A New Chaotic Image Encryption Algorithm using a New Way of Permutation Methods

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 chaos-based cryptosystem for secure transmitted images. In the proposed block encryption/decryption algorithm, two chaotic permutation methods (key-dependant shift approach and Socek method) are used to shuffle the image pixel bits. These methods are controlled using a perturbed chaotic PWLCM map. The perturbing orbit technique improves the dynamical statistical properties of generated chaotic sequences. Our algorithm is based on tree encryption cryptosystems (Socek, Yang and Xiang algorithms). In this paper, we prove that the proposed cryptosystem overcomes the drawbacks of these algorithms. Finally, many standard tools are performed to quantify the security level of the proposed cryptosystem, and experimental results show that the suggested cryptosystem has a high security level.

Keywords—Chaos-based cryptosystem, NIST, perturbed technique.

I. Introduction

Chaos has sensitivity to initial condition and system parameter, ergodicity and mixing which are analogous to the confusion and diffusion properties of a good cryptosystem.

In recent years, a large amount of work using digital chaotic systems to construct cryptosystems has been studied [1] - [4], and has attracted more and more attention in the last years. Basically, a number of very different approaches to the use of chaos can be found in the literature [5] - [9].

In order to be used in all applications, chaotic sequences must seem absolutely random and have good cryptographic properties. Many studies on chaotic maps are drawn [10], [11].

In [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 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. In most of the natural images, the value of any given pixel can be reasonably predicted from the values of its neighbors. Many researchers have proposed schemes with combinational permutation techniques [13], [14].

In this paper, we propose an algorithm based on two chaotic permutation methods: The cyclic shift bits permutation method and a bit permutation method. The first one can be a permutation of bits, bytes or a set of bytes and the last one is applied on eight bits that their positions is also controlled by chaos.

The proposed algorithm is an enhancement of ECKBA proposed by Socek [7] and the cryptosystems proposed by Xiang [8] and Yang [9]. The algorithm proposed by Xiang has two remaining problems: the encryption speed is still slow compared with conventional cryptosystems. The encryption of a symbol needs 320-383 iterations (Table 1 in [8]) and the binary sequence, used to the substitution, will leak the trajectory of the chaotic map for easy cryptanalysis.

To overcome the drawbacks mentioned above, a new block cryptosystems with output feedback is proposed.

Socek and Yang, in their algorithms, propose to perturb the chaotic values by the encrypted data. The perturbation that they propose is not efficient because each encrypted block depends on all the previous encrypted ones. Then, if an error

occurs in the encrypted image transmitted on a noisy channel, we will obtain random errors in the decrypted image. Consequently, it is better to use an extern perturbation independent of the encrypted data, as we did in our algorithm.

In another hand, ECKBA algorithm proposed by Socek is an iterated algorithm that treats a byte in each iteration, it consists on a SP box formed by permutation and substitution controlled by PWLCM. In the proposed algorithm we propose the using of perturbed PWLCM to control two chaotic permutation methods and a substitution that enhance the security of the encryption system.

The paper is organized as follows. Section II briefly introduces the original schemes proposed by Socek [7], Xiang [8] and Yang [9]. Section III describes the proposed algorithm, Section IV introduces the perturbed generator used. Section V explains the S-box transformations used in the algorithm; the simulation results and security analysis are given in section VI. The last section concludes this paper.

II. Overview of Two Existing Algorithms

1. Socek Algorithm

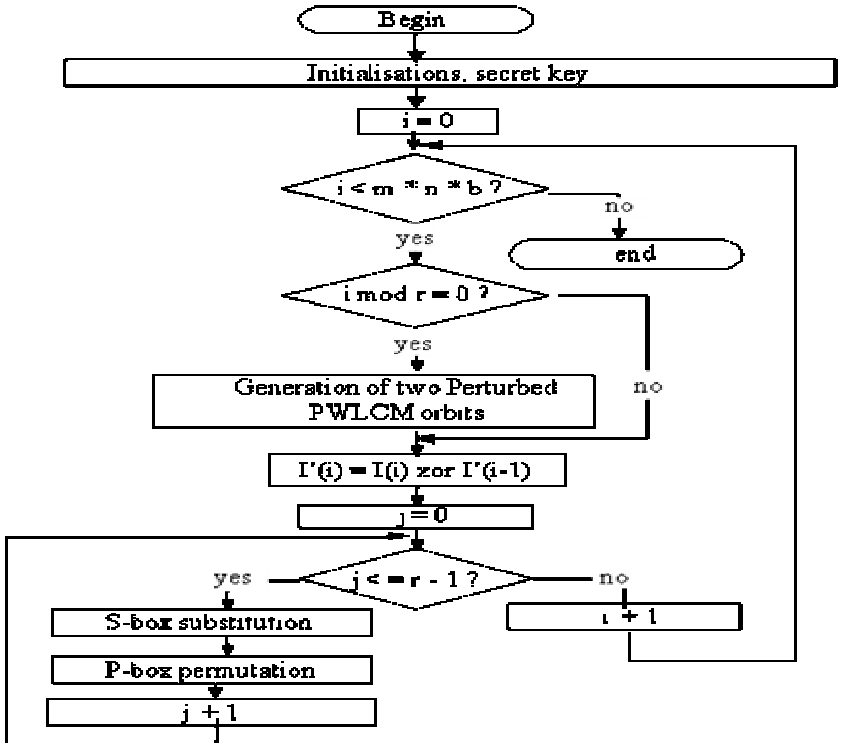


Fig. 1. Socek encryption algorithm

The algorithm, characteristics and steps are:

(1) The key size is 128-bits.
(2) The piecewise linear chaotic map encrypted by the encrypted image.
(3) A pseudo-random permutation generator is used in the encryption and decryption process, forming a permutation box (P-box) and adding diffusion to the system.
(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 *P* using 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 (see Fig. 1).

The permutation is made on the eight bits of each block formed of 4 bytes. 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 [7], [18]-[20]. Among these, Socek method [7] is the most attractive; it is fast and has good cryptographic properties (Fig. 2).

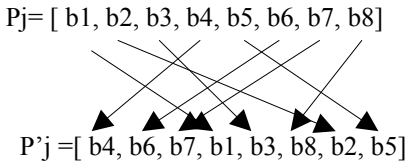


Fig. 2 : Socek method

Where Pj is the original block (byte) and P'j is the encrypted one.

2. Xiang and Yang algorithm

The proposed scheme is described below and an illustration is given in Fig. 3.
The algorithm steps are:

(1) The logistic map is iterated 70 times.
(2) Obtain binary sequences *Aj* supplied by all the third bits of the chaotic values.
(3) An integer D_j is computed as the decimal value of a part of chaotic value bits.
(4) The key dependent permutation method [8], [9] is used. This method permutes the block with left cyclic shift D_j bits as illustrated in Fig. 3.
(5) A bit xor operation is used to mask the permuted data by the binary sequence *Aj*.
(6) The value *Dj* will be used to iterate the logistic map successively after the current block has been encrypted.

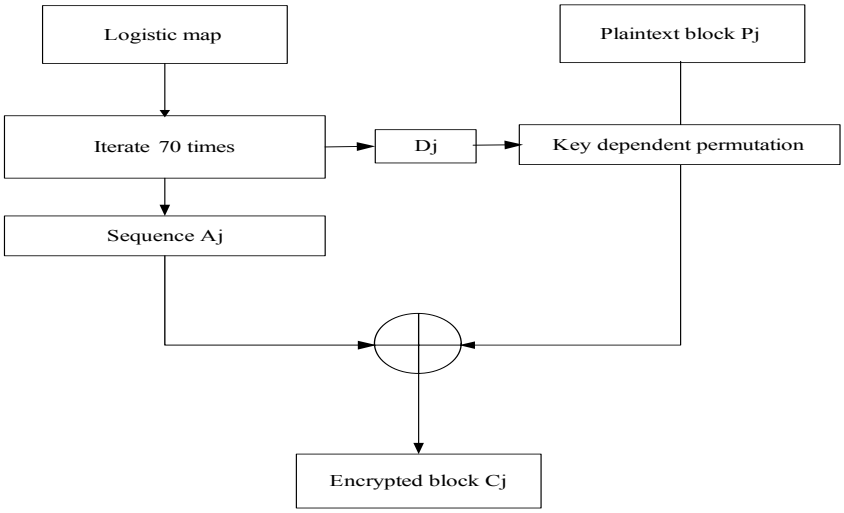


Fig. 3. Xiang encryption algorithm

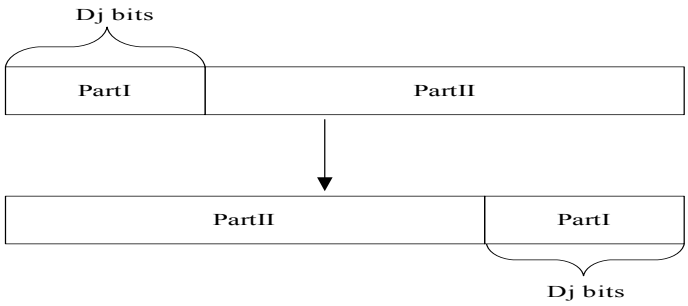


Fig. 4. Xiang method

The key dependent permutation is controlled by the chaotic value. The permutation is then different for different message blocks. It can be pixel positions permutation or a bits permutation.

But this algorithm is not secure. The binary sequence *Aj* leak the trajectory of the chaotic map for easy cryptanalysis and the encryption speed is slow. The number of iterations needed to the encryption of a symbol is big and random.

Then, Yang in his paper [9] has proposed to use an output feed back to overcome this problem. He generates the binary sequence *Aj* using the cipher image.

The use of the encrypted blocks to perturb the chaotic orbits, proposed by Socek or Yang, in their algorithm is not efficient. The resulting algorithms cause propagation of errors in the decrypted images, if a bit error accurate in the transmitted encrypted image. Consequently, they are not suitable to the transmission on a noisy channel. In [21], we propose an improvement of Socek algorithm using a different manner to perturb the chaotic orbit. But the encrypted images cannot pass all NIST tests. In the next section, we propose a new algorithm, secure and suitable for transmission compared to these algorithms.

III. Proposed Encryption Algorithm

In this section, we present the developed Algorithm 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 *P(i)*, $0 \leq i < MxN/b$. A block cipher is an encryption scheme which breaks up the plaintext messages into blocks of fixed length (32 bits or b=4 bytes) and encrypts one block at a time.

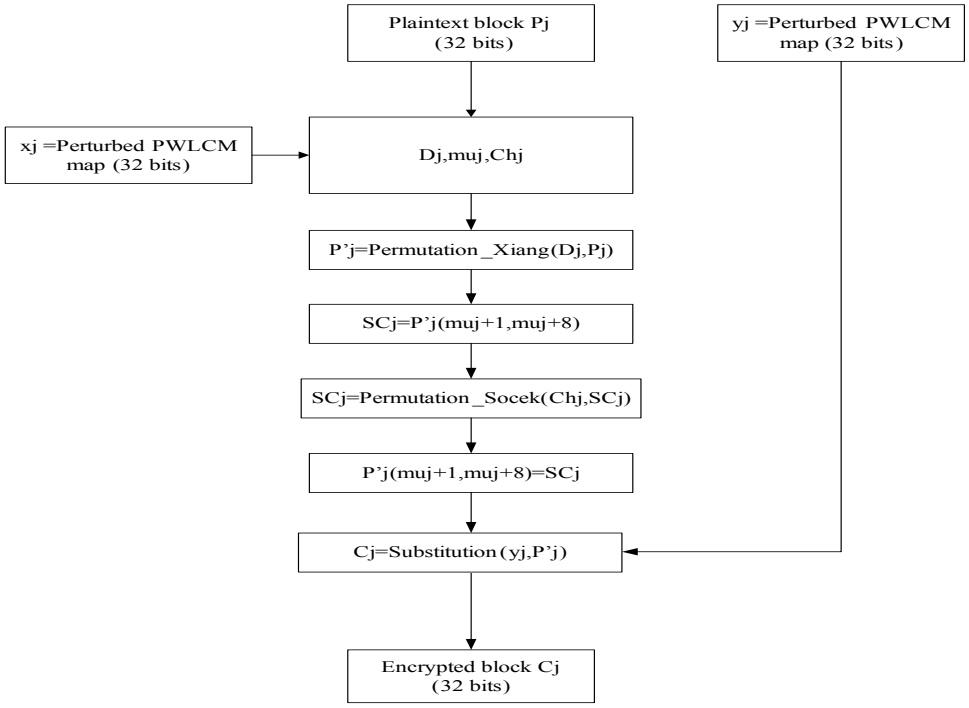


Fig. 5: Proposed encryption Algorithm

The algorithm characteristics and steps are:

(1) The key size is 128-bits.

(2) The piecewise linear chaotic map currently used is substituted by a perturbed one (perturbed PWLCM) to improve statistical properties.

(3) In fact, the chaotic value is generated on 32 bits and then decomposed on tree parts of bits. One part is considered as D_j, the second is equal to mu_j the position of the eight bits considered to be permuted by Socek method and the last part Ch_j is used to control the last permutation method.

(4) The permutation box (P-box) adding diffusion to the system includes two steps:
Firstly, the bits of each block are permuted with left cyclic shift D_j bits according to the approach illustrated in Fig. 4. Then it is permuted by Socek method. The last one permutes only 8 bits of the block. These bits are chosen by the chaotic value and this permutation is also controlled by the chaotic map.

(5). Another perturbed chaotic map is used to control substitution box (S-box). The substitution box used is the classical chaotic masking technique. The following manipulation is applied (1).

(5) $$\text{Substitution}\,(ch_j, P'_j) = chj \oplus P'_j \qquad (1)$$

where u and v are two blocks of 4 bytes.

N In order to disturb the high correlation among adjacent pixels, we propose a scheme that includes two permutation methods. These methods, Xiang and Socek ones, are chaotic. They are applied on a block of four bytes. The first one can be a bit permutation or a pixel permutation method and the second one permutes eights bits that their positions are given by the chaotic value muj.

Our algorithm not permits the propagation of errors and uses a perturbed chaotic map with good dynamical properties that we explain in the next section.

Fig. 6 and 7 show the encryption algorithms with OFB and CBC operation modes.

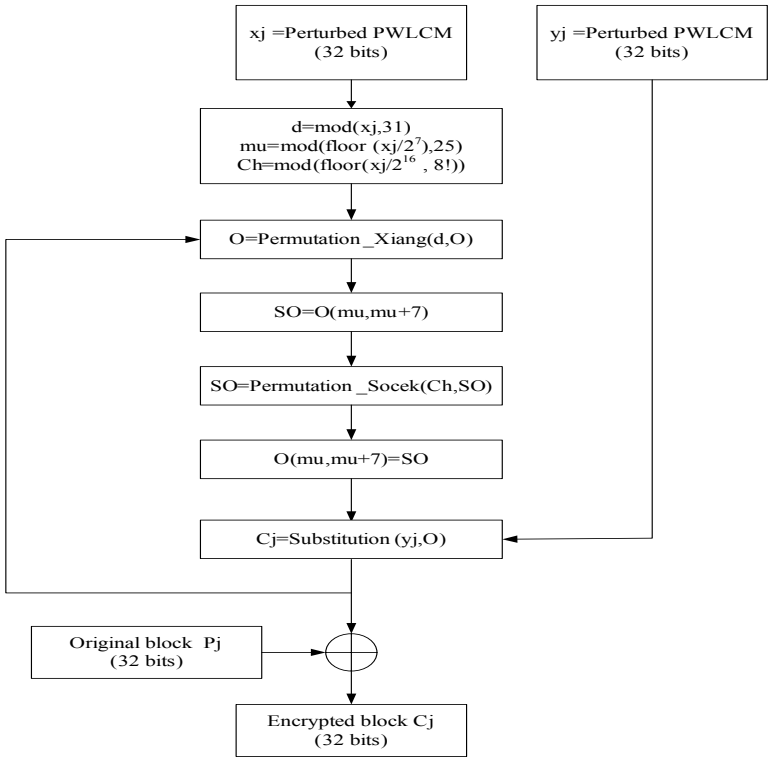


Fig.6. Proposed encryption Algorithm with OFB operation mode

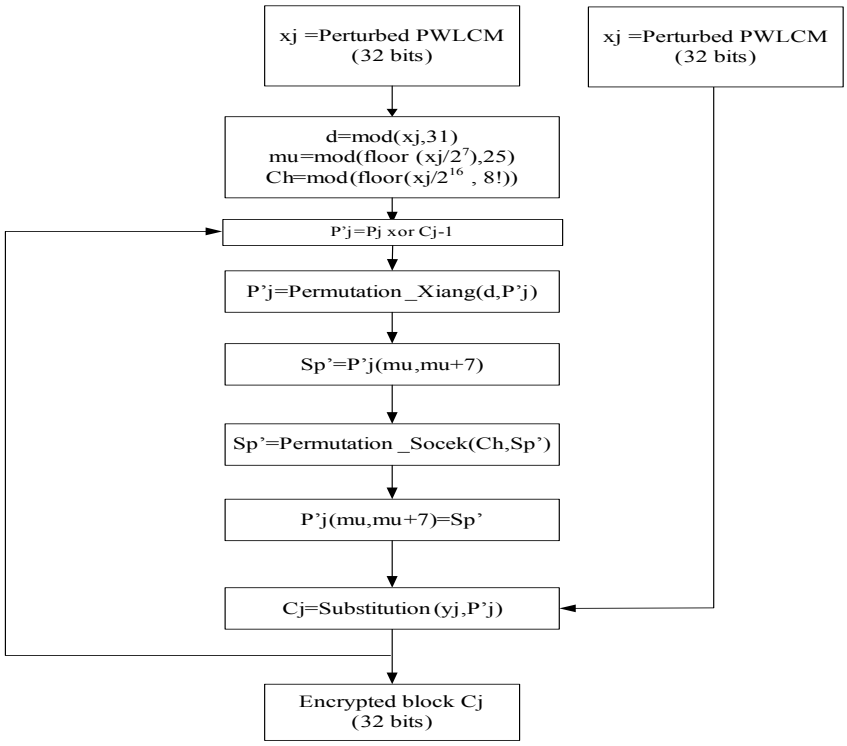


Fig.7. Proposed encryption Algorithm with CBC operation mode

IV. 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 (Fig. 8).

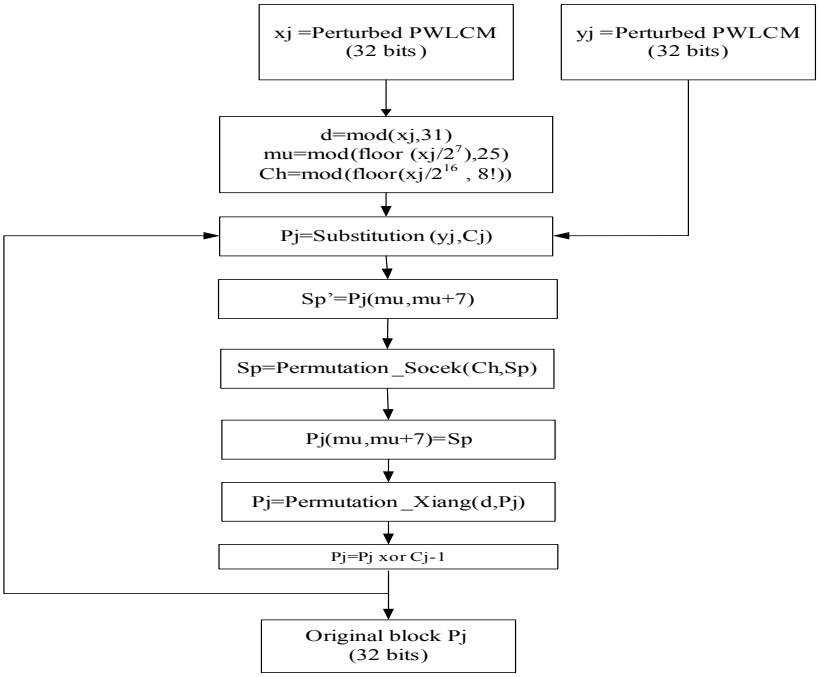


Fig.8. Proposed decryption Algorithm with CBC operation mode

To inverse 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 to reverse the order of the substitution and bit permutation methods. Then, we use the inverse methods to decrypt the image.

V. Perturbed PWLCM Map

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

$$x(n) = F[x(n-1)] = \begin{cases} x(n-1)\times\dfrac{1}{p} & \text{if } 0 \le x(n-1) < p \\ \left[x(n-1)-p\right]\times\dfrac{1}{0.5-p} & \text{if } p \le x(n-1) < 0.5 \\ F[1-x(n-1)] & \text{if } 0.5 \le x(n-1) < 1 \end{cases} \quad (2)$$

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 [15], [16]. 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 [7] 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 [15], [17].

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

$$x(n) = 0.x_1(n)x_2(n)...x_i(n)...x_N(n) \qquad x_i(n) \in \{0,1\} \quad i = 1,2,...,N \tag{3}$$

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

$$Q_{k-1}^{+}(n) = Q_k(n) = g_0 Q_0(n) \oplus g_1 Q_1(n) \oplus ... \oplus g_{k-1} Q_{k-1}(n) \quad with\ n = 0,1,2,... \tag{4}$$

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 Δ iterations (Δ is a positive integer), with *n= l*$\times\ \Delta$, *l=1,2,...,* The perturbed sequence is given by the equation (5):

$$x_i(n) = \begin{cases} F[x_i(n-1)] & 1 \le i \le N-k \\ F[x_i(n-1)] \oplus Q_{N-i}(n) & N-k+1 \le i \le N \end{cases} \tag{5}$$

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 $n \neq l \times \Delta$, no perturbation occurs, so $x(n) = F[x(n-1)]$.

The lower boundary of the system cycle length is given by the formula (6) (see appendix1):

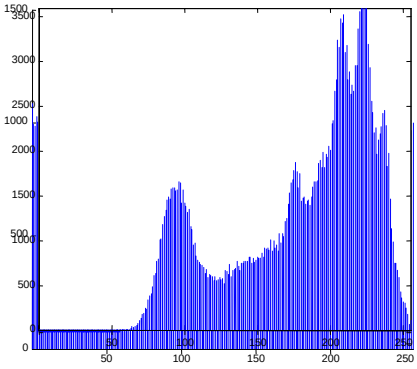
$$T_{\min} = \Delta \times \left(2^k - 1\right) \tag{6}$$

VI. Simulation results and security analysis

Some experimental results are given in this section to demonstrate the efficiency of our scheme. The plain image 'LENA.BMP' with the size 512x512 and its cipher image are shown in Fig. 9. Their histograms are shown in Fig.10. 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.9. (a) 'LENA.BMP' image and (b) his encrypted image



(a) (b)

Fig.10. Histograms of (a) 'LENA.BMP' image, (b) the encrypted image.

1. Comparison between original and 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 (7), (8) and (9), (10):

$$E(x) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N} P_1(i,j) \tag{7}$$

$$D(P_1) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N} [P_1(i,j) - E(P_1(i,j))]^2 \tag{8}$$

$$\mathrm{cov}(P_1,C_1) = \frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N} [P_1(i,j) - E(P_1(i,j))][C_1(i,j) - E(C_1(i,j))] \tag{9}$$

$$r_{P_1C_1} = \frac{\mathrm{cov}(P_1,C_1)}{\sqrt{D(P_1)}\sqrt{D(C_1)}} \tag{10}$$

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 (11):

$$D(i,j) = \begin{cases} 1 & if\ P_1(i,j) \neq C_1(i,j) \\ 0 & else \end{cases} \tag{11}$$

NPCR is defined by the following formula (12):

$$NPCR = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} D(i,j)}{M \times N} \times 100 \tag{12}$$

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 (13):

$$UACI = \frac{1}{MxN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\frac{\left|P_1(i,j) - C_1(i,j)\right|}{255}x100 \quad (13)$$

In table 1, we summarize the correlation, NPCR and UACI obtained between the original image and the encrypted one.

Table 1. The correlation, NPCR and UACI between the original image and the encrypted one

correlation	NPCR	UACI
-0.0022	99.6246	29.9932

We can see that we have obtained a low correlation between the original and the cipher image. The NPCR and UACI are high enough to say that the two images are very different.

2. 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.

Fig.9(b) shows the encrypted image with the following key: alpha= 0.35899926, beta=0.25899926, x_0=0.7239 and y_0= 0.5672.

alpha and beta are the control parameters of the PWLCM chaotic maps and x_0 and y_0 are the initial conditions of theses maps.

We encrypt the same image using the little changed key as follows: alpha= 0.35899927. We obtain a figure similar to Fig. 9(b).

Table 2 shows the difference between the two ciphered images.

Table 2. The correlation, NPCR and UACI between two cipher images encrypted with slightly different keys

Correlation	NPCR	UACI
0.0029	99.6128	33.4420

As we can see that our algorithm has a very good sensibility to the key. The two obtained encrypted images are very different and it looks like random data.

3. Correlation of 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 following formulas (7), (8), (9) and (10).

Fig.6 shows the correlation distributions of two horizontally adjacent pixels in the original and the ciphered image. In table 3, we show the correlation coefficients.

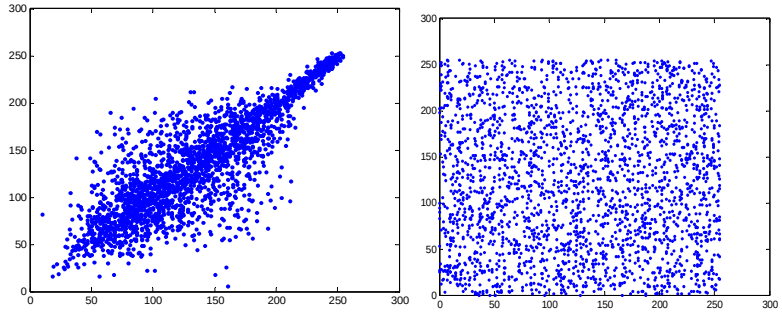


Fig.11. The correlation distributions of two horizontally adjacent pixels in (a) the original image and in (b) the ciphered image

Table 3. Correlation coefficients of adjacent pixels.

Model	Original image	Ciphered image
Horizontal	0.9829	0.0377
Vertical	0.9907	0.0107
Diagonal	0.9722	0.0119

4. 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 formula (14):

$$H(m)=\sum_{i=0}^{2^N-1} p(m_i)\log_2\frac{1}{p(m_i)} \tag{14}$$

Where $p(m_i)$ represents the probability of message m_i, $N=8$.

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 4, we show the entropy of the original image and the encrypted one by Xiang algorithm [8] and the proposed one. The values obtained are very close to the theoretical value 8 and the entropy found using our algorithm is better than the value obtained by Xiang algorithm. This means that information leakage in the encryption process is negligible and the encryption system is secure upon the entropy attack.

Table 4. Entropy value for the images encrypted with different algorithms.

Algorithm	Original image	Xiang algorithm	Proposed image
entropy	7.3479	7.9950	7.9993

5. 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 [22] 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 a sequence formed by 100 encrypted images of length 512x512=2097152 bits. We test sequence given by the improved Socek algorithm that we proposed in [21] and the explained proposed algorithm. In table 5, we show the results for a number of tests knowing that the sequences passed all the other tests. Note that the 100 encrypted images were generated with randomly selected secret keys.

Table 5. NIST statistical test for 100 encrypted images by enhanced Socek and the proposed algorithm.

STATISTICAL TEST	Improved Socek algorithm	Proposed algorithm
Frequency	93*	100
Block frequency	99	98
Runs	97	97
Longest run	97	97
Rank	100	98
Discrete Fourier Transform	99	97
Cumulative sums 1	94*	100
Approximate entropy	98	98
Universal	99	97
Serial 1	99	98
Linear complexity	98	100
Overlapping templates	99	98

VII. Propagation error

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 two cipher block modes of operation, such as: Cipher Block Chaining (CBC), and Output Feedback (OFB).

TABLE 6. The effects of bit errors for cipher block modes operation OFB and CBC.

The erroneous blocks in the ciphered image	Number of erroneous blocks in the deciphered image		The erroneous blocks in the deciphered image	
	OFB mode	CBC mode	OFB mode	CBC mode
(1,1)	1	2	(1,1)	(1,1), (1,4)
(50,100)	1	2	(50,100)	(50,100), (50,104)
(405,238)	1	2	(405,238)	(405,238), (405,241)

As we can see in table 6. 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 another one; the other bit positions are not affected. For the OFB mode, bit errors within a ciphertext block do not affect the decryption of any other block.

The results obtained for Socek and Yang algorithm not respect the expected one [23]. In fact, in their algorithms, they use a perturbation technique of the 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.

VIII. Conclusion

In this paper, a new chaos-based cryptosystem is proposed.

Our cryptosystem is based on the Socek, improved Socek algorithm, Xiang and Yang ones, but attains a higher security level, and produces cryptograms suitable to be transmitted on insecure and noisy channels.

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.

Appendix 1

Theoretical analysis of expanded cycle length

Assume that the system has entered a period T state after n_0 iterations, i.e. $x_i(n+T)=x_i(n)$ $\left(\text{for n} > \text{n}_0\ ;\ 1 \le i \le N\right)$ and $n_1 = l_1 \times \Delta > n_0$ (l_1 is a positive integer), then $x_i(n_1+T)=x_i(n_1)$ for $1 \le i \le N$. If $T \neq l \times \Delta$ (l is a positive integer), the above equation implies $F[x_i(n_1-1+T)]=F[x_i(n_1-1)] \oplus Q_{N-i}(l_1)$ $(for\ N-k+1 \le i \le N)$. Since period T is defined as $F[x_i(n_1-1+T)]=F[x_i(n_1-1)]$ for $(1 \le i \le N)$, thus, $Q_{N-i}(l_1)=0$ $(for\ N-k+1 \le i \le N)$. Because the initial sequences $Q_0, Q_1, ..., Q_{k-1}$ are not all zeros, the previous case will not occur. This implies that we only have $T = l \times \Delta$, which means $F[x_i(n_1-1+T)] \oplus Q_{N-i}(l+l_1) = F[x_i(n_1-1)] \oplus Q_{N-i}(l_1)$ $(\text{for } N-k+1 \le i \le N)$. As a result, we find $Q_{N-i}(l+l_1)=Q_{N-i}(l_1)$ (for $N-k+1 \le i \le N$). This implies: $l = \sigma\left(2^k-1\right)$ where σ is a positive integer. Therefore the system cycle length is given by: $T = \sigma \times \Delta \times (2^k-1)$ and $T_{\min} = \Delta \times (2^L-1)$ is the lower bound of the system cycle length.

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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] T. Xiang, X. Liao, G. Tang, Y. Chen, K. W. Wong, "A novel block cryptosystem based on iterating a chaotic map". *Elsevier,Physics Letter A,*2005.

[9] D. Yang, X. Liao, Y. Wang, H. Yang, P. Wei, "A novel block cryptosystem based on iterating map with output feed-back". *Elsevier,Chaos, Solitons and Fractals,*2008.

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

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] S. E. Assad, "Communications Numériques: Techniques avancées," *Cours 5ème année, Ecole d'ingénieurs, Polytech' Nantes France*, 2008.

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

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

[20] 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.

[21] Abir Awad, Safwan El Assad, Daniel Carragata . "A Robust Cryptosystem Based Chaos for Secure Data**".** *IEEE* **,** *International Symposium on Image/Video communications over fixed and mobile networks.* Bilbao,spain 2008.

[22] A. Rukin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo. "A Statistical Test Suite For Random and Pseudorandom Number Generators for cryptographic applications". NIST Special Publication 800-22 (with revisions dated May 15, 2001).

[23] 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.