A New Chaos-Based Cryptosystem for Secure Transmitted Images

Abir Awad

Laboratoire de cryptologie et de virologie operationnelles (C +V)^O, esiea, IUT, Laval

awad@esiea-ouest.fr

***Abstract*—This paper presents a novel and robust chaos-based cryptosystem for secure transmitted images and four others versions. In the proposed block encryption/decryption algorithms, an 2D chaotic map is used to shuffle the image pixel positions. Then, substitution (confusion) and permutation (diffusion) operations on every block, with multiple rounds, are combined using two perturbed chaotic PWLCM maps. The perturbing orbit technique improves the dynamical statistical properties of generated chaotic sequences. The obtained error propagation in various standard cipher block modes demonstrates that the proposed cryptosystem including OFB, or CTR modes, is suitable to transmit cipher data over a corrupted digital channel. Finally, to quantify the security level of the proposed cryptosystem, many standard tools are performed and experimental results show that the suggested cryptosystem has a high security level.**

***Index Terms*—Chaos-based cryptosystem, perturbed technique, error propagation, security.**

I. INTRODUCTION

SECURE, transmission of confidential digital images has become a common interest on both research and applications. With the desirable properties of pseudo-randomness, ergodiciy, high sensitivity to initial conditions and parameters, chaotic maps have demonstrated great potential for information especially image encryption. Since the 1990s, a large amount of work using digital chaotic systems to construct cryptosystems has been studied [1] - [3], and has attracted more and more attention in the last years [4] - [7]. In order to be used in every application, chaotic sequences must seem absolutely random and have good cryptographic properties. Many studies on chaotic maps are drawn [8] - [10]. In [11] and [12], we study and improve some existing techniques used to generate chaotic signals with desired statistical properties and verifying NIST statistical tests. Indeed, to obtain better dynamical statistical properties and to avoid the dynamical degradation caused by the digital chaotic system working in a 2^N finite state, a perturbation technique is used.

It is well known that images are different from texts in many aspects, such as high redundancy and correlation. The main obstacle in designing effective image encryption algorithms in that it is rather difficult to shuffle and diffuse such image data by traditional cryptographic means [13]-[16]. In most of the natural images, the value of any given pixel can be reasonably predicted from the values of its neighbors.

In order to solve this problem, many researchers have proposed schemes with combinational permutation techniques [17], [18] that divide the image into blocks then shuffle their positions before passing them to the bit manipulation stage. In fact, bit level permutations are particularly difficult for processors. Many researches tend to avoid it in the design of cryptography algorithm or use very simple permutations [19], [20]. But recently, a number of candidate instructions have been proposed to efficiently compute arbitrary bit permutations [6], [21]-[23]. In this paper, we propose a new approach for image encryption using a combination of different permutation techniques: Pixels and bit permutations.

Moreover, cryptographic modes for block ciphers have received much attention lately, partly due to an announcement of NIST [24]. No block cipher is ideally suited for all applications. This comes from differing tolerances of applications to properties of various cryptographic modes. As we search to meet the requirements of the secure image transfer, so we examine the problem of error propagation in various cipher block modes.

The paper is organized as follows: Section 2 describes the proposed algorithm; Section 3 introduces the perturbed chaotic map used; Section 4 explains the S-box transformations used in the algorithm; Section 5 presents the cipher block modes used for the encryption and compares the theoretical propagation error induced by each mode. The simulation results and security analysis are given in section 6. And finally, we summarize our conclusions in section 7.

II. ENCRYPTION ALGORITHM

In this section, we present the developed Algorithm called CBCSTI for Image Encryption that we implemented with Matlab. Let *I* be an *MxN* image with b-byte pixel values, where a pixel value is denoted by *I(i)*, $0 \leq i < MxNxb$. A block cipher is an encryption scheme which breaks up the plaintext messages into blocks of fixed length and encrypts one block at a time.

The algorithm characteristics and steps are:

(1) The key size is 128-bits.

(2) The chaotic maps used is a 2 D chaotic map and two perturbed piecewise linear chaotic maps (P-PWLCM). The perturbed chaotic values are generated each r ietrations.

(3) The permutation box (P-box) adding diffusion to the system includes two steps:
Firstly, the positions of the pixels of the original image are shuffled by 2D chaotic map. Then the pixel values are permuted by bit permutation method.

(4) A more complex substitution box (S-box) is applied.

(5) Multiple rounds for encryption and decryption processes are used.

The encryption algorithm transforms an image *I* using a 2D chaotic map and an SP-network generated by a one dimensional chaotic map and a 128-bit secret key. The algorithm performs r rounds of an SP-network on each pixel. Fig. 1 illustrates the flow chart of the algorithm with OFB mode. This algorithm is implemented also with the other standard modes: ECB, CBC, CFB, CTR (see section VI).

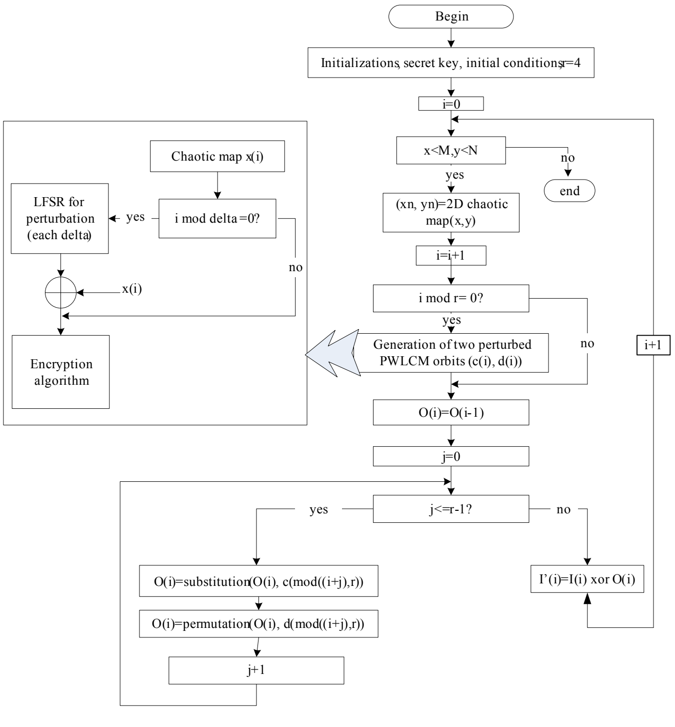

Begin
Initializations, secret key, initial conditions, r=4
i=0
x<M,y<N
no
end
yes
(xn, yn)=2D chaotic map(x,y)
i=i+1
i mod r= 0?
yes
no
Generation of two perturbed PWLCM orbits (c(i), d(i))
i+1
Chaotic map x(i)
i mod delta =0?
yes
LFSR for perturbation (each delta)
no
x(i)
Encryption algorithm
O(i)=O(i-1)
j=0
yes
j<=r-1?
no
O(i)=substitution(O(i), c(mod((i+j),r))
O(i)=permutation(O(i), d(mod((i+j),r))
j+1
I'(i)=I(i) xor O(i)

Fig. 1. The proposed algorithm with the OFB mode

Many variants of the algorithm are drawn. They differ depending on the choice of permutation methods 2-D or 1-D used:
CBCSTI-A: use the standard map to do the pixel position permutation and Socek method as a bit permutation method.
CBCSTI-B: use Standard and CROSS methods.
CBCSTI-C : use Arnold and Socek methods.
CBCSTI-D : use Arnold and CROSS methods.
CBCSTI-E : use Socek method without a 2D chaotic map.

In the next section, we explain the perturbed PWLCM map used in the algorithm. Then, we discuss the SP box adopted in the proposed algorithm.

III. PERTURBED PWLCM MAP

A piecewise linear chaotic map (PWLCM) is a map composed of multiple linear segments.

$$x(n) = F[x(n-1)] = \begin{cases} x(n-1) \times \dfrac{1}{p} & if \;\; 0 \le x(n-1) < p \\ [x(n-1) - p] \times \dfrac{1}{0.5-p} & if \;\; p \le x(n-1) < 0.5 \\ F[1 - x(n-1)] & if \;\; 0.5 \le x(n-1) < 1 \end{cases} \tag{1}$$

where the positive control parameter $p \in (0; 0.5)$ and $x(i) \in (0; 1)$. Since digital chaotic iterations are constrained in a discrete space with 2^N elements, it is obvious that every chaotic orbit will eventually be periodic and will finally go to a cycle with a limited length not greater than 2^N [25], [26]. Generally, each digital chaotic orbit includes two connected parts: $x_1, x_2, ..., x_{l,}$ and $x_l, x_{l+1}, ..., x_{l+n,}$, which are respectively called "transient branch" and "cycle". Accordingly, *l* and *n+1* are respectively called "transient length" and "cycle period", and *l+n* is called "orbit length".

To improve the dynamical statistical properties of generated chaotic sequences, a perturbation-based algorithm is used. The cycle length is expanded and consequently good statistical properties are reached. Many perturbation techniques are proposed. For example, Socek [6] uses a perturbation-based algorithm. The orbits are perturbed by the encrypted blocks. Socek algorithm is very secure but a bit error transmission causes a random number of erroneous bits in the decrypted image. In this paper, we use another perturbation technique using maximal length LFSR, which is a suitable candidate for perturbing the signal generator [25], [27].

Here, for computing precision *N*, each *x* can be described as:

$$x(n) = 0.x_1(n)x_2(n)...x_i(n)...x_N(n) \qquad x_i(n) \in \{0,1\} \quad i = 1,2,...,N \tag{2}$$

The perturbing bit sequence can be generated every *n* clock as follows:

$$Q_{k-1}^{+}(n) = Q_k(n) = g_0Q_0(n) \oplus g_1Q_1(n) \oplus ... \oplus g_{k-1}Q_{k-1}(n) \quad with\, n = 0,1,2,... \tag{3}$$

Where $\oplus$ represents 'exclusive or', $g = [g_0\, g_1 ... g_{k-1}]$ is the tap sequence of the primitive polynomial generator, and $Q_0\, Q_1 ... Q_{k-1}$ are the initial register values of which at least one is non zero.

The perturbation begins at *n= 0*, and the next ones occur periodically every Δiterations (Δ is a positive integer), with *n=* $l \times \Delta$, *l=1,2,...,* The perturbed sequence is given by the equation (4):

$$x_i(n) = \begin{cases} F[x_i(n-1)] & 1 \le i \le N-k \\ F[x_i(n-1)] \oplus Q_{N-i}(n) & N-k+1 \le i \le N \end{cases} \tag{4}$$

Where $F[x_i(n)]$ represents the *i*th bit of $F[x(n)]$.
The perturbation is applied on the last *k* bits of $F[x(n)]$.
When $n \ne l \times \Delta$, no perturbation occurs, so $x(n) = F[x(n-1)]$.
The lower boundary of the system cycle length is

$$T_{\min} = \Delta \times (2^k - 1) \tag{5}$$

IV. SP BOX TRANSFORMATION

In the proposed algorithm, we used two perturbed PWLCM chaotic maps. The chaotic orbits are real values. Then, we transform it to unsigned integer on 32 bits using the following formula (eq. 6):

$$x(n) = round(x(n) \times 2^{32}) \tag{6}$$

The first binary suite is used to control the substitution and the second one is used to control the permutation. The SP box (the substitution and permutation operations) are performed r (r=4) iterations. In this section, we explain the SP box used in the CBCSTI algorithms and how the control is done.

A. Substitution box

A complex substitution box is applied (eq. 7).

$$\mathrm{Sigma}_{\mathrm{r}}(\mathrm{u,v}) = \begin{cases} u \ \oplus \ v & \textit{if } r \textit{ is even} \\ (u+v) \bmod 256 & \textit{if } r \textit{ is odd} \end{cases} \tag{7}$$

where *u* and *v* are two bytes and *r* is the round value.

In fact, *u* and *v* represent respectively the chaotic value and the plaintext block (see figure 1).

O(i)=substitution(O(i), c(mod((i+j),r))

c(mod((i+j),r)) is the chaotic value that we used to control the substitution technique. We said that we use a binary suite to control the substitution technique. This suite in decomposed on four bytes and then transformed on four integers c(1), c(2), c(3) and c(4). These values are used to r iterations of the substitution of O(i) block. O(i) can be the plaintext block or the output key that we encrypted. It depends of the adopted operation mode ECB/CBC or OFB/CTR.

B. Permutation

In order to disturb the high correlation among adjacent pixels, we propose a scheme that includes two phases: firstly, the pixel positions are permuted by 2D chaotic map. Then the gray values of the permuted image are encrypted by a bit permutation method.

2D Permutation

The 2D chaotic map shuffles the pixel positions of the plain image and then disturbs the high correlation among the pixels. Without loss of generality, the size of the original grayscale image *I* is assumed NxN.

Permutation by Arnold cat map

The coordinates of the pixel positions are $S = \{(x,y) | x,y = 0,1,2,...,N-1\}$. Arnold cat map [18] is described as follows (eq. 8).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} (\bmod N) = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (\bmod N) \tag{8}$$

Where *p* and *q* are positive integers, *det(A)=1*. The *(x, y)* and *(x', y')* are the original and the new positions, respectively. After several iterations, the original image can be permuted completely. The parameters *p*, *q* and the iteration number *M* can be chosen as the secret keys.

Permutation by Standard map

The Standard map is described with the following formula (eq. 9):

$$\begin{cases} x' = (x+y) \bmod N \\ y' = (y + k\sin(\frac{xN}{2\pi})) \bmod N \end{cases} \tag{9}$$

Where k is a positive constant, *(x, y)* is the original pixel position and *(x', y')* is the newest one.

Bit permutation method

The permutation is made on the bits of each block formed of a byte. In other words, we use a permutation of degree 8 to add diffusion to the system. Actually, the fastest way to achieve this is by using a table look up approach. This approach is fast but the memory requirements are considerably high. A number of permutation methods have been proposed [6], [18]-[20]. Among these, CROSS and Socek methods are the most attractive. They have good cryptographic properties.

As we explained for the substitution technique. The chaotic binary suite is decomposed on two parts of 16 bits, then transformed on decimal values d(1), d(2). These values are limited by 8! (modulo 8!) that we permute 8 bits. In each iteration (j=1…4), we chose a value to perform the permutation of the block O(i).

O(i)=permutation(O(i), d(mod((i+j),r))

In following, we present the Cross and Socek methods used in the proposed algorithm.

Cross permutation

It is controlled using the perturbed chaos in our algorithm. The control register *R2* is filled by the chaotic binary suite. The CROSS instruction is defined as follows:

R3=CROSS(m1, m2, R1, R2)

R1 is the source register which contains the bits to be permuted, *R2* is the configuration register and *R3* is the destination register for the permuted bits. One CROSS instruction performs two basic operations on the source according to the contents of the configuration register and the values of *m1* and *m2*. Fig. 2 shows how the CROSS instruction works on 8-bit systems.

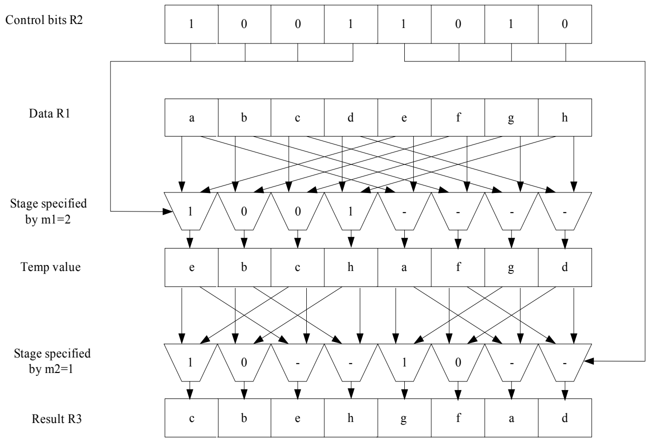


Fig.2. The CROSS instruction on 8-bit systems

Socek permutation

The permutation method proposed by Socek is to permute the indices of bits of each pixel using the output of a chaotic map. Then the bits are rearranged according to the new array indices (see Fig. 4). Fig. 5 presents the algorithm of Socek method.

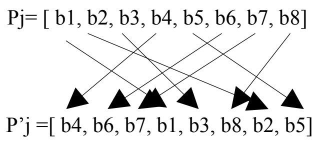

Pj= [b1, b2, b3, b4, b5, b6, b7, b8]
P'j =[b4, b6, b7, b1, b3, b8, b2, b5]

Fig. 3. Socek method on 8 bits

V. DECRYPTION PROCESS

The decryption algorithm depends on the cipher mode used. For the modes OFB, CFB and CTR; the decryption algorithm is the same as that of the encryption. But for the CBC mode, it differs slightly from the encryption algorithm. To decrypt an encrypted image, we need to perform the inverse transformations. The inverse substitution will be as follows (eq. 10):

$$\mathrm{Sigma}_{\mathrm{r}}^{-1}(\mathrm{u,v}) = \begin{cases} u \ \oplus \ v & if \ r \ is \ even \\ (u-v) \bmod 256 & if \ r \ is \ odd \end{cases} \tag{10}$$

The inverse CROSS instruction is the same as that used for the encryption process but the contents of the configuration register m1 and m2 are exchanged. The inverse of Socek method, the bits are rearranged according to the array indices *(8-p(i))* instead of *p(i)* used in the encryption process. Therefore, we need the reverse the order of the substitution and bit permutation method and we use the inverse ones to decrypt the image.

VI. CRYPTOGRAPHY MODES AND ERROR PROPAGATION

A cryptographic mode usually combines the basic cipher, some sort of feedback, and some simple operations. Some applications need to parallelize encryption or decryption, while others need to be able to preprocess as much as possible.

A bit error is the substitution of a ‘0’ bit for a ‘1’ bit, or vice versa. These errors are generated by the transmission channel as a consequence of interference and noise.

The error propagation phenomenon implies that errors in the encrypted text produce errors in the decrypted plaintext. So, it is important that the decrypting process be able to recover from bit errors in the ciphertext.

In this section, we examine the problem of error propagation in various cipher block modes of operation, such as: Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter mode (CTR) [17]. The dependence between input and output error probability of the modes is presented. The results obtained can be used to choose the block cipher and its mode to generate a suitable cryptogram for transmission over a noisy channel.

The CFB is a special mode witch which segments are operated. The segment is an *s* bit block, where $1 \le s \le b$. The j-th plaintext and encrypted segment are denoted by $I_j^{\#}$ and $I_j^{'\#}$ respectively.

The effect of the error bit $I_{i,j}^{'}$ or $I_{i,j}^{'\#}$ in the block $I_j^{'} = \left(I_{1,j}^{'}, I_{2,j}^{'}, ..., I_{b,j}^{'}\right)$ or $I_j^{'\#} = \left(I_{1,j}^{'\#}, I_{2,j}^{'\#}, ..., I_{b,j}^{'\#}\right)$ on the appearance of errors in the plaintext for individual modes is summarized in the following table (table I).

TABLE I
THE EFFECT OF BIT ERRORS FOR CIPHER BLOCK MODES

Mode	Effect of bit errors in $I_j^{'}$
ECB	RBE in I_j
CBC	RBE in I_j SBE in I_{j+1}
CFB	SBE in $I_j^{\#}$ RBE in $I_{j+1}^{\#}$, $I_{j+2}^{\#}, \ldots, I_{j+b/s}^{\#}$
OFB	SBE in I_j
CTR	SBE in I_j

In the table, SBE (Specific Bit Errors) means that an individual error bit $I_{i,j}^{'}$ or $I_{i,j}^{'\#}$ produces in the appropriate decrypted block an individual error bit $I_{i,j}$ or $I_{i,j}^{\#}$. It occurs in the same bit positions as the bit errors in the encrypted image. RBE (random bit errors) means that an individual error bit $I_{i,j}$ or $I_{i,j}^{\#}$ affects randomly all bits in the decrypted block $I_{i,j}$ or in the segments $I_{j+1}^{\#}$, $I_{j+2}^{\#}, \ldots, I_{j+b/s}^{\#}$.

In the CBC mode for example, all bit positions that contain bit errors in a cipher text block will produce an RBE in the same decrypted block and an SBE in the second one; the other bit positions are not affected. For the OFB and CTR modes, bit errors within a ciphertext block do not affect the decryption of any other block.

Denote P_e the bit error probability in the decrypted image and p_e the bit error probability in the encrypted image. As we said, in the case of OFB and CTR modes, only the SBE type of error propagation can occur. For these modes, each error bit $I'_{i,j}$ of the encrypted image causes only one incorrect bit $I_{i,j}$ of the original image and thus the output error probability is equal to the input one:

$$P_e = p_e \tag{11}$$

In the case of other modes (ECB, CBC, CFB), the RBE type of error appears. To present the dependence between the two probability p_e and P_e, we will give some definitions.

The probability $P(x)$ that there are x error bits out of b received bits, is given by the formula (eq. 12):

$$P(x) = \begin{bmatrix} b \\ x \end{bmatrix} . p_e^x .(1-p_e)^{b-x}, x = 01,2,..b \tag{12}$$

Then, it holds for the probability P_0, that b bits are correct:

$$P_0 = (1-p_e)^b \tag{13}$$

and for the probability Q_0, that at least one bit is incorrect:

$$Q_0 = 1 - P_0 = 1-(1-p_e)^b \tag{14}$$

We call P_0 the correct block probability and Q_0 the incorrect block probability.

The probability P_h that the output bit changes its value as a consequence of modifying the input block is called the bit inversion probability. We assume that P_h =1/2.

In the ECB mode, the output error probability P_e is equal to:

$$P_e = P_h Q_0 = \frac{1}{2}.[1-(1-p_e)^b] \tag{15}$$

The resulting output error probability P_e for the CBC mode is given by the equation:

$$P_e = p_e.(1-p_e)^b + \frac{1}{2}.[1-(1-p_e)^b] \tag{16}$$

In fact, $I_j = U_j \oplus I'_{j-1}$ where $U_j = D_k(I'_j)$.

The bit I_{ij} is incorrect in the following cases:

a) the bit $I'_{I,(j-1)}$ is incorrect and the block I'_j is correct,

b) the bit $I'_{I,(j-1)}$ is incorrect, the block I'_j is incorrect and the bit ui,j is not inverted,

c) the bit $I'_{I,(j-1)}$ is correct but the block I_j is incorrect and the bit $u_{i,j}$ is inverted.

The probability of the error bit $I'_{I,(j-1)}$ is equal to p_e and the probability of the correct block I'_j is P_0. Thus, the situation

a) occurs with the probability Pa) = $p_e.P_0$. The probability of the incorrect block I'_j is equal to Q_0 and the probability that the bit $u_{i,j}$ is not inverted, amounts to (1-P_h). Then, the probability of the situation b) is equal to the quantity Pb) = $p_e.Q_0.(1\text{-}P_h)$. The probability of the correct bit $I'_{i,(j-1)}$ is equal to the value (1-p_e), the probability of the incorrect block I'_j is equal to Q_0 and the probability that the bit $u_{i,j}$ is inverted, amounts to P_h. Then, the probability of the situation c) is equal to the quantity Pc) = (1-p_e).P_h.Q_0. Thus the resultant output error probability Pe for the CBC mode

is given by this equation:

$$P_e = P_{a)} + P_{b)} + P_{c)} = P_{a)} = p_e.P_0 + \frac{1}{2}Q_0$$
$$= p_e.(1-p_e)^b + \frac{1}{2}.\left[1-(1-p_e)^b\right] \qquad (17)$$

The equations in the case of (CBC) and (CFB) are the same. It follows that the output error probability *Pe* is the same for both of the modes. Thus the CBC and CFB modes are equivalent from the viewpoint of error propagation. From equality (CFB), it is also evident that the output error probability of the CFB mode does not depend on the length *s* of segments. Table II presents the dependence between the two probabilities p_e and P_e for different operation modes [28].

TABLE II
THE DEPENDENCE BETWEEN THE BIT ERROR PROBABILITIES IN THE ENCRYPTED AND DECRYPTED IMAGES

Mode	Effect of bit errors in I'_j
ECB	$P_e = \frac{1}{2}.\left[1-(1-p_e)^b\right]$
CBC, CFB	$P_e = p_e.(1-p_e)^b + \frac{1}{2}.\left[1-(1-p_e)^b\right]$
OFB,CTR	$P_e = p_e$

The results obtained for ECKBA algorithm not respect the expected one. In fact, Socek in his algorithm, use a perturbation technique of PWLCM chaotic map using the encrypted data. Then, if a transmission error occurs in the cipher image, we obtain random errors in the decrypted image. However, in our algorithm, we perturb the chaotic value with a LFSR. The encrypted blocks are independent. For that, we avoid the propagation error in the decrypted image.

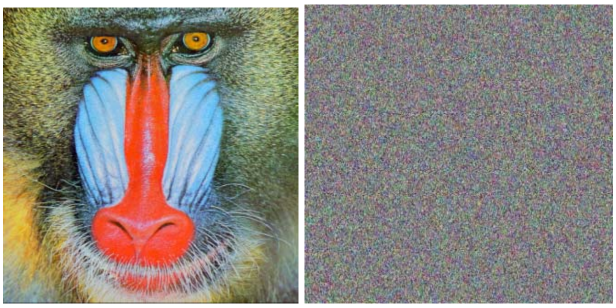
VII. SIMULATION RESULTS AND SECURITY ANALYSIS

Some experimental results are given in this section to demonstrate the efficiency of our scheme. We implemented in Matlab the different algorithms . The computer used is Pentium(R) D CPU 3Ghz, 2.99 Ghz, 2 Go RAM. The plain image used is 'MANDRILL.BMP' with the size 512x512x3 (Fig. 4(a)). The encryption time of different algorithms with different modes are shown in table IV.

TABLE IV
THE ENCRYPTION TIME FOR DIFFERENT ALGORITHMS WITH FOUR OPERATION MODES

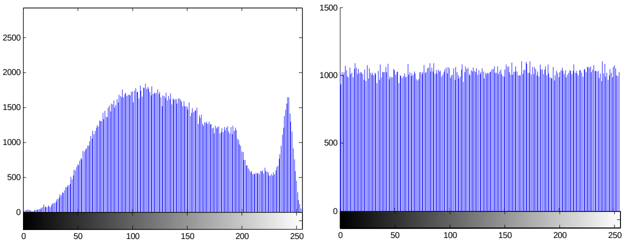
	CBC	OFB	CTR	CFB
CBCSTI-A	525.2344	544.9688	663.6719	527.8281
CBCSTI-B		921.1563		
CBCSTI-C	527.3438	503.0313	717.7344	525.1875
CBCSTI-D		926.1094		
CBCSTI-E	537.9531	496.2813	643.6406	507.2188
AES		1741.9		

The cipher image with CBCSTI-A algorithm is shown in Fig. 4(b). We plot their histograms of RGB color and we present that of Red color in Fig. 5. As we can see, the histogram of the ciphered image is fairly uniform and is significantly different from that of the original image.



(a) (b)

Fig.4. (a) 'MANDRILL.BMP' image and (b) his encrypted image



(a) (b)

Fig.5. Histograms of red component of (a) 'MANDRILL.BMP' image, (b) encrypted image.

Common measures like correlation, NPCR (Number of pixels change rate) and UACI (Unified Average Changing Intensity) are used to test the difference between the original image P1 and the encrypted one C1.

We calculate the correlation coefficient *r* of original and encrypted image by using the following formulas (18), (19) and (20), (21):

$$E(x) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N} P_1(i,j) \tag{18}$$

$$D(P_1) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N} [P_1(i,j) - E(P_1(i,j))]^2 \tag{19}$$

$$\operatorname{cov}(P_1,C_1) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N} [P_1(i,j) - E(P_1(i,j))][C_1(i,j) - E(C_1(i,j))] \tag{20}$$

$$r_{P_1C_1} = \frac{\operatorname{cov}(P_1,C_1)}{\sqrt{D(P_1)}\sqrt{D(C_1)}} \tag{21}$$

where $P_1(i,j)$ and $C_1(i,j)$ are gray values of the original pixel and the encrypted one.

NPCR stands for the number of pixel change rate. Then, if D is a matrix with the same size as images P_1 and C_1, *D (i,j)* is determined as follows (22):

$$D(i,j) = \begin{cases} 1 & if\ P_1(i,j) \neq C_1(i,j) \\ 0 & else \end{cases} \tag{22}$$

NPCR is defined by the following formula (23):

$$NPCR = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} D(i,j)}{M \times N} \times 100 \tag{23}$$

where, M and N are the width and height of P_1 and C_1.

The UACI measures the average intensity of differences between the plain image and the ciphered image.

UACI is defined by the following formula (24):

$$UACI = \frac{1}{MxN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \frac{|P_1(i,j) - C_1(i,j)|}{255} x100 \tag{24}$$

In table V, we summarize the correlation, NPCR and UACI obtained between the original image and the encrypted one.

TABLE V
The CORRELATION, NPCR AND UACI BETWEEN THE ORIGINAL IMAGE AND THE ENCRYPTED ONES

	Operation mode			
	CBC	CFB	OFB	CTR
(R)	0.0012	-0.0028	-0.0032	-0.0022
correlation (V)	0.0007	0.0029	0.0022	-0.0032
(B)	0.0029	0.0034	-0.0003	0.0009
(R)	99.5956	99.6147	99.6147	99.6265
NPCR(%) (V)	99.5804	99.5960	99.6071	99.6136
(B)	99.6117	99.6128	99.6090	99.6021
(R)	29.9514	30.0047	29.9928	30.0103
UACI (%) (V)	28.5867	28.5662	28.5594	28.6119
(B)	31.1631	31.1708	31.2590	31.2632

A. key sensitivity

An encryption scheme has to be key-sensitive, meaning that a tiny change in the key will cause a significant change in the output. In order to demonstrate the key sensitivity, the following experiments have been done with a slightly different key.

key sensitivity at the emission

Fig.4 (b) shows the encrypted image with the following key:

alpha= 0.35899926, beta=0.25899926, x0=0.7239 and y0= 0.5672. We encrypt the same image using the little changed key as follows: alpha= 0.3589992600001. Then, Fig. 10 shows the difference between the two ciphered images.

As we can see even where the control parameter of the first orbit is changed a little (10^{-8}), the encrypted image is absolutely different from the first one.

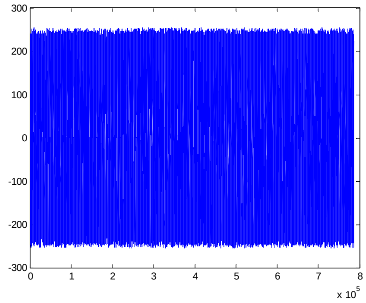


Fig.6. The difference between two ciphered images with a very small changed key.

In table VI, we summarize the NPCR and UACI for different algorithms: These results show that the proposed algorithm can survive differential attacks.

TABLE VI
The NPCR BETWEEN TWO CIPHER IMAGES ENCRYPTED BY CBCSTI WITH DIFFERENT OPERATION MODES

	CBC	CFB	OFB	CTR
(R)	53.1963	53.1967	53.1967	53.1967
NPCR (V)	72.4106	72.4106	99.9996	99.1566
(B)	87.7670	87.7663	100	99.1745
(R)	17.9681	17.8900	17.8439	17.8273
UACI (V)	24.2914	24.3108	33.5499	33.4702
(B)	29.5437	29.5047	33.5449	33.5063

key sensitivity at the reception

The goal of this test is to show the inability to decipher using the key 2 (with alpha= 0.3589992600001), an image encrypted by the key 1 (see Fig. 4(b)). The result of decryption shown in Figure 7.

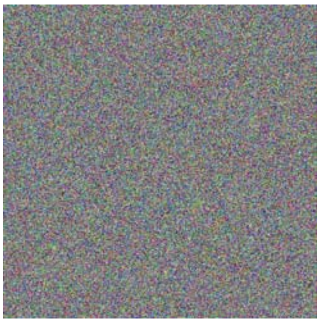
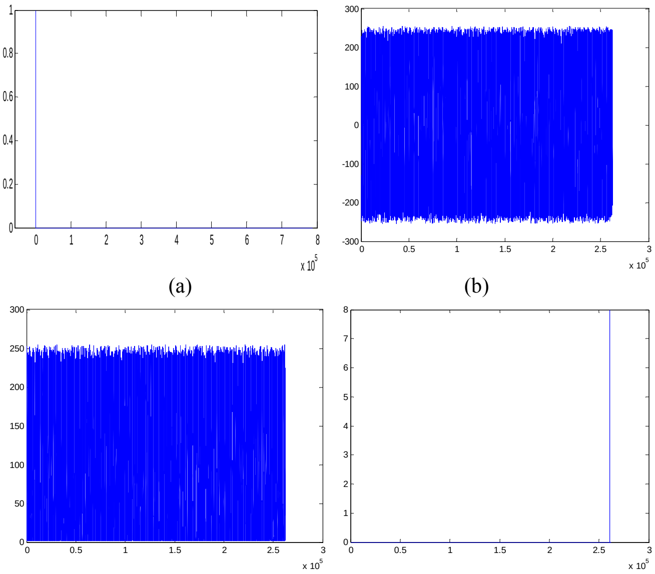


Fig.7. The decrypted image with a very small changed key.

B. Plaintext sensibility

This test measures the difference between two images with a single different bit.



(a) (b)

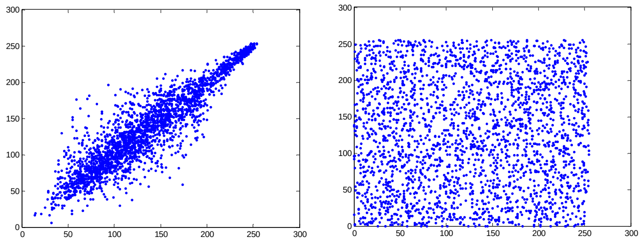
(c) (d)

Figure 8: Difference between two successive: -a) original images of Mandrill with a different bit,-b) encrypted images with CBC mode, -c) encrypted images with CFB mode,-d) encrypted images with OFB and CTR modes.

C. Correlation of two adjacent pixels

Statistical analysis on large amounts of images shows that averagely adjacent 8 to 16 pixels are correlative. To test the correlation between horizontally, vertically and diagonally adjacent pixels from the image, we calculate the correlation coefficient of a sequence of adjacent pixels by using the formulas (18), (19) and (20), (21).

Fig.9 shows the correlation distributions of two horizontally adjacent pixels in the original and the ciphered image



(a) (b)

Fig.9. The correlation distributions of two horizontally adjacent pixels in the original image and in the ciphered image

In table VII, the correlation coefficients are shown for the original. Table VIII gives the coefficients for the encrypted images by CBCSTI-A and CBCSTI-A with different operation mode. Table IX present the coefficients for the encrypted images by the other algorithms with CTR mode.

TABLE VII
CORRELATION COEFFICIENTS OF ADJACENT PIXELS.IN THE ORIGINAL IMAGE

Corrélation horizontale	0.9203
Corrélation verticale	0.8631
Corrélation diagonale	0.8494

TABLE VIII
CORRELATION COEFFICIENTS OF ADJACENT PIXELS FOR CBCSTI-A AND E.

Algorithm	Correlation	Mode opératoire			
		CBC	CFB	OFB	CTR
CBCSTI-A	horizontal	-0.0149	0.0149	-0.0092	0.0168
	vertical	-0.0193	0.0122	0.0096	-0.0171
	diagonal	0.0381	-0.0226	-0.0083	0.0026
CBCSTI-E	horizontal	-0.0119	-0.0351	-0.0269	0.0143
	vertical	0.0022	-0.0123	0.0030	0.0241
	diagonal	-0.0034	-0.0105	-0.0095	-0.0134

TABLE IX
CORRELATION COEFFICIENTS OF ADJACENT PIXELS FOR CBCSTI- B,C,D AND AES.

Correlation	CBCSTI-B	CBCSTI-C	CBCSTI-D	AES
horizontal	0.0282	-0.0064	-0.0123	-0.0188
vertical	0.0289	-0.0126	0.0130	-0.0024
diagonal	-0.0083	0.0374	-0.0046	0.0019

D. Information entropy analysis

Entropy is a statistical measure of randomness that can be used to characterize the texture of an image. It is well known that the entropy *H(m)* of a message source m can be calculated as [29]:

$$H(m) = \sum_{i=0}^{2^N - 1} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{25}$$

Where $p(m_i)$ represents the probability of message m_i.

When an image is encrypted, its entropy should ideally be 8. If it is less than this value, there exists a certain degree of predictability which threatens its security.

In table X, we list the entropy of the images encrypted by three algorithms. The values obtained are very close to the theoretical value 8. This means that information leakage in the encryption process is negligible and the encryption system is secure against the entropy attack.

TABLE X
ENTROPY VALUE FOR THE IMAGES ENCRYPTED WITH THREE DIFFERENT ALGORITHMS

Algorithm	Original image	CBCSTI-A	CBCSTI-C	CBCSTI-D
entropy	7.7624	7.9999	7.9997	7.9998

E. *NIST Statistical Tests*

Among the numerous standard tests for pseudo-randomness, a convincing way to show the randomness of the produced sequences is to confront them to the NIST (National Institute of Standards and Technology) Statistical Tests. The NIST statistical test suite [17] is a statistical package consisting of 188 tests that were developed to test the randomness of arbitrary long binary sequences produced by either hardware or software based cryptographic random or pseudorandom number generators. These tests focus on a variety of different types of non-randomness that could exist in a sequence.

To verify our results, we use the above test suite to test the randomness of 100 encrypted images of length 2097152 bits. In table XI, we show the results for a number of tests knowing that the sequence passed all the other tests. Note that the 100 encrypted images were generated with randomly selected secret keys. For each test, the default significance level α=0.01 was used, at the same time a set of P-values, which is corresponding to the set of images, is produced. Each image is called *success* if the corresponding P-value satisfies the condition P-value $\geq \alpha$, and is called *failure* otherwise and noted by a star.

TABLE XI
NIST STATISTICAL TEST FOR 100 ENCRYPTED IMAGES BY THREE PROPOSED ALGORITHMS CBCSTI-A, D AND E

	CBCSTI-E	CBCSTI-D	CBCSTI-A
Frequency Monobit Test	93*	95*	97
Block Frequency Test	99	100	100
Cumulative Sums Test	94*	95*	98
Random Excursion Test	95*	97	100
Random Excursion Variant Test	95*	98	98
Runs Test	97	99	98
Longest Runs Test	97	99	98
Rank Test	100	98	100
Discrete Fourier Transform	99	98	100
Serial	99	99	100
Non Overlapping Template Matching Test	98	98	100
Overlapping Template Matching Test	99	98	99
Approximate Entropy Test	98	99	100
Maurer’s « Universal Statistical » Test	99	98	100
Linear Complexity Test	98	99	99

VIII. CONCLUSION

In this paper, a new chaos-based cryptosystem is proposed. Our cryptosystem has some similarity with the Socek one, but attains a higher security level, and produces cryptograms in OFB and CTR modes, suitable to be transmitted on insecure and noisy channels. Indeed, in the new encryption/decryption algorithms, the key space is larger and the multi-rounds S-P network operations on each pixel are more complex than some existing algorithms. Furthermore, the introduction of the perturbation technique has expanded the length of the chaotic orbit cycle and then enhanced the dynamical statistical properties of the generated chaotic sequences. The obtained results: uniformity, key sensitivity, correlation, entropy, NIST statistical tests, prove the robustness and the high security level of the proposed cryptosystem.

REFERENCES

[1] A. Riaz, M. Ali, “Chaotic Communications, their applications and advantages over traditional methods of communication,” *IEEE*, *Communication Systems, Networks and Digital Signal Processing*, pp. 21-24, July 2008.

[2] G. Millérioux, J. M. Amigo, J. Daafouz, “A connection between chaotic and conventional cryptography,” *IEEE Trans. Circuits and Systems*, vol. 55, no. 6, pp. 1695-1703, Jul. 2008.

[3] L. Kocarev, “Chaos based cryptography: a brief overview,” *IEEE Trans. Circuits and Systems Magazine*, vol. 1, no. 3, pp. 6-21, 2001.

[4] T. Yang, C. W. Wu, L. O. Chua, “Cryptography Based on Chaotic Systems,” *IEEE Trans. Circuits and Systems*, vol. 44, no. 5, pp. 469–472, Feb. 1997.

[5] G. Jakimoski, L. Kocarev, “Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps,” *IEEE Trans. Circuits and Systems*, vol. 48, no. 2, pp. 163–169, Feb. 2001.

[6] D. Socek, S. Li, S. S. Magliveras, B. Furht, “Enhanced 1-D Chaotic Key Based Algorithm for Image Encryption,” *IEEE, Security and Privacy for Emerging Areas in Communications Networks,* 2005.

[7] G. Alvarez, S. Li, “Some Basic Cryptographic Requirements for Chaos Based Cryptosystems,” *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129-2151, 2006.

[8] S. El Assad, C.Vladeanu, “Digital chaotic codec for DS-CDMA Communication Systems,” Lebanese Science Journal, vol. 7, No. 2, 2006.

[9] L. Kocarev, J. Szczepanski, J. M. Amigo, I. Tomovski, “Discrete Chaos —I: Theory,” *IEEE Trans. Circuits and Systems Magazine*, vol. 53, no. 6, pp. 1300-1309, June 2006.

[10] S. Behnia, A. Akshani, S. Ahadpour, H. Mahmodi, A. Akhavan, “A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps,” *Physics Letters* A, pp. 391-396, 2007.

[11] A. Awad, S. E. Assad, Q. Wang, C. Vlădeanu, B. Bakhache, “Comparative Study of 1-D Chaotic Generators for Digital Data Encryption*,” IAENG International Journal of Computer Science*, vol. 35, no. 4, 2008.

[12] A. Awad, S. E. Assad, D. Carragata, “A Robust Cryptosystem Based Chaos for Secure Data,” *IEEE, Image/Video Communications over fixed and mobile networks*, Bilbao Spain, 2008.

[13] H. Xiao, S. Qiu, C. Deng, “A Composite Image Encryption Scheme Using AES and Chaotic Series,” First International Symposium on Data, Privacy and E-Commerce, pp. 277279 – 277279, 2007.

[14] Y. Wang, K. W. Wong, X. Liao, T. Xiang , G. Chen,“A chaos-based image encryption algorithm with variable control parameters Chaos,” Elsevier, Chaos, Solitons and Fractals 41 (2009) 1773–1783, 2008.

[15] Meghdad Ashtiyani, Parmida Moradi Birgani, Hesam M. Hosseini, “Chaos-Based Medical Image Encryption Using Symmetric Cryptography ,” *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on***,** pp. 1-5, Damascus,2008.

[16] C. Fu, Z. Zhu, “A Chaotic Image Encryption Scheme Based on Circular Bit Shift Method,” *9th International Conference for Young Computer Scientists,* pp. 3057-3061, 2008.

[17] D. Xiao, X. Liao, P. Wei, “Analysis and improvement of a chaos-based image encryption algorithm,” *Elsevier, Chaos, Solitons & Fractals,* 2007.

[18] M. Ali B. Younes, A. Jantan, “An Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption,” *IAENG International Journal of Computer Science and Network Security*, pp. 191-197, vol. 8, No. 4, 2008.

[19] NIST, “Announcing request for candidate algorithm nominations for the advanced encryption standard (AES),” *http://csrc.nist.gov/encryption/aes/pre-round1/aes_9709.htm.*

[20] Shujun Li; Xuan Zheng, “On the security of an image encryption method. *IEEE, Image Processing*, vol. 2, pp. 925-928, 2002.

[21] Z. Shi, R. Lee, "Bit Permutation Instructions for Accelerating Software Cryptography," *IEEE, Application-specific Systems, Architectures and Processors*, pp. 138-148, 2000.

[22] R. B. Lee, Z. Shi, X. Yang, “Efficient Permutation Instructions for Fast Software Cryptography,” *IEEE Micro*, vol. 21, no. 6, pp. 56-69, 2001.

[23] Y. Hilewitz, Z. J. Shi, R. B. Lee, “Comparing Fast Implementations of Bit Permutation Instruction,” *IEEE, Signals Systems and Computers*, vol.2, 1856 – 1863, 2004.

[24] M. Dworkin, “Recommendation for Block Cipher Modes of Operation. Methods and Techniques. Computers security,” Computer Security Division, *National Institute of Standards and Technology*, Gaithersburg, MD 20899-8930, 2001.

[25] S. Tao, W. Ruli, Y. Yixun, “Perturbance based algorithm to expand cycle length of chaotic key stream,” *IEEE, Electronics Letters*, vol. 34, no. 9, pp. 873-874, 1998.

[26] S. Li, X. Mou, and Y Cai, Z. Ji, J. Zhang, “On the security of a chaotic encryption scheme: problems with computerized chaos in finite computing precision,” *Computer physics communications*, vol. 153, no. 1, pp. 52-58, 2003.

[27] S. E. Assad, “Communications Numériques: Techniques avancées,” *Cours 5ème année, Ecole d’ingénieurs, Polytech’ Nantes France*, 2008.

[28] K. Burda, “Error Propagation in Various Cipher Block Modes,” *International Journal of Computer Science and Network Security*, vol. 6, no. 11, 2006.

[29] T. Xiang, X. Liao, G. Tang, Y. Chen, W. Kwok-Wo, “A novel block cryptosystem based on iterating a chaotic map,” *Physics Letters A,* vol. 349, pp**.** 109-115, 2006.