Open Access
Baghdad Science Journal **Vol.16(1)Supplement 2019**

DOI: http://dx.doi.org/10.21123/bsj.2019.16.1(Suppl.).0270

A New Cipher Based on Feistel Structure and Chaotic Maps

Ekhlas Abbas Al-Bahrani ***Riyam N.J Kadhum****

Received 6/5/2018, Accepted 2/9/2018, Published 17/3/2019

This work is licensed under a Creative Commons Attribution 4.0 International License.

Abstract:

Chaotic systems have been proved to be useful and effective for cryptography. Through this work, a new Feistel cipher depend upon chaos systems and Feistel network structure with dynamic secret key size according to the message size have been proposed. Compared with the classical traditional ciphers like Feistel-based structure ciphers, Data Encryption Standards (DES), is the common example of Feistel-based ciphers, the process of confusion and diffusion, will contains the dynamical permutation choice boxes, dynamical substitution choice boxes, which will be generated once and hence, considered static,

While using chaotic maps, in the suggested system, called **C**haotic-based **P**roposed **F**eistel **C**ipher **S**ystem (CPFCS), we made the confusion and diffusion in dynamical behavior based on Standard and Lorenz maps. The first is used for substitution, and the second one for permutation operations .A proposed cryptographic system uses the same work (the same way) for both enciphering and deciphering. The proposed cipher operates on more than 500 bytes (4000-bit) readable text blocks by six round computing. Within the basic operator of the cipher, i.e., in the function of the round F, a dynamical lookup table 2D standard map system is used to enhance the complexity and diffusion of the unreadable text. Also, a 3D Logistic map used for key sequence generator and chaos based dynamical Initial Permutation (dynamical IP) are used to increase the diffusion and confusion. Three different image sizes and three different text length were implemented in CPFCS. The results of the proposed system and security tests improve the applicability of PFCS in the data protection and security.

Key words: Block Cipher, Chaotic Maps, Feistel Cipher, Logistic map, Standard maps.

Introduction:

Previously, In the past years, chaos theory has big attention for many papers researchers and been widely used in the fields of coding and security of data and secure connecting benefit from its great properties such as periodicity, unability to predict, and sensitivity to the seed and parameters which met some conditions such as diffusion and confusion in the sensibility of coding. As a chaotic system characterizes excellent properties of diffusion and confusion, it is widely used to design various cryptographic schemes, for example, the cryptographic systems based on chaotic maps (1), and chaotic s-box function (2).

Chaotic maps are widely applied in secure communication which is carefully studied in (3, 4). We could not only use chaotic signals to encrypt the information needed to be secure but also decipher encrypted one as well according to (5). Feistel Cipher is a pattern or sample for design block cipher, which does not mean that the feistel cipher is a type of block ciphers, for example DES algorithm is derived from feistel structure.

Compared with the classical traditional ciphers like Feistel-based structure ciphers, many problems becomes clear in that structure; weakness point of substitution boxes, weakness of permutation choices boxes, weakness in the key generation algorithm, two chosen input to an S-box can create the same output and breaking Feistel ciphers has become easy with the help of brute force attack which was impossible during that time.

There are few references to the problem of designing Feistel encryption algorithms using chaotic maps: V.Patidar and K. K. Sud, 2010, '***Block cipher using 1D and 2D chaotic maps***', This work, block cipher but with a new chaotic has been presented in henon maps, logistic maps, and secretive keys are be utilized. In this suggestion cipher based chaos, diffusion and confusion are completed consequently beside the CBC and secretive keys depend upon permute choice box (6). Wenyang Liu, Xiaomin.W and Wenfang. Z, 2014, "***A lightweight Block cipher based on chaotic maps***", through that work, a lightweight block cipher applied in the resource-limited environment

Department of Computer, College of Education, Al_Mustansiriyah University, Baghdad, Iraq
*Corresponding author: qweenr80@gmail.com

is proposed by combining general Feistel network with chaotic maps (7). J. De D. Nkapkop, Joseph. Y. Effa, Jean S. A. E. Fouda, Mohamadou. A, Laurent. B and Monica. B,2014, in this paper, the proposed work is a one round chaos-based image encryption scheme based on the fast generation of big permute use a logistics maps to change the pixel positions and at diffusion stage, permutated image is divided into *n* sub of original image and the concatenation of solving of LDE (linear Diophantine equation)and PWLCM (Piecewise Linear Chaotic Map) have been generated (8). Xiao.J, Zha W., Yang. L, M. Zhang and Lianjie .X, 2015, through this work proposes a block ciphers for WSN , depend upon compound chaotic map. That mean the applicability of chaotic maps work with networks security, the algorithm improves Feistel network and build a cubic chaotic map function and its keys spaces was large, generated by the compound chaotic sequence (9).

Generally, compared with reference (6) the proposed cryptosystem used standards chaotic maps not suggested chaotic maps, and compared with the reference (8) the proposed cryptosystem used 6 rounds implemented on two study cases; texts and images in addition to structure of the proposed cryptosystem based on Feistel structure.

In the proposed cryptosystems, new dynamic substitution and permutation methods is proposed to solve all problems. The dynamical lookup substitution table is created for dynamical substitution. The dynamical permutation method is proposed to solve the problem of static permutation choices boxes. Dynamic key generation random algorithm will be designed based on randomization of chaotic maps. In addition, the proposed cryptosystem is designed to resist the brute force-attack.

This paper is classified into seven parts. In part 2, a simple presentation to some types of chose maps are used in proposed system in **"Chaotic Maps".** Explaining of properties of Feistel structure in part 3, in **"Feistel Cipher".** While part 4, a new Feistel cipher depend upon the Feistel structure and chaotic maps proposed in **"Proposed Cryptosystem"**, and the experimental results has been presented in **"Experimental Results for texts"**. And Section 6, the results of images in **"Experimental Results for images"** while the last part was talking and discuss the result which it stated in part 6, in Section 7.

Chaotic Maps

Standard Map

The standards maps are described in (8), and can be processed by the following form:

$$a_{i+1} = (a_i + b_i)\ mod\ 2\pi$$
$$b_{i+1} = (b_i + k * \sin(a_i + b_i)\ mod\ 2\pi \quad \ldots (1)$$

where k is the one of the parameters belong to the standard map chaotic met the k > 0, and the i_{th} situations a_i and b_i both take real numbers in [0,2*p) for every i. The standards maps was discretized in a straight forward way by substitute x = a*N/2*p, y = b*N/2*p, K = k*N/2*p to Eqe. 1, which map from [0, 2*p) · [0, 2*p) to N · N. and then discretize, the map becomes**:**

$$xi + 1 = (xi + yi)\ \ mod\ N$$
$$yi + 1 = \left(yi + k * sin\frac{(xi+1)^N}{2\pi}\right)\ mod\ N \quad \ldots (2)$$

where K is a integer positive number. The characteristics of this discretize maps may not be as perfect as the first one, but it can be applied in the integer direction and range, which minimize the calculation complication and fits more for online data enciphering, the characteristics of this discretize standard maps is analyzed, firstly, then enhancing by introducing many aims, and at last used in information enciphering (8).

Logistics Maps

Logistics maps are the simple type of uncontinuous chaotic dynamic systems, utilized as chaotic work. The Logistics maps are a polynomial mapping of degree 2 that exhibits chaotic behavior (10). The logistics maps are simplest chaotic methods and provided by an Equation $Xn + 1 = \lambda Xn\ (1 - Xn)$ .For 0<Xn<1 and λ=4 the equation exhibit the chaotic behavior. Hongjuan. L. et. al suggested the 2D logistics maps processed by the processing form:

$$Xi + 1 = \mu Xi(1 - Xi) + Y1Yi^2$$
$$Yi + 1 = \mu 2Yi\ (1 - Yi) + Y2(Xi2 + Xi\ Yi \quad \ldots (\mathbf{3})$$

The above formulas increase the quadratic coupling of the items$y^2\ i\ x^2 i,\ yi\ xi$and give the high protection and security to the cryptosystems. When 2.75 < μ1 < 3.4, 2.7< μ2 < 3.45, 0.15 < y1 < 0.21, and 0.13 <y2 <0.15, the cryptographic system passes to chaotic state and can produce a chaos sequences in the region (0, 1]. Through this work, we are extension the idea of the 2D Logistics maps to three dimensions (3d) by use the following forms:

$$Xi + 1 = \lambda Xi(1 - Xi) + \beta Y1^2 Xi + \alpha Zi^3,$$
$$Yi + 1 = \lambda Yi(1 - Yi) + \beta Z1^2 Yi + \alpha Xi^3, \ldots (\mathbf{4})$$
$$Zi + 1 = \lambda Zi(1 - Zi) + \beta X1^2 Zi + \alpha Yi^2.$$

Also, the equation (4) utilize the chaotic properties for 3.53< λ< 3.81, and 0< β< 0.022,and 0 < α < 0.015 and can be taken between the range: [0, 1], (10).

Gauss Iterated Map:

There are multiple names for this map, called by (**Gauss map) and** (named as as **Gaussian map**s or **mouse map**s), is a nonlinear iterated map of the real to a real range given by the following form:

$$Xn + 1 = exp(-a * x^2 n) + \beta \quad \ldots (\mathbf{5})$$

when α and β are reals parameter. Named .Johann C.F. Gauss, the function maps the bell shaped gaussian functions similar to the logistics maps. Bifurcation of the Gauss map with $\boldsymbol{\alpha}$ =4.90 and $\boldsymbol{\beta}$ values in range [−1 , +1]. (11).

Lorenz Map

In 1963, Lorenz *et al*. found the famous Lorenz chaotic system (12), The Lorenz equations were originally derived by Saltzman (1962) as a 'minimalist' modelof thermal convection of 4 D Lorenz map in a box:

$$nx = a * (y - x) - e * w;$$
$$ny = x * z - h * y; \quad \ldots \quad (\mathbf{6})$$
$$nz = b - x * y - c * z;$$
$$nw = k * y - d * y;$$

Where *a, b, c, d, e, k* and *h* is a control parameters, and *x ,y ,z* and *w* consider the seed which have been fed to the Eq.6 to generate new values can be used as seed for second iteration of chaotic sequences

Feistel Cipher

Feistel Cipher is a pattern designed for block ciphers designing, that means the feistel cipher is not a specific scheme of block cipher. It is a design fashion which many different block ciphers depending on this pattern can be derived. DES is one of the known examples of a feistel cipher. A cryptosystem system depends upon feistel cipher network uses the same process and operations for both enciphering and deciphering (13).

Encryption Stage

The enciphering stage, which can be use the feistel model design consists of more than one rounds of process for the readable text, the single round consisting of a substitution stage and then by a permutat step. Feistel network is presented in Figure 1. In the input block to single round is split into two parts (also known as halves), that can be referred to as L_i and R_i for the left and right halves. For single round, the right part of the input block, R_i, passes by unexchanged. While the left part, L, passes by an processing that based upon R_i and the enciphering keys. First, an enciphering function f_i that takes two inputs - the key *Ki* and *Ri*. The round Function *fi* generate the output $f(R_i)$, *Ki*. Then the output of the set of mathematical operations function is Xored with *Li*. In right applying of the feistel cipher, like DES, instead of using the entire enciphering keys at single round, a round-dependent key *asubkey* is taken from the enciphering keys. This indicates that in the single round uses a different keys, although all these subkeys are derived from basic key. The permutate stage at the end of single round exchanges the altered Li and unaltered Ri. So, the Li for the next round would be Ri of the current round. And Ri for the next round be the output Li of the current round. Above 'substitution' and 'permutation stages form a single round, the number of rounds is determined by the algorithm pattern. For one time the final round is completed and the two sub blocks, 'Ri and Li are combination in this order to form the unreadable text block. The complex half of designing a feistel cipher is designing of round function *'fi'*. In order to be un breakable system, this function needs to have several important (13). The process of deciphering stage in feistel cipher is the same work (13). Figure 1, shows the Feistel Network Structure**.**

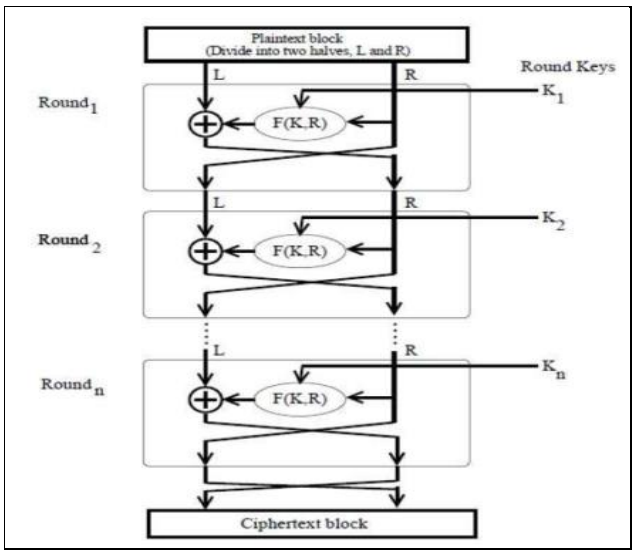


Figure 1. Feistel Network Structure

Design of Chaotic-based Proposed Feistel Cipher System (CPFCS).

In the propose cryptosystem algorithm secretive key with length of (4000 bit), (500 byte) according to the plaintext block size and based upon the construction of secretive key has been designed. 3 D logistic map is used for key generation, while 4 D Lorenz for permutation, 2D standard map for substitution operation, Gauss iterated map for enciphering/ deciphering operations and the controls parameter values for all chaos methods are kept without any change in the chaotic domain throughout the algorithm. During enciphering/ deciphering stages, the dynamic of the 3D logistic map will be mixed with the dynamic of the 4D Lorenz map. The domain produced by the logistics

maps, based upon the secretive keys used in the propose algorithm, is passed to the 4D Lorenz maps and 2D standard maps. The 4D Lorenz map is used for dynamic permutation. In the Substitution process, 2D standard map is used in the round function *fi*, to generate the right part of the ciphertext. Based upon the secretive keys are used in the algorithm. In addition to the number of rounds in the proposed system equal to six. Figure 2 presents the single round of the **P**roposed **F**eistel **C**ipher **S**ystem (PFCS).

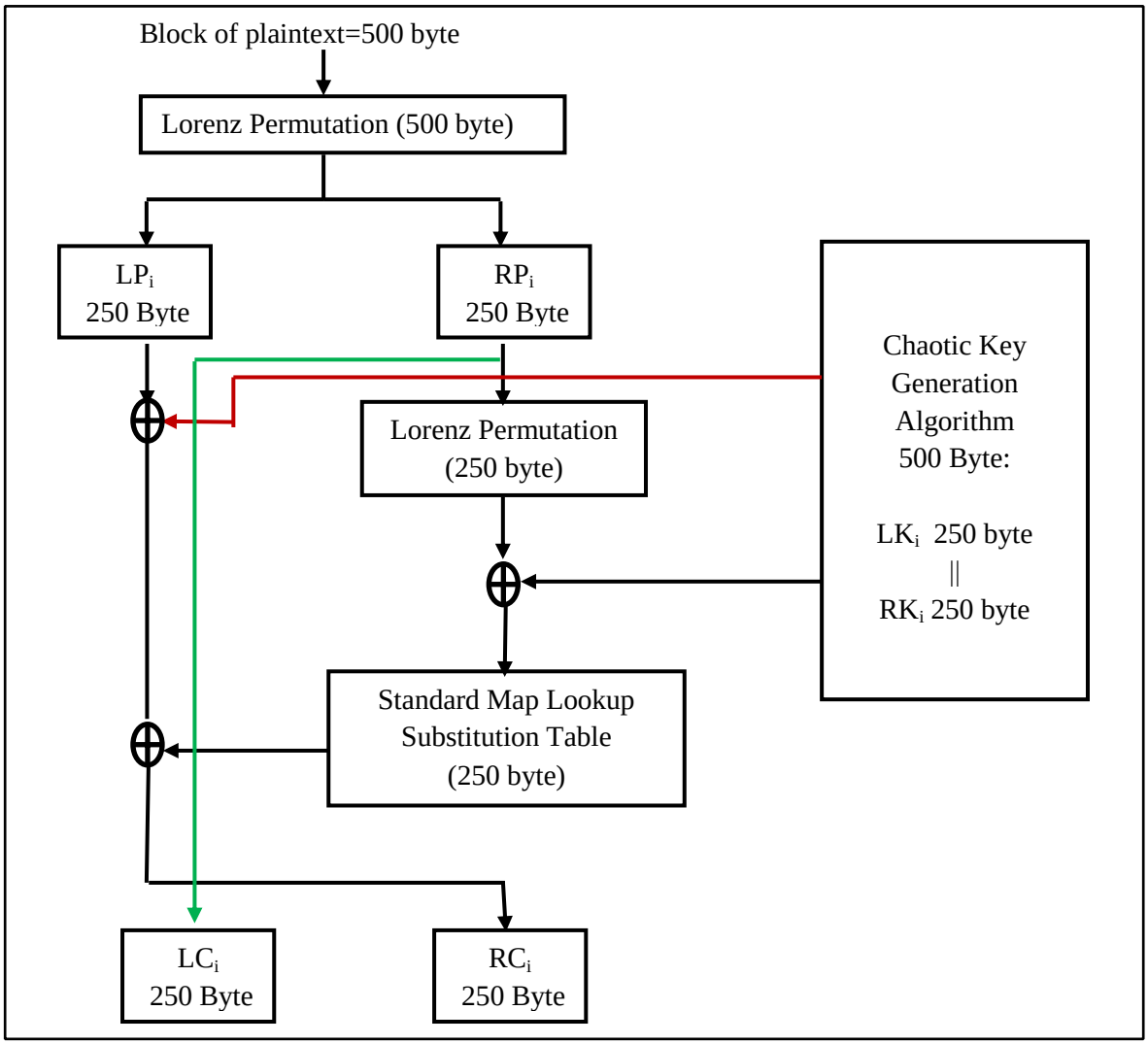


Figure 2. One Round of CPFCS

Elements and Parameters of CPFCS Design:

- **Block size.**

Each plaintext/ciphertext block equal to 500 byte (4000 bits).

- **Key size.**

A secret key has been used of 500 byte (4000 bits) size, with the same size of plaintext. This key generated by 3 D Logistic map as following steps:

Step1) the initial x_0, y_0 and z_0 for Eq.5, are determined.

Step2) the parameters of 3d logistic map a, b and u, are determined.

Step3) apply the Eq.5 to generate new values and save it respectively in, N_X, N_Y and N_Z.

Step4) convert the values N_X, N_Y and N_Z from floating numbers to binary sequences of 16 bits, and save it respectively N_{X2}, N_{Y2} and N_{Z2} by repeating the multiplication by 2 until getting of binary sequences of 16 bits.

Step5) the binary sequences N_{X2}, N_{Y2} and N_{Z2} will be process as following:

X_1=*xor* (N_{X2}, N_{Y2});
Y_1=*xor* (N_{Y2}, N_{Z2});
Z_1=*xor* (N_{X2}, N_{Z2});

Step6) Applying Xor with each four adjacent bits of the four values; X_1, Y_1 and Z_1, which are consists of 16 bits post-processed as following:

AL(1,:) = xor(X_1(1:4),X_1(5:8));
AH(1,:) = xor(X_1(9:12),X_1(13:16));
BL(2,:) = xor(Y_1(1:4),Y_1(5:8));
BH(2,:) = xor(Y_1(9:12),Y_1(13:16));
CL(3,:) = xor(Z_1(1:4),Z_1(5:8));

CH(3,:) = xor(Z_1(9:12),Z_1(13:16));

Step7) Concatenation the binary sequence between *AL* and *AH* to generate the first byte of the key, the same work to the *BL*, *BH* and *CL*, *CH* to generate the second and third bytes of the key.

Step8) The outputs of single iteration of the applying 3 dimensions Logistic map will be three bytes, these bytes will represent part of the key , that's mean the loop will be continuous until all the required key block (500 bytes) are generated.

Step9) When the key block is generated, the initial values of Logistic map are updated by Xoring the last values of N_X, N_Y and N_Z with the initial values to generate the second key block and so on for all the remaining key blocks.

Lorenz Permutation Step.

The Lorenz permutation is the first step in the proposed algorithm where the plaintext block is permuted using 4D Lorenz map in the following way:

a) The initial values x_0, y_0, z_0 *and* w_0 are determined and then used it in the Eq.6.

b) The parameters of Lorenz chaotic maps; *a,b,c,d,e,k and h* are determined.

c) Apply the Eq.6 to generate new three float numbers (N_X, N_Y, N_Z and N_w) which are converted to three integer numbers in the rang [1..500] (where 500 represent the length of block) for example in X case the equation will be as following: the same work for N_Y, N_Z and N_W.

$$X=mod\ (round\ (N_X))$$

d) The iteration of the Eq.6 will be continuous until all 500 new positions are generated randomly and saved in **L**orenz **P**ermutation **A**rray (LPA). These 500 new positions in LPA are used to generate the Permutated array by taking each two adjacent positions in LPA and the corresponding value in the plaintext block and swap the contents of these values and such on the remaining bytes are generated. For example let A represent the LPA of 10 positions array and B represent the plaintext array, C will represent the output array (Permutated array), as the following:

-A= LPA array

A= [**2** **10** 1 9 7 5 3 8 4 6];

-B=Plaintext array

B= [112 **76** 45 98 222 111 24 76 99 **27**];

-C=Permutated array

C= [99 **27** 76 111 24 98 222 45 112 **76**];

Number of Rounds.

Six rounds are be used for encryption and decryption process.

Number of halves.

In each round, each plaintext/ciphertext block is divided into two halves: the right half *RPi* and the left half *LPi*. Each half consists of 250 byte.

Round Function *fi*

The Round Function *fi* is applied to the right half yielding *fi*(Ri_i1), Round Function contain many important operations as follow:

1- **Permutation Step:** the Right half (*RPi*) of plaintext block of size 250 byte is permutated using the same algorithm of **Lorenz Permutation**.

2- **First Xor operation step**: xoring the Right key (*RKi*) and the output of Permutation Step.

3- **Lookup Substitution Table:** lookup substitution table is a 16 × 16 hex table of that is generated using 2D standard map (Eq.2) in the following steps:

a. Initial condition r1 ,r2 and k and parameter h are determined in the Eq.2

b. Apply the 2D standard map to generate new i and, new j as the result of single iteration.

c. These two output numbers are converted from decimal to hexadecimal.

d. Return to 1 until all the hex numbers of lookup substitution table are generated.

Table 1 shows an example of lookup substitution table that is generated using the initial conditions:

Table 1. Example of 16*16 lookup substitution tables generated by 2D standard map

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

Each byte of permutated array is substituted with a byte in lookup substitution table in the following way: the byte of permutated array is represented as binary sequence. This binary sequence is split into two parts, first four bits will represent the row in substitution array and the second four bits will represent the column in substitution array. This operation is repeated until all bytes of permutated array substituted in substituted array. For example, if the first byte in permutated array is 201 and it is represented as 11001001 in binary sequence. This sequence is split into row=1100 (C in hexadecimal) and column=1001(9 in hexadecimal). The intersection between row C and column 9 in the lookup substitution table shown in Table1 is 32H (50 D). So the byte 201 will be substituted by 32H (50 D).

4- **Second Xor operation step:** Xor operation between Left half (*LKi*) of key and the result is xored with the substituted array resulted from step3.

CPFCS Algorithm Steps:-

The proposed cryptographic system uses the same algorithm for both encryption and decryption. The only difference is the order of the sub keys. The encryption algorithm steps are as following:-

1- Select the plaintext file (convert it to the ASCII array) or bit map image (convert it to three arrays based on color component: red array, green array and blue array).

2- Each array is divided into blocks of 500 byte and each block will process in the following way :-

- **Lorenz Permutation Step:** in this step, the plaintext block is permutated by using Lorenz Permutation operation that is based on 4D Lorenz chaotic map and result the permutated array.
- **Split the permutated array into two halves** *(LP_i and RP_i).* Each half consists of 250 byte (2000 bit).
- **The round function *fi* will be applied on the right halve *fi(RPi).*** The round function, as discussed in previous section, consists from four main operations: Permutation Step, First Xor operation step, Lookup Substitution Table and Second Xor operation step.
- **The left half (*LPi*) of plaintext block is Xored with the left key (*LKi).***
- **The resulted array from previous step is Xored with the result of the round function *fi.*** The resulted array will be represents the left part of the ciphertext, while the right part, same part of the RP_i before enters the round function.

Experimental results For Texts

Statistical analysis

An ideal cipher must be able to protect its secret data against all types of strong attacks such as crystallize, statistical and Brute_Force Attacks. In the following tests, the results of the statistical tests after implemented upon the PFCS have been presented to measure its security characteristics confirm that the propose by using chaotic maps cipher is secure against the many known attack.

Correlation analysis

Correlation test is useful to know how to specify the strength of the linear relationship between two sequences. i.e, how strongly are these two sequences correlated or not? The correlation coefficient (*CR*) between the readable text sequence $p_1, p_2, p_3 \ldots p_N$ and its corresponding un sequence $c_1, c_2, c_3 \ldots c_N$ is computed as the following form:

$$CR = \frac{N\sum(P_jC_j) - (\sum P_j)(\sum C_j)}{\sqrt{(N\sum {P_j}^2 - (\sum P_j)^2)\,(N\sum {C_j}^2 - \sqrt{\sum {C_j}^2}}} \quad \ldots (7)$$

where *N* is length of letters in readable text/ unreadable text. We have computed the correlation coefficient between the readable text and its corresponding unreadable text for numerous sets with a different of keys. The correlation coefficients between different readable text sizes 1325,4628 and 11133 symbols (bytes) and its corresponding ciphertext for four different secretive key and different number of rounds is shown in the Table 2.

Table 2. Correlation values for different Text lengths

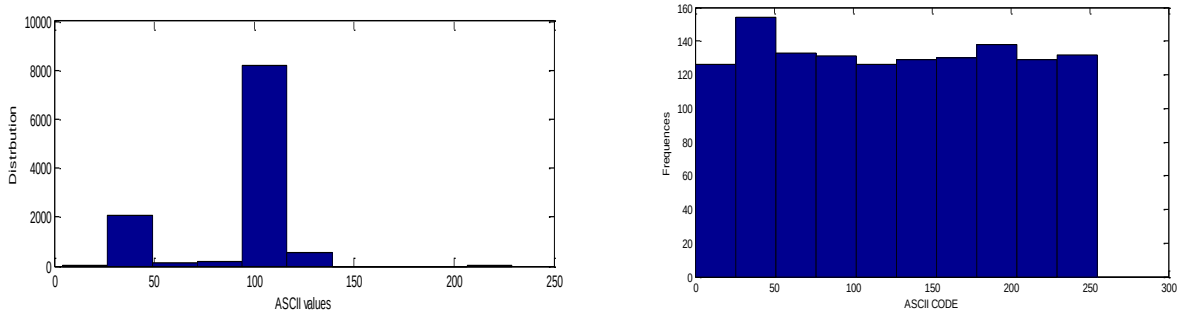
Text Length (bytes)	No. Rounds	Correlation Value
11133	2	0.0912
11133	4	0.0221
11133	6	0.0111

The best value of the *CR* falls between μ – 2σ and μ + 2σ, i.e., (–0.024, 0.024) [the range of perfect correlation coefficient values (Knuth, 1997) for good cryptosystem method when the result of correlation test as presented in the Table 1,2 and 3, are small near to 0 ($CR \approx 0$), it indicates that the readable text sequences and its corresponding unreadable text sequences are completely independent of each other.

$$\mu = \frac{-1}{N-1} \quad \text{and} \quad \sigma = \frac{1}{N-1}\sqrt{\frac{N(N-3}{N+1}} \quad \ldots (8)$$

Histogram test results

For a perfect cryptographic application, no model must be show in the unreadable text as well as the unreadable text letters must be distributed in equal way in the total interval of the ASCII codes. the cipher cryptanalyze use the histograms gauges to look distribution of letters or symbols of cipher-text and plain-text . Different sizes of plaintext sequences have been used having 1325,4628 and 11133 bytes .In Fig 3(a ,b) we have presented the ASCII codes frequencies of 11133 symbols. Fig. 3(b) shows the ASCII codes distribution of the cipher-text.



(a) (b)

Figure 3. (a): Histogram of plaintext for Text length=11133 byte
(b): Histogram of ciphertext for Text length=11133 byte

Fig. 3(a) above shows the high peaks start from 90 to 125 ASCII values and this describes the height of the plaintext letters, and low peaks as represents the low plaintext letters, while in the ciphertext we can notice the uniform distribution of the letters as shown in figure 3(b).

Information Entropy Analysis:

Il-legibility and in-determinateness are the basic aims of data enciphering. This in-determinateness can be presented through one of the most commonly used theoretical measure - information entropy. Information entropy states the degree of un-certainties in the cryptosystem and process as following form.

$$H(m) = -\sum_{i=0}^{2^N-1} P(m_i)\, log_2\,[P(m_i] \quad \ldots(9)$$

Where P_{mi} is the emergence probability of *mi*. If all single letter have an equal probability, i.e $m=\{m_0,m_1,m_2,\ldots m_{28-1}\}$ and $P(mi)=1/28(i=0,1,\ldots 255)$, and the entropy **is *H*(*m*)=8** which is equal to an ideal state. Furthermore, contrast to those algorithms in (14) the proposed algorithm is much closer to the ideal situation and more robust against entropy attack. Practically, the information entropy of enciphered images are less compared to the ideal state. To build a perfect encryption system, the entropy of enciphered data (text or image) near to the optimal state is possible. As shown in Table 2 below:

Table 2. Entropy values for different No.Rounds size

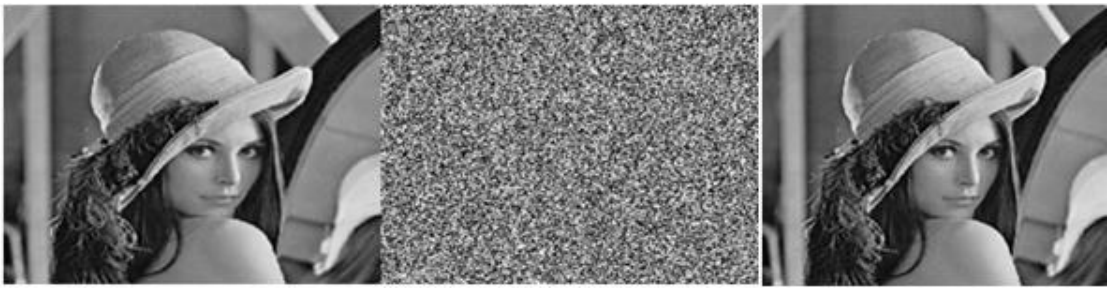
Text Length (bytes)	No.Rounds	Entropy value
11133	2	7.5677
11133	4	7.9344
11133	6	7.9991

Key space

The total keys are being used in the cryptographic application, which it's constructed of many parts: 8 for permutation of the entire system, 3 for key generator, 3 for substitution process. These 14 parts are different from each other. Therefore, for *m* iterations, the size of the space of the total enciphering operations in the PFCS, **N=662,** If N ≥ 128, and the whole lengths satisfies H(N) > 2^128. That's mean there is no contradiction among these different keys, which keeps the cryptographic application in high security and protected from all known attacks.

Experimental Results for Images

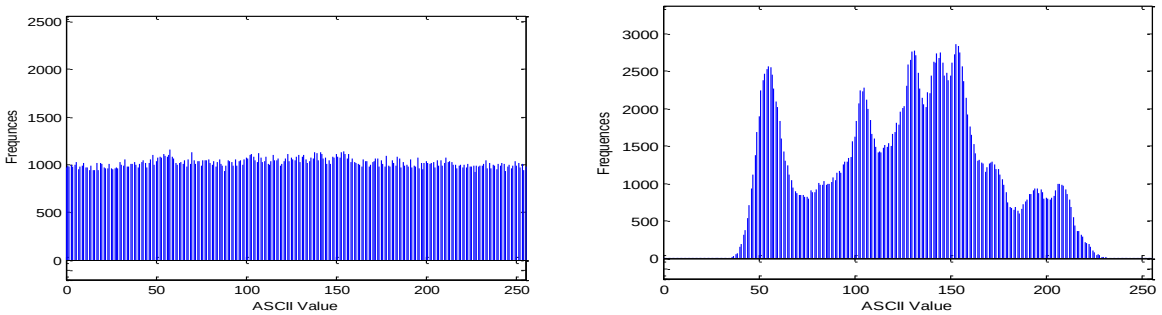
In the proposed system, many types of images has vebeen experimented; binary, gray scale and color images for example the first image was lena image, has size 512*512 pixels as shown in Fig 4(a) stated the original image, Figure 4(b) stated the encrypted image using the proposed system for six rounds, while in Fig. 4(c) shows the decrypted image by using the proposed system.



(a) (b) (c)

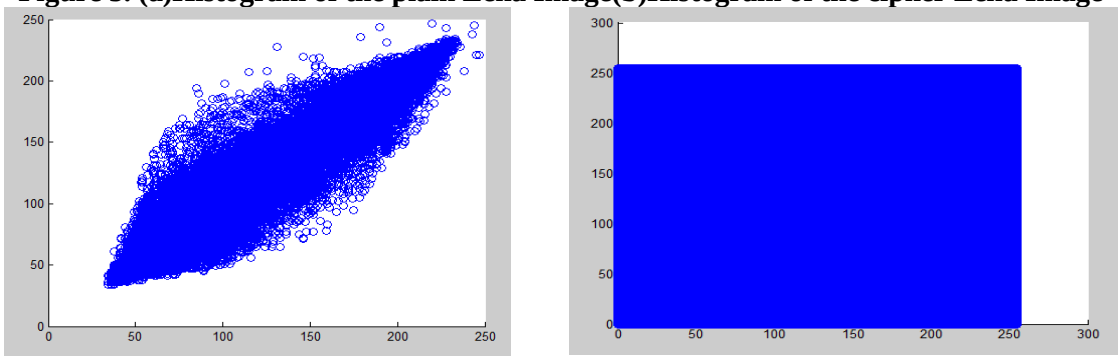
Figure 4(a): Original Image, (b): Encrypted Image, (c): Decrypted Image.

This fig shows the histograms of the some enciphered images as well as its plain images that have widely different content. One of commonly example among them is presented in Figure 5(a,b). The histogram of a plain-image consists of big spikes as presented in Fig. 5(a). The histogram of the ciphered-image is presented in Fig. 5(b), it is equally observed differ from that of the plain-image, and bears no statistical resemblance to the plain- image, Hence does not provide any clue to utilize any statistical attack on the PFCS image and text enciphering code**.**



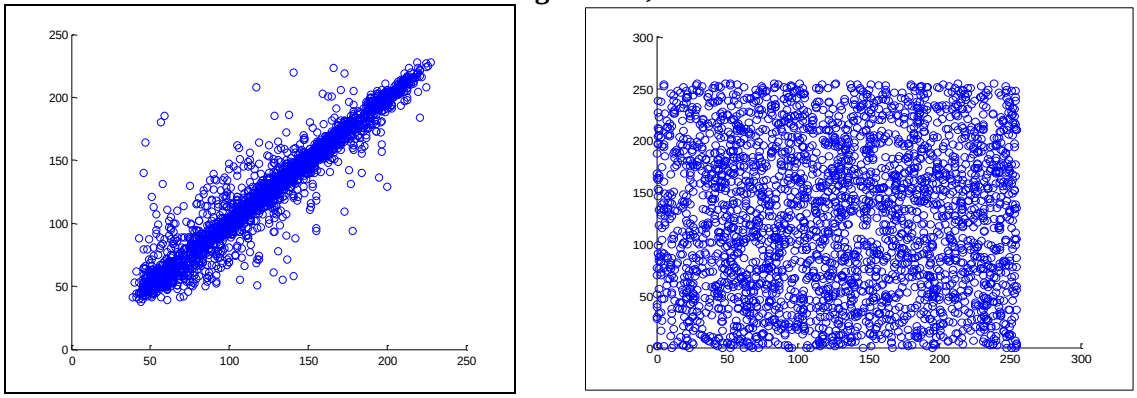
(a) (b)

Figure 5. (a)Histogram of the plain Lena Image(b)Histogram of the cipher Lena Image



(a) (b)

Figure 6. (a):Correlation for Vertical Plain lena image=0.9387;(b):Correlation for Vertical cipher lena image=0.004;



(a) **(b)**

Figure 7. (a) Correlation for Diagonal plain lena image=0.8743,(b)Correlation for Diagonal cipher lena image=0.0038

Key sensitivity:

The optimal ciphers systems must be sense with respect to secretive keys, i.e., alteration of each bit in key must generate a completely different cipher-text**.** Fig 8 shows the ASCII codes of two cipher-texts corresponding to each plain-text use two observed different secretive key shows that both cipher-texts, which corresponds to the same plain-data (text or image) and obtained by using slightly different secret keys, are completely different and showing thereby sensitivity of the cryptosystem to the secret key. In the text, the results stated in the Table3 below:

Table 3. Key Sensitivity values for different text length

Text Length (bytes)	No. Rounds	Key Sensitivity values
1325	2	0.0532
1325	4	0.0312
1325	6	0.0152
6428	2	0.0432
6428	4	0.0217
4628	6	0.0019
11133	2	0.0732
11133	4	0.0419
11133	6	0.0132

While in the other hand, For example in cat image we will encrypt the image with these initial values of secret key as shown in figure 8 a and b
x0=0.6234265487668976;
y0=0.9854342287239876;
z0=0.8376879934565543;
x1=0.6532765454387633;
y1=0.8812988834127654;
z1=0.4145233376598734;
w1=0.9334887655433345;
x=0.3331657888887654;
y=0.5698765543435255; z=0.8534877765653444;
w=0.7376542345267844; r1=10; r2=20; k=5;
as shown in figure 8(c): while in decryption of the same image by change the 16_{th} digit in the values above by +1 or -1 the 16_{th} as the following:
x0=0.6234265487668975;
y0=0.9854342287239877;
z0=0.8376879934565544;
x1=0.6532765454387634;
y1=0.8812988834127655;
z1=0.4145233376598735;
w1=0.9334887655433344;
x=0.3331657888887655; y=0.5698765543435254;
z=0.8534877765653445; w=0.7376542345267845;
r1=**11**; r2=**21**; k=**6**;
as shown in figure 8 (c):

Figure 8. (a): Original image, (b): Encrypted Image, (c): Decrypted image with wrong key

Table 4. Entropy values for different Images size

Image Name	Image Size	No. Rounds	Entropy value
Lena image	512*512	2	7.9592
Lena image	512*512	4	7.9950
Lena image	512*512	6	7.9986
Camera man image	512*512	2	7.9395
Camera man image	512*512	4	7.9897
Camera man image	512*512	6	7.9978
Cat image	186*271*3	2	7.9700
Cat image	186*271*3	4	7.9954
Cat image	186*271*3	6	7.9982

Encryption/decryption time calculation

One of the important measures of security cryptographic applications is an implementation time (15), also the factor of the time consider important to the chaotic. Table 5 states the time measure according to different images size, As shown in Table5 below:

Table 5. Encryption and Decryption time for different images

Image Name	Image Size	No. Rounds	Enc Time (Sec)	Dec Time (Sec)
Lena image	512*512	2	150.88	140.56
Lena image	512*512	4	378.04	340.13
Lena image	512*512	6	542.83	520.19
Camera man image	512*512	2	112.566	90.431
Camera man image	512*512	4	376.999	330.715
Camera man image	512*512	6	523.118	511.651
Cat image	186*271*3	2	89.613	88.540
Cat image	186*271*3	4	179.408	179.448
Cat image	186*271*3	6	267.122	266.114

Conclusions:

In this section, the conclusion of the proposed cryptosystem for two study cases; texts and images will be discussed as follows:

1. The Feistel Cipher design is not an easy task. First of all we must take into account that the decryption algorithm for Feistel Cipher is the same of encryption algorithm. So we must choose efficient chaotic maps that can be used for encryption and decryption and at the same time appropriate to confusion and diffusion principles. Second numerous issues must be viewed as precisely to stay away from potential security defects and to get desired performance. For instance, the key space sufficiently vast to brut fort attack, impact of a solitary of a single plaintext bits over as much of the cipher text as possible so as to hide the statistical structure of the plaintext in order to avoid the known-plaintext attack and the chosen-plaintext attack and so on.
2. The results of several experimental show that the mixing of several chaotic maps (4D Lorenz map, 1D Gauess Iterated map, 2D Standard map, and 3D Logistic map) for designing the proposed cryptosystem is very good choice because theses chaotic maps will increase the confusion and diffusion in the algorithm.
3. Ability of generation a Peseudo-random or random numbers based on chaotic map systems considered as a powerful tool in generating pseudo random sequences with high sensitivity and overcame all problems and difficulties which face designers of cipher systems. The proposed Peseudo-random numbers can be utilized in cryptographic applications.
4. The using of lookup table for substitution in the proposed cryptosystem is very efficient because the same byte is substituted to different byte in each round compared to fixed S-box where the same byte is substituted to same byte in each round. This is because in the proposed cipher, new S-box is created after each round; one S-box for each round.
5. In this paper, a new Feistel cipher algorithm is presented based on chaotic maps for enciphering and deciphering of texts and image. All the samples and experimental analysis show that the PFCS has very big key space, High sense to secretive key, as shown in Table 3, and Fig. 8 (a,b,c). Has information entropy near to optimal value(8) , as shown in Tables 2 and 4. And has low correlation coefficients close to the ideal value 0**.** As shown in Tables 2 and 3, and Figs. 6(a,b), and 7 (a,b). Has uniform histogram as shown in Figures 3(a,b), and 5(a,b) . We also conclude that's the number of rounds factor has a great significant to enhance the security of the proposed system, whereas all Tables about security tests proved that's the two case studies texts and images, when we increase the number of rounds, the proposed system will be more secure, more diffusion, more confusion as shown in Tables 2, 3, 4 and 5. Hence, and after all the results we can say that all the tests proved the security, robustness and effectiveness of the PFCS.

Conflicts of Interest: None.

References:

1. Tao Y, Chai Wu, Leon O. Cryptography Based on Chaotic Systems. IEEE Trans. on CAS-I, 1997; 44 (5) : 469-472.
2. Zaibi G, Peyrard F, Kachouri A . Efficient and Secure Chaotic S-Box For Wireless Sensor Network. Security and Communication Networks, 2014; 7(2) : 279-292.
3. Oher L, Kwon M, Park J, Lee S. Secure communication Based on Chaotic Synchronization via Interval Time-Varying Delay Feedback Control. Nonlinear Dynamics, Published in scopus, 2011; 63(1-2): 239–252.
4. Rhol K, Luo Y, Wang W . Finite-Time Stochastic Combination Synchronization of Three Different Chaotic Systems and its Application in Secure Communication. Published in scopus, Chaos, 2012; 22(2) , Article ID 023109.
5. Cao A. A New Hybrid Chaotic Map and its Application on Image Encryption and Hiding. Mathematical Problems in Engineering, Published in scopus, 2013, Article ID 728375.
6. Narendra K,Pareek,Vinod P, Sud K. Block Cipher Using 1D and 2D Chaotic Maps.Int. J. Information and Communication Technology, 2010; 2(3).
7. Wenyang L,Xiaomin W, Wenfang K. A Lightweight Block Cipher Based on Chaotic Maps. IEEE,2014, University of Chengdu,chaina.
8. Jean D , Joseph Y , Jean S , Mohamadou A, Laurent B , Monica B. A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine

Equation. Computer science and applications, 2014; 1(4) : 232-243.
9. Xiao J, Zha W, Yang L, Miao Z , Lianjie X . A Novel Compound Chaotic Block Cipher For Wireless Sensor Networks. Communications in Nonlinear Sci and Numer Simul, 2015; 22(1-3): 120-133.
10. Xiangjun W, Haibin K, Jürgen K. A New Color Image Encryption Scheme Based on DNA Sequences and Multiple Improved 1D Chaotic Maps. Published by Elsevier B.V. Available from: http://dx.doi.org/10.1016/j.asoc.2015.08.008 1568-4946/ 2015 .
11. Robert C. Chaos and Nonlinear Dynamics: An Introduction For Scientists and Engineers, Oxford, Univ. Press, New York, 2004.
12. Guangyun Z, Fuchen Z, Xiaofeng L, Da L, Ping Z. On the Dynamics of New 4D Lorenz-Type Chaos Systems. Zhang et al. Advances in Difference Equations 2017:2017, DOI:10.1186/s13662-017-1280-5
13. William S. Cryptography and Network Security. Fifth Edition: Protocols and Practice,2011.
14. Kadir A, Hamdulla A, Guo W. Color Image Encryption Using Skew Tent Map and Hyper Chaotic System of 6th-Order CNN, Optics ,2014; 125 (5) : 1671–1675.
15. Zhang Y, Wang X. A New Image Encryption Algorithm Based on Non-Adjacent Coupled Map Lattices, Appl. Soft Computing, 2015; 26 (1) : 10–20.

تصميم شفرة جديدة بالأعتماد على هيكلية فستل والدوال الفوضوية

اخلاص عباس البحراني **ريام نوري جواد**

قسم علوم الحاسوب، كلية التربية، الجامعة المستنصرية، بغداد ، العراق .

الخلاصة:

الأنظمة الفوضوية اثبتت فائدتها وكفاءتها وفعاليتها في علم التشفير، وخلال هذا العمل تم اقتراح نظام تشفير جديد بالأعتماد على هيكلية نظام تشفير Feistel مع الأنظمة الفوضية (Chaos Systems) مع حجم مفتاح متغير طبقا لحجم الرسالة الواضحة او النص الواضح. مقارنة مع شفرات الأنظمة القديمة التقليدية مثل الشفرات التي اعتمدت في تصميمها على هيكلية Feistel .عمليات التشويش والانتشار ستتضمن هنا صناديق ديناميكية متغيرة السلوك خاصة بالاستبدال، وصناديق ديناميكية متغيرة السلوك خاصة بالتعويض ،والتي تتولد مرة واحدة فقط وتعتبر ثابتة.

بينما بأستخدام نظام التحويلات الفوضوية (Chaotic Maps) والذي يسمى نظام التشفير المقترح بالأعتماد على هيكلية Feistel والأنظمة الفوضوية (CPFCS) هنا جعلنا عملية التشويش والأنتشار تعتمد على سلوك ديناميكي متغير بالأعتماد على نظام التحويلات القياسية ونظام Lorenz الفوضوي ، الأول يستخدم للتعويض والثاني يستخدم لعمليات التبديل .

نظام التشفير المقترح يستخدم نفس العمل (نفس الطريقة او الستراتيجية) لكلا العمليتين; التشفير وفك التشفير ،الشفرة المقترحة تعمل على اكثر من 500 بايت (4000 بت)، اي استخدام نظام التشفير الكتلي للنص القابل للقراءة (النص الواضح) بأستخدام ستة جولات six (rounds). من ضمن العامل الأساسي للشفرة خطوة العمليات الخاصة بحساب Round Function هو بناء جدول ديناميكي بالأعتماد على المعايير القياسية ثنائية الأبعاد ، استخدم لتحسين التعقيد وزيادة الغموض والانتشار للنص المشفر. ايضا استخدام الدالة اللوجستية Logistic Map ثلاثي الأبعاد استخدم لتوليد سلاسل الأرقام الثنائية (binary sequence) للمفتاح، وايضا توليد صناديق الأستبدال الديناميكية لزيادة الغموض . ثلاثة صور متغيرة الحجم وثلاثة اطوال مختلفة الطول طبقت في النظام المقترح ، والنتائج للنظام المقترح واختبارات الأمنية اثبتت امكانية تطبيق النظام المقترح في حماية وأمن البيانات.

الكلمات المفتاحية: الشفرة الكتلية، الدوال الفوضوية، شفرة فستل، الدالة اللوجستية، الدوال القياسية.