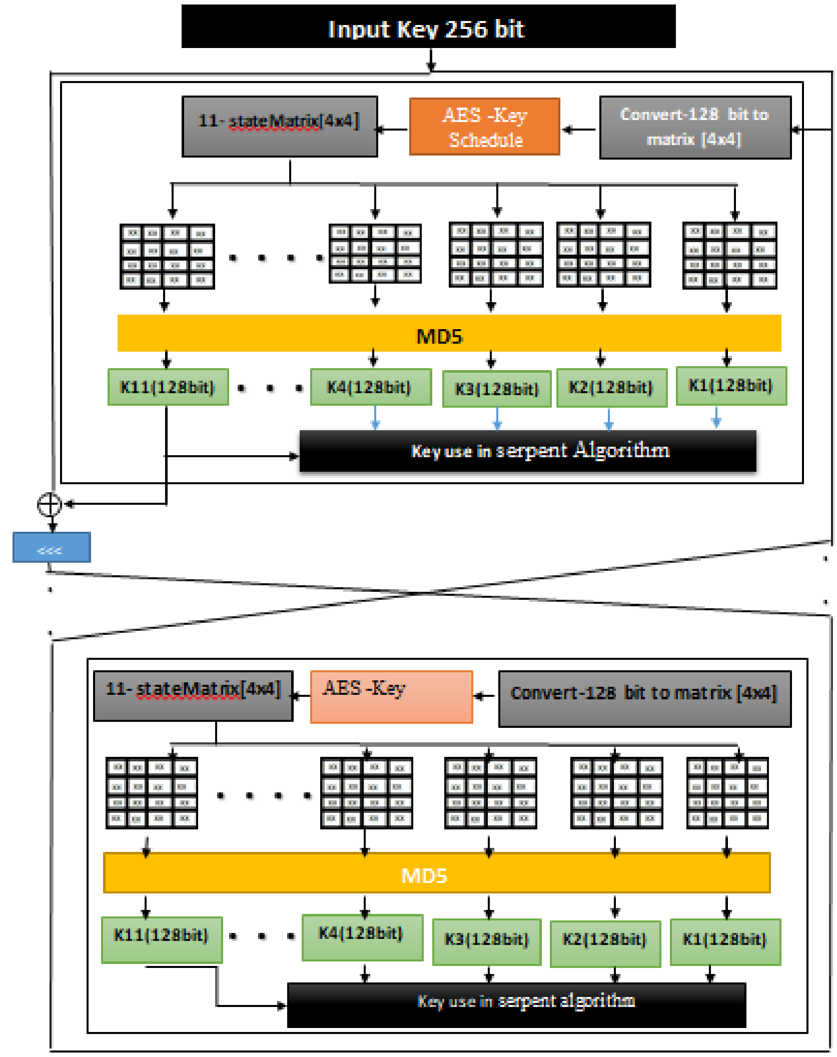


Fig. 10 : Serpent- Key Modified Block Diagram.

Algorithm (1): proposed Key Schedule Modification
Input: 256-bit plainText for the proposed key generation algorithm **Output**:32- keys Each 128-bit key is used for encryption and decryption
Begin
Step1: Split a 256-bit plaintext into the left (L_l=128 bit) and the right (R=128bit).
Step2: The right part (R=128bit) passes to the F-Function to start the processing phase according to the following steps:
1- Convert R_i -128- bits to an array[b[4x4]].
2- The array b[4x4] enter to function(AES -Key Schedule) produced 11 array [C [4x4]].
3–The array C [4x4] convert into the binary system a length of 128 bits in enters the hash function (MD5) produced key1-128-bit.
4-Through the logical function (XOR), the last key generated and the left (L_l=128 bit) key are combined as in the equation: L(i)= (key_{12}) XOR (L_i)
5- Switch operation - L(i+1) =R(i) -R(i) =shift(L(i))
End

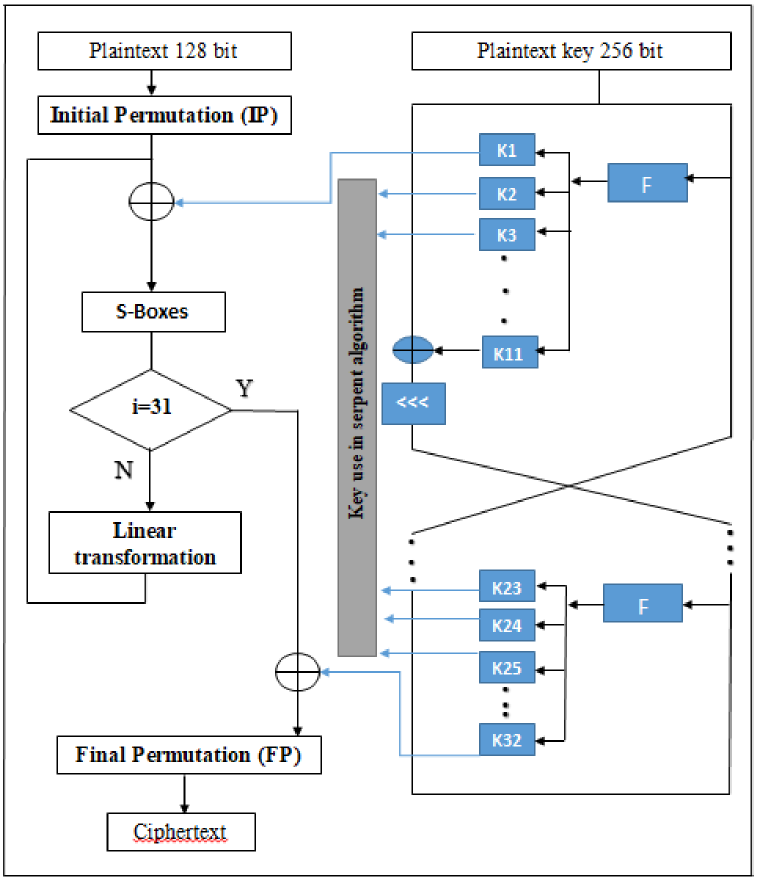


Fig.11 :proposed modified serpent encryption Modifed Block Diagram

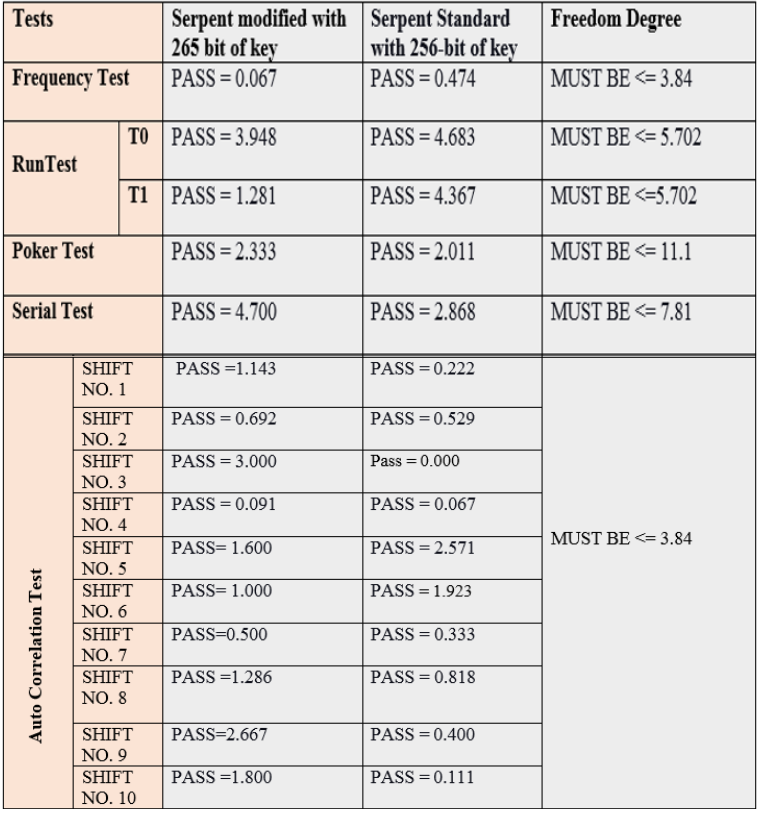
5-Tests and Experiments

The modified serpent algorithm includes 5 tests to check the output of the ciphertext (modified snake and snake algorithms) and a comparison of results of experiments and tests between the modified algorithm and the standard algorithm by relying on statistical measures (5 basic statistical tests, Nist test set, encryption run time, brute force attack and cipher attack). In this paper, the same standard serpent is used in terms of input to the algorithm And in terms of maintaining the general structure of the algorithm. Consequently, the experiments conducted show effective, more efficient, and more random results for the ciphertext, and these results will be illustrated through the table (5).

5.1 Basic Five Statistical Tests

The efficacy of the ciphertext is defined by the modified algorithm through experiments conducted on the ciphertext by 5 statistical test measures (repetition, race, poker, chain, and correlation) and through the scale test, effective and sharp scores appear compared to the original algorithm as shown in Table (5): FRQ- FRE " 0 " = 7 FRQ – FRE " 1 " = 8, FRQ-RUN$_0$ " 0 " = 4 FRQ –RUN$_1$ -OF " 1 " = 3 , FRQ –SER $_0$ " 00 " = 1 ,FRQ -SER $_1$ " 11 " = 2 FRQ – POK$_0$ " 1 " =0, FRQ– POK$_1$ " 1 " = 0,Correl-T1 = 1, respectively.

Table 5:Tests and Experiments Applied to Standard and Modified Serpent Algorithms.



Tests		Serpent modified with 265 bit of key	Serpent Standard with 256-bit of key	Freedom Degree
Frequency Test		PASS = 0.067	PASS = 0.474	MUST BE <= 3.84
RunTest	T0	PASS = 3.948	PASS = 4.683	MUST BE <= 5.702
	T1	PASS = 1.281	PASS = 4.367	MUST BE <=5.702
Poker Test		PASS = 2.333	PASS = 2.011	MUST BE <= 11.1
Serial Test		PASS = 4.700	PASS = 2.868	MUST BE <= 7.81
Auto Correlation Test	SHIFT NO. 1	PASS =1.143	PASS = 0.222	MUST BE <= 3.84
	SHIFT NO. 2	PASS = 0.692	PASS = 0.529	
	SHIFT NO. 3	PASS = 3.000	Pass = 0.000	
	SHIFT NO. 4	PASS = 0.091	PASS = 0.067	
	SHIFT NO. 5	PASS= 1.600	PASS = 2.571	
	SHIFT NO. 6	PASS= 1.000	PASS = 1.923	
	SHIFT NO. 7	PASS=0.500	PASS = 0.333	
	SHIFT NO. 8	PASS =1.286	PASS = 0.818	
	SHIFT NO. 9	PASS=2.667	PASS = 0.400	
	SHIFT NO. 10	PASS =1.800	PASS = 0.111	

6- Complexity and Time

In this section, the most important strengths and complexities will be clarified in the new proposal for the algorithm. Because block encryption algorithms depend on the key agreed upon in coding and decoding, a new approach has been built in the process of generating and producing all keys of the algorithm. The approach depends on:

1-The external structure of the algorithm Gost was used, whereby 4096 bits are produced in a complex way, with a function (shift<<<) being added to the left side before the exchange process takes place between the right and left sides.

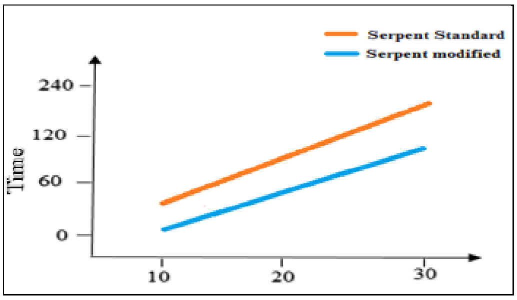
2-When the right side enters a function (AES -Key Schedule) through which the properties of "diffusion" and "confusion" are enhanced.

3-The result of the previous function is passed to a hash function MD5, which in turn creates one value for each entry. If a bit is changed, then the output will be completely different.

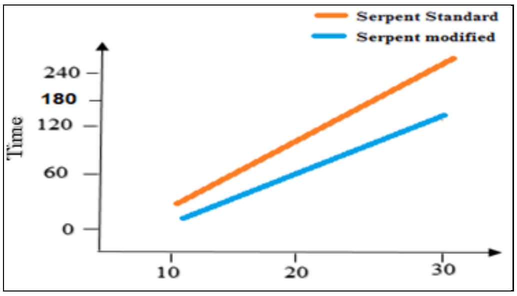
4-By this approach, if the ciphertext is exposed to the dangers of different attackers, it is necessary to reverse all the complicated operations, but it is MD5 that works in one way function to increase credibility The proposed new approach helps to add complexity time to the standard algorithm. As will be explained in Table (6).

Table 6: Time complexity comparison between modified and standard Serpent algorithm

File size	operation	Serpent Standard in Sec	Serpent modified in Sec
10kb	Encryption	12.6	9.6
	Decryption	28.8	17.7
20kb	Encryption	7.5	102
	Decryption	154.2	84
30kb	Encryption	138.6	110
	Decryption	240	175



Curve 1: Time complexity encryption serpent algorithm



Curve 2: Time complexity decryption serpent algorithm

7- Conclusions

By evaluating the experiments that were conducted on the proposal:

1- Several functions have been used to increase the complexity of key generation through the use of an extended key generator algorithm (AES -Key Schedule) the one that uses an equation GF(2^8) and to achieve the properties of "diffusion and" confusion ". In conjunction with the product(AES -Key Schedule) that enters the hash function (MD5) to achieve the generation of key unique involved in the encryption and decryption process.

2-Taking into consideration The work of the cryptanalyst to obtain the key used in encryption and decoding takes time and effort because the hash functions work in one direction.

3-The serpent algorithm is further complicated by the use of several new functions

4-The new proposal is used to encrypt all types of data

Then this paper presents several works related to amending the standard serpent. Then the modified serpent measure the (difusion) and (confusion) property of the bits, which were tested by five metrics (five basic statistical tests, the NIST test set, coding runtime, brute force attack and analytical attack) the result subkeys passed the test and achieved a full spread For bits

8- References

[1] Kabilan. K; Saketh. M. and Nagarajan. kk.(2017). Implementation of Serpent Cryptographic Algorithm For Secured Data Transmission. International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS):p.1-6.

[2] Alaa .k. F.; Semaa. H, Gada .A. and Israa. A.(2017). New Approach for Serpent Block Cipher Algorithm Based on Multi Techniques. *Iraqi Journal of Information Technology*,**7(3)**:1-13.

[3] Tariq .S.; Tanveer .u. and Ghazanfar .F.(2018). Serpent Algorithm: An improvement by 4 ×4 Sbox from finite Chain ring . *International Conference on Applied and Engineering Mathematics(ICAEM),* **2018(3):**32-37**.**

[4] Saleh, M. Al. and Ashwaq, T.H. (2005). An Improvement of Serpent algorithm.*Eng.&Tech.Journal*, **6(24):**750-765.

[5] Najafi. B.; Sadeghian, B. Zamani. M. Saheb. and Valizadeh, A.(2004). High speed implementation of serpent algorithm. 16th edn ,.ICM: 718-721pp

[6] Intisar ,A. Y.(2019). Proposed A Permutation and Substitution Methods of Serpent Block. *Ibn Al Haitham Journal for Pure and Applied Science,* **32 (2)**: 131-144.

[7] Moni . N . and Moti .Y.(1995). Universal One-Way Hash Functions and their Cryptographic Applications. *International Business Machines(IBM),* **172(1)**:1-8.

[8] William. S. (2012). Cryptography and Network Security Principles and Practice. 5th edn,. USA: Prentice Hall: 900 pp.

[9] Shweta .M.; Shikha ,M.and Nilesh .K .(2013).Hashing Algorithm: MD5. *International Journal for Scientific Research and Development,* ,**1(9)**: 2321-0613**.**

[10] Aso, A.M.(2017). Cluster forming based on spatial information using HMAC in WSN. *Tikrit Journal of Pure Science,***22(6)** *:*131-139.

[11] Ye .Ta, Kun .Z, Pu .W, Yuming .Z.and Jun .Y.(2018). Add "Salt" MD5 Algorithm's FPGA Implementation. *Procedia computer science*,**131**:255-260.

[12] Ross. A.; Eli, B.and Lars. K.(1998). Serpent A proposal for The AdvancedEncryption Standard. *National Institute of Standards and Technology (*NIST*)* AES Proposal,**174(1-2)**:1-7.

[13] Ala'a .K. and Abeer .T.(2018) Improve Block Cipher Algorithms to Protect Banking Storage Sensitive Information, *University of Technology Department of Computer Science*,pp.20-35,2019.

[14] Piotr .B. and Tomasz.C. (2000). Implementation of the Serpent algorithm using altera FPGA devices. *journal Public Comments on AES Candidate Algorithms-Round*, vol.**2**,pp.

تحسين جديد لخوارزمية السيربنت بالاعتماد على جدولة المفاتيح الرئسية لزيادة الامان

حسن رحمه زاجي ، عبير طارق مولود

قسم علوم الحاسبات ، الجامعة التكنولوجية ، بغداد ، العراق

الملخص

تعتمد خوارزميات تشفير الكتلي على أهم ميزتين هي التعقيد وسهولة الاستخدام لدعم متطلبات الأمان (السرية، وسلامة البيانات، وعدم التنصل). من اجل منع المستخدمين الغير مصرح لهم من الدخول للنظام والتلاعب بالبيانات المركزية أو تعطيلها أو الكشف عنها. تتم عملية تشفير البيانات وفك تشفيرها باستخدام خوارزمية والتي تعد من أهم مقترحات لخوارزمية.

في هذه الورقة، تم تقديم اقتراح جديد لتحسين ودعم سرية البيانات مع الالتزام بالهيكل الخارجي للخوارزمية القياسية ، بالاعتماد على تصميم نهج جديد لوظيفة توليد المفتاح.لان رصانة تشفير الكتله في التشفير وفك الشفره تعتمد على استخدام مفتاح قوي وفريد. حيث تم استخدام عدة وظائف (   ) مع دمج داله (   ) , (   ),(   ) ( ). تم فحص نتائج الطريقة المقترحة باستخدام المقاييس الإحصائية ، مما أدى إلى نتائج جيدة ، والتغلب على ضعف وظيفة توليد المفاتيح للخوارزمية الأصلية ، بالإضافة إلى تعزيز أهم ميزات التشفير مثل "الارتباك" و "الانتشار" و "زيادة العشوائية