Chaos-based Block Ciphers: An Overview

S. El Assad[1,*], M. Farajallah [1], C. Vladeanu[2]
[1] *IETR UMR CNRS 6164, Image team -site of Nantes, Nantes, France*
[2]*Polytechnical University of Bucharest, Romania*
[*]Corresponding author (E-mail: safwan.elassad@univ-nantes.fr)

Abstract— **A variety of chaos-based cryptosystems have been investigated during the last decade. Most of them are based on the structure of Fridrich, which is based on the traditional confusion-diffusion architecture proposed by Shannon. Compared with traditional cryptosystems (DES, 3DES, AES, etc.), the chaos-based cryptosystems are more flexible, more modular and easier to be implemented, which make them more suitable for large scale-data encryption, such as images and videos. The heart of any chaos-based cryptosystem is the chaotic generator and so, a part of the efficiency (robustness, speed) of the system depends greatly on it. In this talk, we give an overview of the state of the art of chaos-based block ciphers and we describe some of our schemes already proposed. Also, we will focus on the essential characteristics of the digital chaotic generator that we published in October 2011 as a French patent and also we published in 2013 an extension of the patent in Europe, China, Japan, and USA. The needed performance of a chaos-based block cipher in terms of security level and speed of calculus depends on the considered application. There is a compromise between the security and the speed of the calculation. The security of these block ciphers will be analyzed.**

Keywords: Chaos-based cryptosystems; digital chaotic generator; security analysis.

I. Introduction (Heading 1)

Chaotic cryptography is a multidisciplinary field covering many different areas, such as: nonlinear dynamics, cryptology communications, etc. Nowadays, with the development of Internet technology, there is a growing demand for cryptographic techniques to secure transmitted multimedia contents (audios, images, videos) over the Internet and mobile-phone networks.

Traditional encryption algorithms, such as AES, DES, 3DES, RSA, etc., are not very suitable for multimedia data. Therefore, for two decades several chaos-based cryptosystems have been proposed. Among these, the chaos-based block ciphers were mostly considered. A general structure of any chaos-based cryptosystem contains two layers: a confusion layer and a diffusion one. The confusion effect measures how a change in the secret key affects the ciphered data. The diffusion effect assesses how a change in the plain data affects the ciphered one. Several chaos-based cryptosystems are based on the Fridrich structure in which the confusion and diffusion layers are separated [1 - 6]. In all chaos-based cryptosystems, the chaotic generator is an important component of the system and so, a part of the effectiveness of the cryptographic system depends significantly on it. Most of the proposed chaotic generators in the literature work with floating data operation. This may cause problems when the computers' processors of the sender and the receiver are different. To overcome this problem, it is necessary to work with a fixed finite precision of N bits.

In this paper, we give an overview of some chaos-based cryptosystems that adopt the Fridrich structure [1]. The rest of the paper is organized as follows. Section II describes the main chaos-based cryptosystems in the literature. In Section III, we present their performance. Section IV describes the main characteristics of the complete proposed chaotic generator, before concluding.

II. Main Chaos-based Cryptosystems in the Literature

A general structure of chaos-based cryptosystems is given in Fig. 1, where the confusion and the diffusion layers are working separately. First, the confusion process is applied rc times on the block (or on the whole image), then the diffusion process is applied rd times on the output of the confusion process, and finally, the two processes are repeated r times. As we can see, both layers required image-scanning (for rc = rd = r =1).

The confusion process is usually done by substitution operation. The substitution can be achieved by any 2-D chaotic permutation map, such as: Cat map, Standard map, or Baker map, and also, by using any nonlinear chaotic function as the 1-D finite state Skew tent map.

In the permutation case, the image pixels are relocated, but their values remain unchanged.

The diffusion process changes the statistical properties of the plain-image by spreading the influence of each bit of the plain-image over all the ciphered ones. The diffusion process is essential for any secure cryptosystem, otherwise it is easy to break the system.

The dynamic keys *Kc* and *Kd* are supplied by the chaotic generator(s) (keys generator(s)).

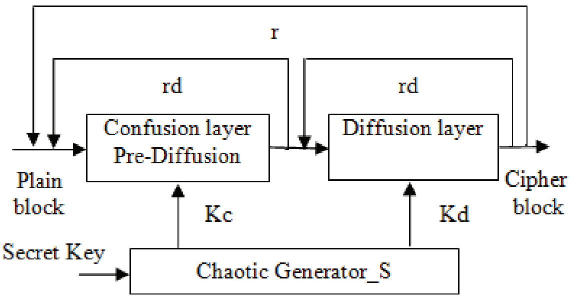


Figure 1. General structure of chaos-based cryptosystems

978-1-4799-2385-4/14/$31.00 ©2014 IEEE

In Fridrich scheme, the confusion is achieved by permuting all the pixels as a whole, using one of the three types of 2-D chaotic maps, namely, Standard map, Cat map, and generalized Baker map.

The diffusion process changes sequentially the pixel values, in such a manner that the change to a particular pixel depends on the accumulated effect of all the previous pixel values.

Chen et al. [2], used 3-D Cat map to shuffle the positions of image pixels and a quantized Logistic map in the diffusion stage. Lian et al. [3], pointed out that the key space of the Standard map is larger than the keys space of the Cat and Baker maps. Then, they employed in the substitution stage a Standard map instead of Cat or Baker maps, while keeping the quantized Logistic map in the diffusion stage. Each pixel v(i) of the 2-D permuted image (or block) is scanned sequentially and then processed by the diffusion stage as follows:

$$\begin{cases} c(i) = v(i) \oplus q\left\{ f\left[c(i-1)\right], L \right\} \\ c(-1) = Kd, \quad L = 8 \end{cases} \tag{1}$$

where c(i) and c(i-1) are the values of the ith and the (i-1)th pixels of the diffused image, respectively. The c(-1) is obtained from the diffusion key Kd. The nonlinear function f(.), namely the Logistic map, and the bit extraction function q(.), that extract L bits just after the decimal point, are defined by equation (2):

$$\begin{cases} f\left[c(i-1)\right] = 4c(i-1) \times \left[1 - c(i-1)\right] \\ q\left[b, L\right] = \left\lfloor b \times 2^{L} \right\rfloor, \quad b = 0.b_1 b_2 \cdots b_L \cdots, \quad b_j \in \left[0, 1\right] \end{cases} \tag{2}$$

Equation (1) indicates that the new diffused pixel value c(i) is obtained by xor-ing the current pixel value v(i) of the permuted image (block) with the quantized Logistic map taking the previous diffused value c(i-1) as input. Because the previous diffused pixel affects the current one, a tiny change in the plain image is reflected in several pixels in the cipher image and so the diffusion effect is achieved.

To speed up the diffusion layer, equation (1) can be implemented as a lookup table. Indeed, there are at most 2L distinct values and then, all the possible outputs of the bit extraction function q{f[c(i-1), L)} can be pre-computed and stored in a lookup table.

In the cryptosystem of Chen et al., as well in the cryptosystem of Lian et al., the confusion and diffusion effects are solely contributed by the substitution and diffusion stages. So, too many rounds are required to achieve a satisfactory level of security. Specifically, a total of 16 permutation and 4 diffusion rounds are necessary in Lian et al.'s cryptosystem.

To accelerate the overall encryption time, while keeping similar security performance, Wong et al. [4], introduced a certain diffusion effect in the substitution stage of Lian et al.'s cryptosystem. This is done by simple sequential add-and-shift operation, while the diffusion process remains unchanged (see Fig. 1). Consequently, both the shuffling of pixels and the change of their values are carried out at the same time in the substitution stage achieved by equation (3).

$$\begin{cases} v(i) = Cyc\left\{ Mod\left\{\left[p(i) + v(i-1)\right], Q\right\}, LSB_3\left[v(i-1)\right]\right\} \\ v(-1) = Kc \in \left[1, (Q-1)\right], \quad Q = 2^8 = 256 \end{cases} \tag{3}$$

As we can see, before relocating the pixels, diffusion effect is injected by adding the current pixel value p(i) of the plain image to the previous permuted pixel v(i-1) and then performs a cyclic shift on their sum. The function Cyc[s, z] performs the z-bit right cyclic shift on the binary sequence s; LSB3(s) refers to the value of the least three significant bits of s and v(i) is resultant pixel value in the permuted image. The used key generator is composed by skewed tent maps.

Wang et al. [5], proposed a chaos-based image encryption algorithm with variable control parameters, based on the permutation-diffusion structure as Lian et al.'s cryptosystem. However, the dynamic keys (control parameters) Kc and Kd used in the permutation stage and the diffusion stage, respectively, depend not only on the secret key, but also on the plain image for Kc and on the cipher feedback for Kd, related to the plain-image (see Fig. 2). Two Logistic maps are used as keys generators, i.e., one to produce Kc, and the other one to produce Kd, and as diffusion layer.

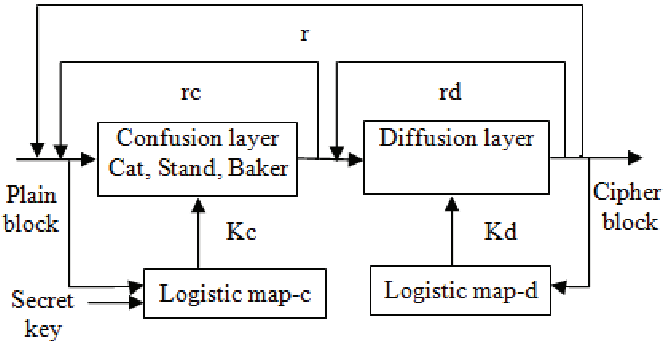


Figure 2. Chaos-based cryptosystem of Wang et al.

With this structure, the number of iterations of the Logistic map-c, used to produce Kc, is related to the plain-image by the following relation:

$$nc = p(0,0) + 100 \tag{4}$$

where p(0,0) is the grey-level value of the (0, 0) pixel. The pixel at position (0, 0) must be changed after permutation and then, it is necessary to introduce ri and rj parameters in the 2-D equation maps that are used as permutation layer.

The number of iterations of the Logistic map-d, used to produce Kd, is given by equation (5):

$$nd = 1 + c(i)\%2 = g\left[p(i)\right] \tag{5}$$

where c(i) is the current diffused pixel produced by the following equation:

$$\begin{cases} c(i) = f(i) \oplus \left\{ Mod\left\{\left[p(i) \oplus f(i)\right], Q\right\}\right\} \oplus c(i-1) \\ f(i) = Mod\left\{\left[xd(i) \times 2^{2L}\right], 2^{L}\right\} \end{cases} \tag{6}$$

The function f(i) is obtained from the current state of the Logistc map-d, p(i) is the current plain-pixel, and c(i-1) is the previous diffused pixel. L = log2G and G is the number of possible grey levels in image pixels. Here, L = 8, because G = 256 for the grey-scale image.

From equation (4), using the same secret key, different plain-images give distinct Kc key values and non identical permuted images. The attacker can't obtain useful information by encrypting some special images since the resultant information is only related to those chosen-images. From equations (5) and (6), when different plain-images are encrypted, the corresponding Kd key values are different. As a result, the cryptosystem can resist to known-plaintext and chosen-plaintext attacks effectively.

In Farajallah et al., [6], we proposed an efficient chaos-based cryptosystem in terms of robustness against all known attacks and of computing encryption time.

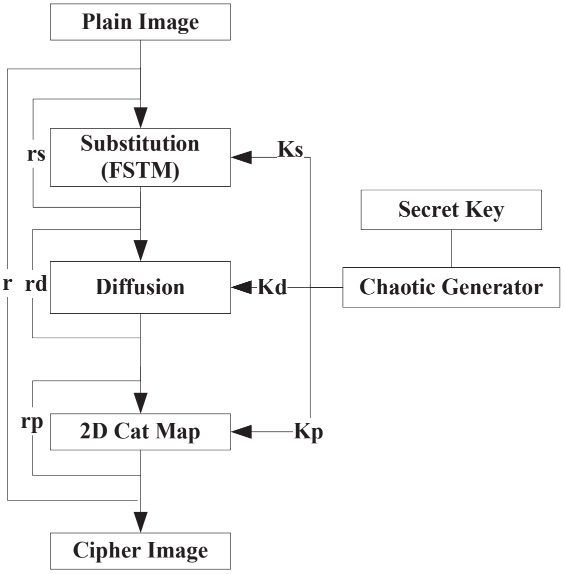


Figure 3. Chaos-based cryptosystem of Farajallah et al., first type

The proposed scheme, given in Fig. 3, comprises 3 layers: a substitution layer achieved by a Finite Skew Tent Map (FSTM); a diffusion layer implemented by simple sequential modulo operation between the current substituted pixel and the previous substituted one; and finally, a permutation layer achieved by a modified 2D-Cat map. The chaotic generator consists of a very simplified version of the one proposed by El Assad et al., in their Patent [7].

For each iteration j, the FSTM given by equation (7) changes the value X of each pixel in a block into a new value Y, in accordance with a dynamic key Ksj = [aj||a0].

$$Y = S_a(X) = \begin{cases} \left\lceil \dfrac{Q}{a_j}\times X \right\rceil + a0 & 0 \le X \le a_j \\ \left\lfloor \dfrac{Q}{Q-a_j}\times (Q-X) \right\rfloor + 1 & a_j < X < Q \end{cases} \qquad (7)$$

where X, Y are limited to {0, 1, 2,…,Q-1}, and aj, a0 are limited to {1, 2,…,Q-1}, with Q = 256.

Equation (7) and its invertible equation are implemented by lookup tables for fast execution.

The diffusion effect on each byte is introduced by the following simple and fast diffusion equation:

$$\begin{cases} X(i) = Mod\{[X(i) + X(i-1), Q]\} \\ i = 0, 1, 2, \cdots, blocksize - 1 \end{cases} \qquad (8)$$

The first pixel value X(-1) = Kd is produced by the chaotic generator. Finally on each diffused block, we apply a permutation layer achieved by the following modified Cat map:

$$\begin{bmatrix} i_n \\ j_n \end{bmatrix} = Mod\left\{ \begin{pmatrix} 1 & u \\ v & 1+u\times v \end{pmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} ri+rj \\ rj \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right\} \qquad (9)$$

The block size is M2, the dynamic keys u, v, ri, and rj of the system are in the range of [0, M-1].

In the decryption side, we introduce the inverse layers of the encryption ones and the dynamic keys are used in reverse order.

III. SECURITY ANALYSIS AND EXPERIMENTAL RESULTS.

In this section, we present a comparative security analysis and experimental results of the presented chaos-based cryptosystems.

A. Key space, key sensitivity and plaintext sensitivity

All presented algorithms have large key space and so can avoid brute-force attack. They are also, very sensitive to the secret key and to the plaintext sensitivity attack. In order to test the influence of one bit change in the secret key or in the plain-image on the ciphered one, two common measures are used: the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). The optimal values of these two parameters are 99.609% and 33.463. Table 1, presents, for three algorithms, the number of necessary iterations (r, rc, rd) to achieve satisfactory values of NPCR and UACI.

Table I
Comparative results of NPCR and UACI

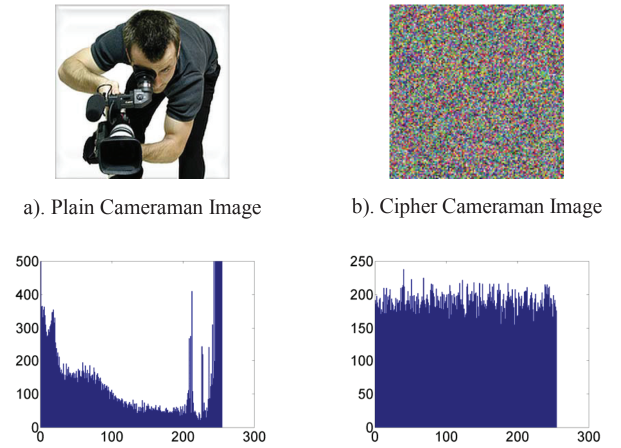
Nb of necessary iterations (*r*, *rc*, *rd*)			NPCR% UACI%		
Lian	Wong	Faraja	Lian	Wong	Faraja
(6, 3, 1)	(2, 2, 1)	(1, 1, 1)	99.586 33.419	99.608 33.427	99.609 33.462

B. *Statistical analysis: Histogram and correlation*

To resist the statistical attacks, in the encrypted image, the histogram should be uniformly distributed and adjacent pixels should have a correlation close to zero. All described algorithms fulfill the above conditions, and we present below some results obtained by us in Farajallah et al., [6].

We give in Fig. 4 (a) and (b) the plain-image of Cameraman and its ciphered image, and in Fig. 4, (c), (d), we give the histograms of the plain-image and of the ciphered one. As we can see, the histogram of the ciphered image is almost uniform, because its measured chi-square value is 255.12 and it is less than the theoretical one, i.e., equal to 293 in case of alpha=0.05.

To measure the correlation coefficient, we randomly select 8000 pairs of adjacent pixels in vertical, horizontal, and diagonal directions from the plain-image and its ciphered image. Table II presents the obtained results and in Fig. 5, we



a). Plain Cameraman Image
b). Cipher Cameraman Image
c). Histogram of the Plain image
d). Histogram of the cipher image

Figure 4. Histograms of the plain and its ciphered images

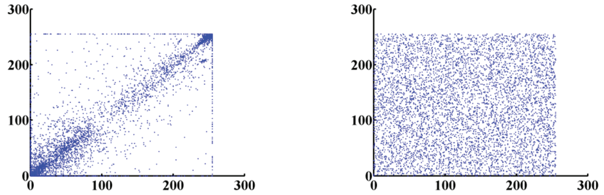


Figure 5. a) Plain-image correlation of adjacent pixels

Figure 5. b) Ciphered-image correlation of adjacent pixels

Figure 4. Correlation of the plain and its ciphered images

show the correlation curves of the adjacent pixels in horizontal direction for the plain image and its ciphered version. These results imply a high security level against statistical attack.

Table II
Comparative results of NPCR and UACI

	Plain image	Ciphered image
Horizontal	0.898492	0.010523
Vertical	0.925182	0.010650
Diagonal	0.851377	0.010842

C. *Speed results*

To compare the speed performance of presented algorithms, we took Lena image of size 3x512x512 as reference and we used a PC with 3.1 GHz Intel processor and 4 GB RAM. In table III, we present the obtained comparative speed results: the average encryption/decryption time in milliseconds (ms), the encryption throughput (ET) in Mega Bytes per second (MBps), and the number of cycles that are necessary to encrypt one byte.

Table III
Speed results

Average Enc time (ms) Average Dec time (ms)				ET (MBps) Number of Cycles per Byte			
Lian	Wong	Wang	Faraja	Lian	Wong	Wang	Faraja
349 358	95.6 104.3	159.6 -	24 25	214 754	754 398	5022 264	32.77 94.6

We can see from table III that Farajallah et al.'s algorithm is 2.8 times faster than Wang et al.'s algorithm, which is the best as compared to the other two algorithms.

IV. MAIN CHARACTERISTICS OF THE CHAOTIC GENERATOR.

In this section, we give the main innovative characteristics of the complete generator of discrete chaotic sequences published as a Patent [7]:

Modularity, allowing to operate in a simple way with between 1 to 28 basic chaotic generators, each one having the possibility to work with variable delays (from zero to three), according to the degree of necessary security and the speed required by the application. The size of the secret key is variable from 256 to 15540 bits.

Adaptability with regard to the planned applications in the field of information hiding and security.

Bit rate performance is larger than 82 Mbit/s.

Robustness against statistical attacks (NIST test); generated orbits have lengths that can reach centuries.

V. CONCUSION

In this paper, we gave an overview of chaos-based cryptosystems that are based on Fridrich structure and we compared their performance in terms of robustness and speed. Also, we presented the main innovative characteristics of the chaotic generator published as a Patent.

REFERENCES

[1] J. Fridrich , "Symmetric ciphers based on two-dimensional chaotic maps". Int. J. Bifurcat Chaos, vol. 8, no. 6, 1998, pp. 1259-1284.

[2] G. Chen, Y. Mao, C. K. Chui, "A symmetric image encryption schemes based on 3D chaotic cat maps". Chaos Solitons and Fractals vol. 21, 2004, pp. 749-761.

[3] S. Lian, J. Sun, Z. Wang, "A bloc cipher based on a suitable use of the chaotic standard map". Chaos Solitons and Fractals, vol. 26, 2005, pp. 117-129.

[4] K-W. Wong, B. S-H. Kwok, W-S. Law, "A fast image encryption scheme based on chaotic standard map". Physics Letters A, vol. 372, no. 15, 2008, pp. 2645-2652.

[5] Y. Wang, K-W. Wong, X. Liao, T. Xiang, G. Chen, "A chaos-based image encryption algorithm with variable control parameters". Chaos Solitons and Fractals vol. 41, 2009, pp. 1773-1783.

[6] M. Farajallah, S. El Assad, M. Chetto, "Dynamic adjustment of the chaos-based security in real-time energy harvesting sensors". IEEE, International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, August 2013, pp. 282-289.

[7] S. El Assad, H. Noura, "Generator of chaotic Sequences and corresponding generating system" WO Patent WO/2011/121,218,2011.