Chaos-based cryptosystems using dependent diffusion: An overview

Mohammad Abu Taha, Safwan El Assad, Mousa Farajallah, Audrey Queudet, Olivier Deforges

Chaos-based Cryptosystems using Dependent Diffusion : An Overview

Mohammed AbuTaha*, Safwan El Assad*, Mousa Farajallah*, Audrey Queudet†, Olivier Deforge‡,
*Institut d'Electronique et de Télécommunications de Rennes, Polytech Nantes, France, UMR CNRS 6164
Email: mohammad.abu-taha@univ-nantes.fr , safwan.elassad@univ-nantes.fr , mousa.farajallah@univ-nantes.fr
‡Institut d'Electronique et de Télécommunications de Rennes,INSA, France, UMR CNRS 6164
Email: olivier.deforges@insa-rennes.f
†University of Nantes, IRCCyN, Real-time team
Email:audrey.queudet@univ-nantes.fr

***Abstract*—Based on some important properties of chaos, such as ergodicity, quasi-randomness, and high sensitivity to the secret key, chaos is a hot research field in secured communication, and recently a variety of chaos-based cryptosystems have been proposed for achieving the confidentiality of transmitted images over public channels. Most of chaos-based encryption algorithms are based on the Fridrich structure, which use separate confusion-diffusion layers. In this paper, we give an overview of some chaos-based block cryptosystems, including our cryptosystem, using dependent diffusion, in which the confusion process and diffusion process are performed sequentially on each pixel of the plain plain image. This kind of cryptosystems are more efficient than the traditional confusion-diffusion architecture of Shannon. A comparison study of efficiency in terms of speed performance and of robustness against cryptanalysis is done.**

Keywords: chaos-based cryptosystems; dependent confusion-diffusion; speed performance; security analysis.

I. Introduction

Since two decades, many chaos-based image encryption schemes have been proposed. The typical structure of these encryption schemes is based on separate confusion-diffusion layers as that is done in Fridrich architecture [1], [2], [3]. El Assad et al., [4] have given an overview of main chaos-based cryptosystems, using Fridrich's structure. Recently, many chaos-based cryptosystems, more efficient in terms of time consuming and of resistance against cryptanalysis, than the previous ones have been investigated [5], [6], [7], [8], and [9]. These cryptosystems are based on dependent confusion-diffusion layers, generally achieved by substitution-diffusion processes that use dynamic keys supplied by a chaotic sequence. The chaotic sequence is produced by a chaotic generator which is the heart of any chaos-based cryptosystem and so a big part of the efficiency of the system depends on it. Compared to the conventional cryptographic algorithms (3DES, AES), chaos-based cryptosystems have several advantages such as: a very high security level, more flexibility, more modularity, a low power consuming, and easily implemented, which make them more suitable for large scale-data encryption, such as images and videos. In this paper we present an overview of chaos based cryptosystems, including our cryptosystem, using dependent confusion-diffusion. The rest of the paper is organized as follows. In section II, we describe four known chaos based cryptosystems of the literature. Section III, gives a comparative results of the speed performance and the security analysis of these cryptosystems, before concluding.

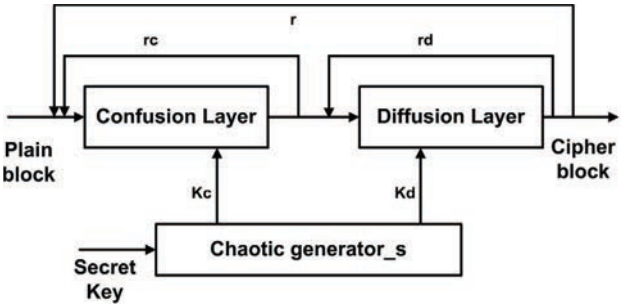


Fig. 1: General structure of chaos-based cryptosystems

II. Principal chaos-based cryptosystems using dependent diffusion of the literature

All chaos-based and non chaos cryptosystems must achieve the confusion and diffusion effects. The confusion effect is measured by how much a change in the secret key affect the ciphered message. The diffusion effect is measured by how much a change in the plain message affect the ciphered message. In the literature there are mainly two types of chaos-based cryptosystems. The structure of the first type, as we can see in figure 1, is composed of two layers: a confusion layer followed by a diffusion layer that work separately. 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. Both layers required image-scanning (for rc = rd = r =1). Most of chaos-based cryptosystems of first type are considered insecure upon chosen/known plain text attacks. El Assad et al., [4] gave in their paper an overview of main chaos-based cryptosystems of first type.

The structure of the second type of cryptosystems is similar to the structure of the first type of cryptosystems, but the confusion and diffusion processes are performed sequentially on each pixel of the plain block or plain image as shown in figure2. This type of cryptosystems are more efficient, in

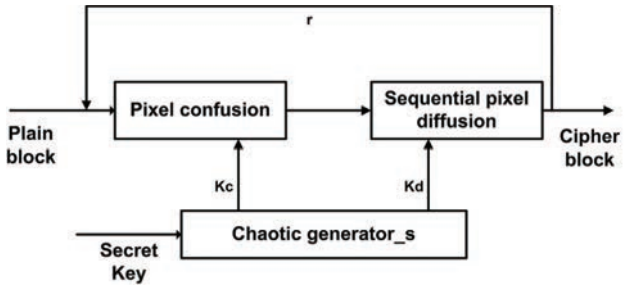


Fig. 2: Dependent diffusion structure of chaos-based cryptosystems

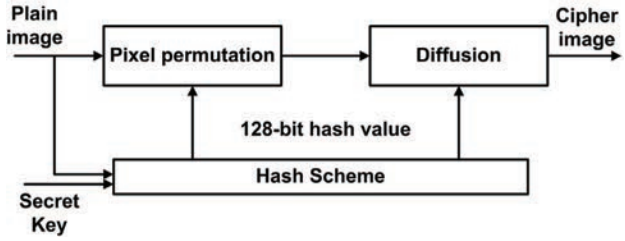


Fig. 3: Yang et al scheme

terms of security and speed performance, than the first type of cryptosystems. Indeed, first, the diffusion process at the pixel level is governed by the confusion process, second, a single scan of plain image pixels is needed to perform the confusion and diffusion effects.

In the following we will describe the main chaos-based cryptosystems of the second type including our cryptosystem. Yang et al., [5], used a permutation operation, as a confusion layer, achieved by a modified standard map to avoid the problem of permutation of the corner pixel $(s = 0, t = 0)$, while using a logistic map as a diffusion layer. In addition a keyed hash function is used to generate a $128 - bit$ hash value from both the plain image and the secret hash keys. The hash value plays the role of the key for encryption and decryption while the secret hash keys are used to authenticate the decrypted image. As we can see from the encryption scheme of figure 3, a one bit change in the plain image changing the dynamic keys of the processes of confusion and diffusion and then the encrypted image become completely different from the previous encrypted one. So, the immunity against known-chosen plain text attacks is easily obtained from one round. However for the decryption scheme it is necessary to transmit the secret encryption-decryption key for each new encrypted image.

In their paper, Wang et al [6] introduced the idea of mixing the two layers of permutation and diffusion into a single layer of dependent permutation-diffusion. As a result, one image scanning is required instead of two scanning stages, to accelerate the encryption algorithm. The main steps of Wang et.al., cryptosystem [6] are summarized in the following:

1) Division step: The image is divided into a number of blocks (num), each one is 64 pixels.
$$num = \frac{L \times P}{64} \quad (1)$$
Where L and P are the height and the width of the image, respectively.
2) First generation step: In this step, the used keys $K_0, K_1, ...K_{15}$ are generated, and each one is an 8-bit number.
3) Second generation step: 64 pseudorandom values are generated from the spatio-temporal chaotic map [10]. These pseudorandom values are defined below as $\Phi(i, j)$.
4) Diffusion step: The pixel values inside each block are modified based on the following equation:
$$\begin{aligned} &G_t(i,j) = cycl\{X, Y\} \\ &X = [P_t(i,j) \oplus \Phi(i,j) + C_t(i-1,j)]Mod\ G \\ &Y = LSB_3(C_t(i-1,(j-1)mod\ 8) \oplus \Phi(i,j)) \end{aligned} \quad (2)$$
Where: $G_t(i, j)$ is the number of levels in the gray image (here is 256 levels) and is performed for all pixels in the t^{th} block. $P_t(i, j)$ and $C_t(i, j)$ are the plain and the ciphered pixels at i and j positions, respectively.
5) Moving block positions: A block at the position t is moved to a new position according to the following equation:
$$t_{new} = \lfloor X(0) \times num \rfloor \quad (3)$$
$X(0)$ is a generated value from the used chaotic map. If the t_{new} is a non-visited place, then the block is moved to this place, otherwise, the t_{new} is incremented until the new value point to a non-visited place. As not all pixels inside a given block are permuted, this step is necessary to increase the security of this model to the statistical analysis attacks.
6) Exchanging lattice values: The pixels $C(7, i)$ and $C(7, (i+d)mod\ 8)$ are exchanged, where d is calculated as:
$$d = LSB_3(C(7, 0)) \quad (4)$$
7) First Block pixel exchange: $C_0(0, s)$ is exchanged with $C_{k_l}(7, 7)$ where: $s = LSB_3(k_l)$

$$k_l = \left\lfloor \frac{[K_0 \oplus K_1 \oplus K_2 \oplus ... K_{15}] \times num}{256} \right\rfloor$$

Steps, $3 \rightarrow 6$ are performed for all blocks, and repeated r rounds until the required security level is reached.

In Zhang's paper [7], two cryptosystems were designed based on Fridrich's architecture. The first one consists of a dependent diffusion layer based on the reverse 2-D cat map. The second algorithm presents new mapping from a pseudo-random position to another pseudo-random one for the confusion effect. The diffusion layer in the cryptosystems is based on the logistic map. In these versions, Zhang tried to achieve the confusion and the diffusion effects sequentially. Then, the effect of one ciphered pixel is transferred to the next one and so on. From this idea, only two rounds (in the

first version) and one round (in the second version) of the diffusion-confusion process are/is needed instead of many rounds of separated confusion and diffusion processes used in the traditional structures such as Fridrich cryptosystem and other cryptosystems. In the following, our work is directed to the first Zhang cryptosystem. The mathematical model of the first Zhang algorithm is:(Enc=the encryption process).

$$Enc = \begin{cases} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & p_i \\ qi & p_i q_i + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (ModN) \\ ciph(x,\ y) = arr(x',\ y') \oplus f(t) \\ t = ciph(x,\ y) \end{cases} \quad (5)$$

The general block diagram of the first Zhang cryptosystem is shown in Figure. 4. It consists of the following steps iterated n times (with $n > 0$):

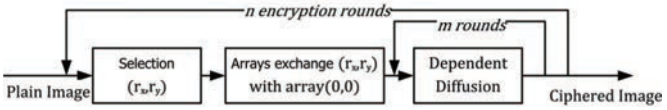


Fig. 4: Zhang image encryption cryptosystem architecture

1) Selection: this step generates a random pair $arr({r_x}^j,\ {r_y}^j)$ from the whole image. The values of ${r_x}^j$ and ${r_y}^j$ are calculated using equation (6) and equation (7), where j is a counter ranging from 0 to $n-1$ encryption rounds.

$${r_x}^j = (SQ_1(2000 + 100 + j) \times 10^9) \mod (512) \quad (6)$$

$${r_y}^j = (SQ_2(2000 + 100 + j) \times 10^9) \mod (512) \quad (7)$$

2) Array exchanges: the second step is to exchange the first byte $arr(0,\ 0)$ with the random byte from the previous step $arr({r_x}^j,\ {r_y}^j)$.
3) Dependent diffusion: then the cryptosystem goes to the dependent diffusion layer for m rounds ($m = 2$ in the Zhang algorithm case), which also includes three stages.
   a) New position estimation: in the dependent diffusion layer the first step is to calculate the new byte position $(x',\ y')$ from the old byte position $(x,\ y)$ using equation (8).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & p_i \\ qi & p_i q_i + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (ModN) \quad (8)$$

   where
   N is the size of the square test image.
   p_i and q_i are calculated using the following equations:

$$p_i = (SQ_1(2000 + i) \times 10^9) mod\ 512 \quad (9)$$

$$q_i = (SQ_2(2000 + i) \times 10^9) mod\ 512 \quad (10)$$

   The variable i in the last equations is a counter ranging from 0 to $m-1$. The two sequences SQ_1 and SQ_2 as can be seen in equations (6, 7, 9, and 10) are calculated using the following equation:

$$f(x_n) = \alpha \times x_{n-1}(1 - x_{n-1}) \quad (11)$$

   With the initial values x_{-1}=0.12345678912345 for SQ_1, and x_{-1}=0.67856746347633 for SQ_2. The value of α is set to 3.99999.
   b) Calculation of the local ciphered pixel: the next step of the dependent diffusion layer is to calculate the $ciph(x,\ y)$ value using the following equation:

$$ciph(x,\ y) = arr(x',\ y') \oplus f(t) \quad (12)$$

   where

$$f(t) = [\alpha(\frac{t}{1000}) \times [1 - \frac{t}{1000}] \times 1000] mod\ 256 \quad (13)$$

   c) Update of t: the last step of the dependent diffusion layer is to change the value of the t variable using equation (14).

$$t = ciph(x,\ y) \quad (14)$$

   The initial value t_0 is defined by the following equation:

$$t_0 = [4 \times key_d \times (1 - key_d) \times 1000] mod\ 256. \quad (15)$$

   Where the initial value of the key_d=0.33456434300001.

Farajallah et al., 2015 [8], and Farajallah [9] first they partially cryptanalysis the first algorithm of Zhang (described above) based on the following partial cryptanalysis equation.

$$arr_{k'_k} = ciph_k \oplus f(ciph_{k-1}) \quad (16)$$

where $k = x \times N + y$ and $k' = x' \times N + y'$ (see equation (8)), $arr_{k'_k}$ is the input pixel of the last dependent diffusion round (m) in the last encryption round (n) and $ciph_k$ is the ciphered pixel. As the function $f(t)$ is known, then equation (16) can be used to remove the diffusion effect of the last (m and n) rounds from the ciphered pixels. This allows recovery of a permuted version of the previous ciphered image. This removal gives the attacker the possibility to:

1) Decrease the dynamic key space of the whole cryptosystem.
2) Perform partial cryptanalysis of the Zhang cryptosystem for ($n = 1$, $m = 1$) and ($n = 1$, $m = 2$).
3) Decrease the UACI and NPCR values significantly.

Then, based on the previous analysis they proposed an efficient cryptosystem that overcome the weaknesses of Zhang cryptosystem while keeping a very high speed compared to the main chaos-based cryptosystem of the literature. The encryption side of the proposed cryptosystem is given in Figure 5), for the first block. Each pixel from the plain block ($p_0(k)$) is XOR-ed with the initial byte ($iv(k)$) from the initial vector (IV), then the output is XOR-ed with the discrete logistic map output to carry out the diffusion process. Then, the 8 least significant bits resulting from the diffusion

process ($LSB_8(y_0(k))$ are relocated using the modified 2-D cat map to obtain the ciphered pixel at the new position ($c_0(k_n)$). It is important to note that the input of the discrete logistic map is based on the previous ciphered pixel (since $c_0(k_n) = LSB_8(y_0(k))$ and the input of the discrete logistic map is 32 bits and the ciphered pixel is 8 bits. That is why the cryptosystem takes $(y_0(k-1))$ before the LSB_8 function and not after. For the first encrypted byte, the input of the discrete logistic map is Kd_m, and this value is re-initialized every new encryption round. Because the $c_0(k_n)$ is only a part of the logistic map input, it is impossible to recover $y_0(k-1)$ from $c_0(k_n)$ only. The encryption of the next blocks is almost the same. Each pixel from the plain block ($p_l(k)$) is XOR-ed with ciphered byte from the previous block at the same position(i.e., $c_{l-1}(k)$ to achieve the CBC mode). Then the rest of the operations are the same as in the first encryption block. The modified cat-map is given by the following equation

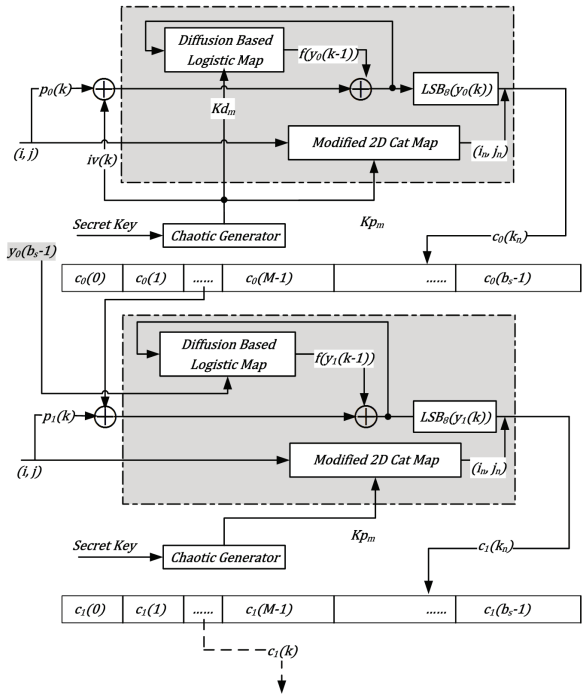


Fig. 5: Encryption structure of the proposed cryptosystem

$$\begin{bmatrix} i_n \\ j_n \end{bmatrix} = Mod\left(\begin{bmatrix} 1 & u \\ v & 1+uv \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} ri+rj \\ rj \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right) \quad (17)$$

(17) which is a one-to-one function, which means that each point of the square matrix can be transferred to exactly one unique point. So, instead of exchanging the values at the new position $(i_n,\ j_n)$ with the old one $(i,\ j)$, we use a transfer operation because of its speed compared to the swap operation that is usually used. The block size b_s is M^2 (M is the square root of the block size in our proposed cryptosystem). The system parameters u, v, ri and rj are in the range of $[0,\ M-1]$. The structure of the dynamic keys which are produced by the chaotic generator [11] during the permutation process is:

$$\begin{aligned} Kp &= \left[Kp_0 \parallel Kp_1 \parallel Kp_2 \parallel \cdots \parallel Kp_{r-1}\right] \\ Kp_m &= \left[u_m \parallel v_m \parallel ri_m \parallel rj_m\right] \end{aligned} \quad (18)$$

The modulo operation of equation (17) makes it a non-invertible equation. But it is still a reversible one. Thus, in the decryption part of the proposed cryptosystem, the reverse layer is also achieved by equation (17).

The Logistic map is a non-linear chaotic discrete function that produces random sequences. In the proposed cryptosystem, the logistic map is used as a diffusion function to achieve the diffusion effect, by transferring the effect from one byte in the block to other bytes in the same block. This structure makes the proposed cryptosystem highly sensitive to the plaintext. The mathematical model of the discrete logistic map is:

$$X_{k+1} = \begin{cases} \left\lfloor \frac{X_k \times (2^N - X_k)}{2^{N-2}} \right\rfloor & \text{if} X_k \neq \left[3 \times 2^{N-2},\ 2^N\right] \\ 2^N - 1 & \text{if} X_k = \left[3 \times 2^{N-2},\ 2^N\right] \end{cases} \quad (19)$$

where X_{k+1} is the new value calculated from the previous one X_k. N is the number of bits representing the integer output of the discrete logistic map, which is equal to 32 bits. From the figure 5, we can write the encryption mathematical model as:

$$c_l(k_n) = LSB_8[y_l(k)] \quad (20)$$

$$y_l(k) = p_l(k) \oplus s_{l-1}(k) \oplus f(y_l(k-1)) \quad (21)$$

where $y_l(k)$ is a 32-bit variable, $p_l(k),\ s_{l-1}(k)$ are 8-bit variables and f is the logistic map. The following remarks should be considered:

1) During the encryption, equation (21) should be evaluated before equation (20), for each byte of a block and for all blocks.
2) The input of the logistic map for $k = 0$ is kd_m when $l = 0$ and it is $y_{l-1}(b_s - 1)$ for $l > 0$.
3) For $k > 0$ and for all l, the input of the logistic map is the result of equation (21) and not the previous output (see equation 19).

Note that: $k = i \times M + j$
$k_n = i_n \times M + j_n$
i_n and j_n are calculated using equation (17). The sequence $s_{l-1}(k)$ is given by the following equation:

$$s_{l-1}(k) = \begin{cases} iv(k) & \text{if } l = 0 \\ c_{l-1}(k) & \text{if } l > 0 \end{cases} \quad (22)$$

where $l = 0,\ 1,\ 2,\ ...b_n - 1$, $k = 0,\ 1,\ 2,\ ...b_s - 1$, $IV = \{iv(0),\ iv(1),\ iv(2),\ ...,\ iv(b_s - 1)\}$, b_s is block size in bytes
$b_n = \frac{\text{image size}}{\text{block size}} = \frac{L \times C \times P}{b_s}$, is the number of blocks.
with, L, C, and P are the number of lines, the number of columns, and the number of planes of the image respectively.

TABLE I: Average encryption/decryption times for different cryptosystems for Lena image with different sizes.

Cryptosystem	$256 \times 256 \times 3$	$512 \times 512 \times 3$	$1024 \times 1024 \times 3$
Proposed Cryptosystem	2.04/2.68	8.08/10.57	31.85/41.83
Zhang	7.5/7.5	30/30	120/120
Wang	7.79/8.39	31.16/33.54	124.64/134.16

TABLE II: Average Encryption throughput and average Number of cycles for to encrypt/decrypt one byte.

Cryptosystem	ET in MBps	Number of cycles per byte
Proposed Cryptosystem	92.975/70.879	29.87/40.57
Zhang	25/25	122.07/122.07
Wang	24.06/22.35	122.85/132.24

III. Time performance and security analysis

A. Time performance

The time performance is determined by evaluating the running speed that can be measured by: the average encryption/decryption times, the average encryption throughput, and the average needed number of cycles to encrypt one byte. The encryption throughput (ET) and the number of cycles, which are required to encrypt (or decrypt) one byte, are defined as:

$$ET = \frac{Image_{Size}(Byte)}{Encryption_{Time}(second)} \tag{23}$$

$$\text{Number of cycles per Byte} = \frac{\text{CPU Speed}_{(Hertz)}}{ET_{(Byte)}} \tag{24}$$

The last equation permits to compare the running speed of different cryptosystems working on different platforms. In table I, we give the average encryption/decryption times for Lena image at different sizes with different cryptosystems(for our cryptosystem the used block size is 1024 byte). In table II we give the average throughput and the average number of cycles. The proposed cryptosystem is at least 3 times faster that zhang and wang cryptosytems.

B. security analysis

usually A cryptanalyst tries to break the cipher without knowing the secret key, and this with several levels of difficulties based on the available resources. Chosen plain-text attack is the easiest one for the attacker: The attacker has access to the system without knowing the secret keys. Then, he has the possibility to choose a set of plain-text messages and to encrypt them. If a cryptosystem can resist to chosen plain-text attack, then it can resist to all other attacks such as:cipher text only, known plain-text and chosen cipher attacks. The chosen plain-text attack can be realised by the plain-text sensitivity attack or differential attacks introduced by Eli Biham and Adi Shamir [12]. Most researchers use two security parameters to measure the resistance of any chaos-based cryptosystem for plain-text sensitivity attacks. These parameters are: the Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI); they are given by the following equations, respectively:

$$NPCR = \frac{1}{L \times C \times P} \times \sum_{p=1}^{P} \sum_{i=1}^{L} \sum_{j=1}^{C} D(i,j,p) \times 100\% \tag{25}$$

where

$$D(i,j,p) = \begin{cases} 0, & \text{if} \quad C_1(i,j,p) = C_2(i,j,p) \\ 1, & \text{if} \quad C_1(i,j,p) \neq C_2(i,j,p) \end{cases} \tag{26}$$

$$\begin{aligned} \text{UACI} = & \frac{1}{L \times C \times P \times 255} \\ & \times \sum_{p=1}^{P} \sum_{i=1}^{L} \sum_{j=1}^{C} |C_1(i,j,p) - C_2(i,j,p)| \times 100\% \end{aligned} \tag{27}$$

In the previous equations, i, j and p are the row, column, and plane indexes of the image, respectively. L, C and P are, respectively, the length, width, and plane sizes of the image. The optimal NPCR value is 99.61%, and the optimal UACI value is 33.46% [13]. The previous metrics are necessary but not sufficient to ensure that the proposed cryptosystem is resistant against plain-text sensitivity attacks. Then, a new metric measurement, the Hamming Distance (HD) is used to quantify the avalanche effect. The Avalanche effect is achieved for any block cipher, when a small change (for example, flipping a single bit) in either the plain-text or the secret key, causes a drastic change in the cipher-text (e.g., half of the output bits are flipped), [14]. Therefore, this evaluation test is used to measure the resistance of any cryptosystem to the plain-text and the key sensitivity attacks. The HD is defined by:

$$HD(C_1, C_2) = \frac{1}{|Ib|} \sum_{K=1}^{|Ib|} (C_1(K) \oplus C_2(K)) \tag{28}$$

where $|Ib| = L \times C \times P \times 8$, is the size of the image in bits. The optimum HD value is 50%. A good block cipher should produce an HD close to 50% (probability of bit changes, which means that a one bit difference in plain-image will make every bit of the corresponding cipher-image change with a probability of a half. In table III, we show the obtained comparative results of NPCR, UACI and HD for Boat bmb image of size $256 \times 256 \times 3$.

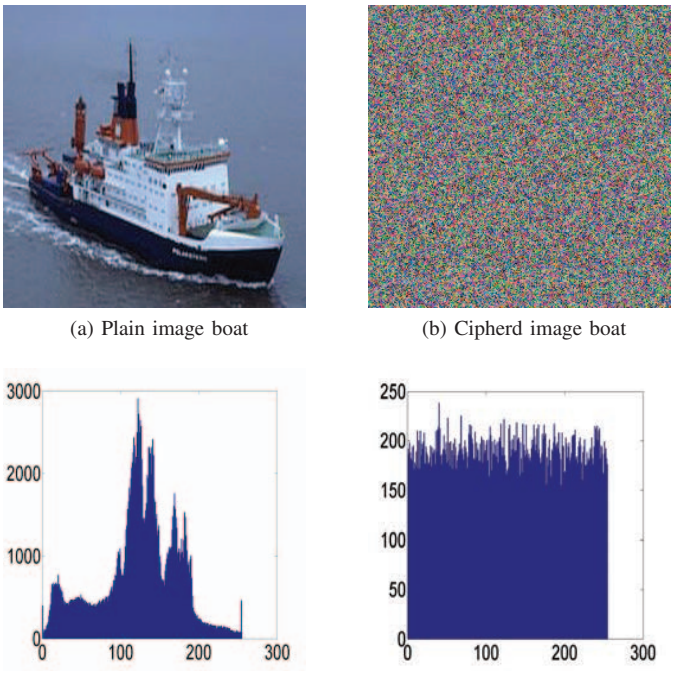
Statistical attack

To resist the statistical attacks, the histogram of the ciphered image must be uniform and the correlation coefficient should be close to zero . In figure 6a, 6b we show the plain image

TABLE III: The NPCR, UACI and HD

Cryptosystem	NPCR	UACI	HD
Proposed Cryptosystem	99.615	33.463	0.500030
Zhang	99.633	33.34	–
wang	99.96	33.33	–

for boat and its ciphered one, and in the figure 6c, 6d we give the histograms for the plain/cipher images. As we can see the histogram of the ciphered image seems to be uniform, to be sure we made the Chi square test with the following parameters: alpha=0.05, and number of classes equal to 256. The obtained experimental value is 255.12 which is less than the theoretical one that equal 293. This means that the histogram is uniform.



(a) Plain image boat (b) Cipherd image boat

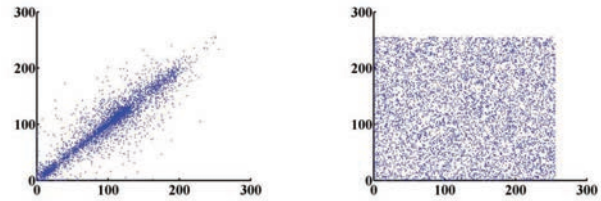
(c) Histogram for the plain image (d) Histograme for the cipher image

Fig. 6: Histogram of the boat plain image and its ciphered image

Figure 7 shows the correlation curves of the adjacent pixels in the horizontal direction for the plain image and its ciphered one. The values of their corresponding correlation coefficient are 0.96606 and 0.0085.
All these results imply a high security level of the proposed cryptosystem.

IV. Conclusion

In this paper, we presented an overview of chaos-based cryptosystems, including our cryptosystem, that are based on dependent confusion-diffusion processes, and we compared their speed performance and their robustness against cryptanalysis. The obtained results show that, our cryptosystem is faster than the others while having a very high security level.



(a) Plain image correlation of adjacent pixels (b) ciphered image correlation of adjacent pixels

Fig. 7: correlation of the boat plain image and its ciphered image

Acknowledgement

This work is supported by the European Celtic-Plus project 4KREPROSYS - 4K ultraHD TV wireless REmote PROduction SYStems, 2015 -

References

[1] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *International journal of bifurcation and chaos*, vol. 8, no. 06, pp. 1259–1284, 1998.

[2] S. Li, X. Mou, and Y. Cai, "Improving security of a chaotic encryption approach," *Physics Letters A*, vol. 290, no. 3, pp. 127–133, 2001.

[3] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3d chaotic cat maps," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[4] S. El Assad, M. Farajallah, and C. Vladeanu, "Chaos-based block ciphers: An overview," in *Communications (COMM), 2014 10th International Conference on*. IEEE, 2014, pp. 23–26.

[5] H. Yang, K.-W. Wong, X. Liao, W. Zhang, and P. Wei, "A fast image encryption and authentication scheme based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3507–3517, 2010.

[6] Y. Wang, K.-W. Wong, X. Liao, and G. Chen, "A new chaos-based fast image encryption algorithm," *Applied soft computing*, vol. 11, no. 1, pp. 514–522, 2011.

[7] W. Zhang, K.-w. Wong, H. Yu, and Z.-l. Zhu, "An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 8, pp. 2066–2080, 2013.

[8] M. Farajallah, S. El Assad, and O. Deforges, "Fast and secure chaos-based cryptosystem for images," *International Journal of Bifurcation and Chaos, accepted in 15-June-2015*, 2015.

[9] M. Farajallah, "Chaos based crypto and joint crypto-compression systems for images and videos," Ph.D. dissertation, University of Nantes, 2015.

[10] K. Kaneko, "Spatiotemporal chaos in one-and two-dimensional coupled map lattices," *Physica D: Nonlinear Phenomena*, vol. 37, no. 1, pp. 60–82, 1989.

[11] S. El Assad and H. Noura, "Generator of chaotic sequences and corresponding generating system," WO2011121218 A1 Extension to : Europe EP-2553567 A1, February 2013 ; China : CN-103124955 A, May 2013 ; Japan : JP-2013524271 A, June 2013 ; United states : US-20130170641, July 2013., 10 2011.

[12] E. Biham and A. Shamir, "Differential cryptanalysis of des-like cryptosystems," *Journal of CRYPTOLOGY*, vol. 4, no. 1, pp. 3–72, 1991.

[13] Y. Wu, J. P. Noonan, and S. Agaian, "Npcr and uaci randomness tests for image encryption," *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)*, pp. 31–38, 2011.

[14] P. P. Mar and K. M. Latt, "New analysis methods on strict avalanche criterion of s-boxes," *World Academy of Science, Engineering and Technology*, vol. 48, pp. 150–154, 2008.