Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 41(1), 2014, Pages 47–58
ISSN: 1223-6934

A novel chaos-based image encryption scheme

Ana Cristina Dăscălescu, Radu Boriga, and Marius Iulian Mihăilescu

Abstract. Recently, the chaotic maps have been investigated in order to develop more secure encryption schemes. In this paper we propose a new image encryption scheme with a classic bi-modular architecture: a diffusion stage, in which the pixels of the plain image are shuffled using a random permutation generated with a new algorithm, and a confusion stage, in which the pixels are modified with a XOR-scheme. In both stages are used four piecewise linear chaotic maps with very good cryptographic properties. The results of its statistical analysis show that the proposed image encryption scheme provides an efficient and secure way for image encryption.

2010 Mathematics Subject Classification. Primary 94A60; Secondary 68P25.
Key words and phrases. chaos based cryptography, chaotic maps, image encryption, piecewise linear chaotic map, keystream encryption.

1. Introduction

Recently, chaotic maps have been widely used in new proposed image encryption scheme due to their properties such as sensitivity to initial conditions and control parameters, pseudo randomness, ergodicity and simple analytical forms ([1], [2],[3],[4],[5], [6]). Compared with traditional cryptosystems ([5],[6]) the ones based on chaos are easier to be realized, which makes it more suitable for large-scale data encryption such as images, videos or audio data. However, there is still not a model for security analysis that is suitable for most chaos-based cryptosystems, which makes that chaotic cryptosystems are seldom accompanied by detailed security analysis ([4],[9],[10],[11], [12]). Mostly, the proposed chaos based encryption schemes used chaotic maps to generate a random permutation for shuffling the pixels from plain image and to generate a pseudo-random bit sequence for substituting the pixels, in order to obtain an encrypted image. The secret key consist from the control parameters of the chaotic map and its initial conditions, so it's necessary to use maps which are chaotic for a large interval of parameter's values. In this paper we propose a new chaos based image encryption, based on Piece-Wise Linear Chaotic Map (PWLCM), which has two stages: a diffusion stage, in which the pixels of plain image are shuffled using a random permutation generated with a new efficient algorithm, and the confusion stage, in which the pixels are modify using a XOR-scheme based on the proposed map. The keystream is obtained using 2 PWLCMs, randomly chosen from 4 PWLCMs implied in the encryption scheme. The paper is organized as follows: in section 2 we presents the piecewise linear chaotic map used in the image encryption scheme; section 3 presents a new image encryption scheme based on the map presented in section 2; section 4 presents the performance analysis of the proposed image encryption scheme. Finally, section 5 concludes the work carried out.

Received February 16, 2014.

2. PWLCM MAP

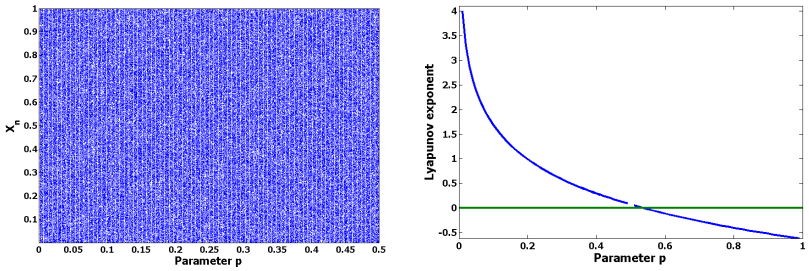
Some of the simplest chaotic systems commonly used in chaos based encryption scheme is PWLCM, which is a map composed of multiple linear segments (limited breaking points are allowed), defined by ([13],[14]):

$$PWLCM : (0,1) \to (0,1),$$

$$PWLCM(p, x_n) = \begin{cases} \frac{x_n}{p}, 0 \leq x_n < p \\ \frac{x_n - p}{0.5 - p}, p \leq x_n < 0.5 \\ PWLCM(p, 1 - x_n), 0.5 \leq x_n < 1 \end{cases} \quad (1)$$

where $p \in (0,1)$ is the control parameter and $x_0 \in (0,1)$ is the initial condition.

The time behaviour of PWLCM depends on the control parameter p. In Figure 1a is plotted the bifurcation diagram for $p \in (0, 0.5)$ and in Figure 1b is plotted the Lyapunov exponent for $p \in (0, 0.1)$.



(a) Bifurcation diagram of PWLCM (b) Lyapunov exponent of PWLCM

FIGURE 1. Time behaviour of PWLCM

From the figure above it can be observe that the PWLCM is in a chaotic regime for $p \in (0, 0.5)$. PWLCM has uniform invariant distribution and very good ergodicity, confusion and determinacy, so it can provide excellent random sequence, being suitable for information encryption ([13],[14]).

3. The proposed scheme

3.1. Description of the proposed cryptosystem. The proposed cryptosystem is a symmetric one and has a bi-modular architecture, in which one of the modules performs the diffusion process using a random permutation generated by 4 PWLCMs (1), while the second one performs the confusion process by modifying pixels' values using a deterministic algorithm which implies the same 4 chaotic PWLCMs (1).

Assuming that the pixels of a RGB image with size $n = W \times H$ pixels are numbered on rows, from top to down and from left to right on each row, we denote the plain image by $P = \{p_0, p_1, \ldots, p_{n-1}\}$ and the corresponding encrypted one by $C = \{c_0, c_1, \ldots, c_{n-1}\}$, both of same size n. Also, both in the encryption and decryption process, we will use an auxiliary image of the same size n, denoted by $A = \{a_0, a_1, \ldots, a_{n-1}\}$.

Next, we describe in detail the implementation and functionality of each module of the proposed cryptosystem.

3.2. The secret key. The secret key of the proposed cryptosystem, shared by the emitter and the receiver, consists of the next elements:

- 4 real numbers r_1,r_2,r_3 and r_4 representing the parameters of the 4 PWLCMs (1), chosen so as all maps to be in a chaotic regime, i.e. r_1,r_2,r_3,$r_4 \in (0, 0.5)$;
- 4 real numbers x_0^1, x_0^2, x_0^3 and x_0^4 representing the initial conditions of the 4 PWLCMs (1), chosen from the interval $(0, 1)$;
- 4 unsigned integers m_1, m_2, m_3 and m_4 representing the number of the pre-iterations of the 4 PWLCMs (1), required to assure them a chaotic behavior - we recommend to choose m_1, m_2 m_3, $m_4 \geq 1000$;
- 1 unsigned integer IV, representing the initial value used to encrypt/decrypt the first pixel of the plain/encrypted image.

3.3. The encryption scheme.

3.3.1. *The diffusion process.* Usually, in an image encryption scheme, the diffusion process is assured by the permutation of the pixels from the plain image, so that the default redundancy of the image will be spread over the whole encrypted image ([2],[3],[5],[7],[5],[16],[17],[18],[19],[20]).

The proposed algorithm combines the use of random values, generated using the 4 PWLCMs (1), with the use of non-random ones, determined algorithmically. Thus, a permutation $q = (q_1, q_2, \ldots, q_n)$ of degree n is constructed element by element, as follows: all the 4 random unsigned integer values between 1 and n, obtained by discretizing and scaling the real values generated by the 4 PWLCMs (1), are checked if were previously used - if not, up to 4 elements of the permutation are completed properly, otherwise the minimum unused value is assigned to the current element of the permutation.

Algorithm 1 Generate a random permutation $q = (q_1, q_2, \ldots, q_n)$ of degree n

Input: unsigned integer n, array of real numbers $r[1..4] \in (0, 0.5)$, array of real numbers $x_0[1..4] \in (0, 1)$ and unsigned integer array $m[1..4] \in \{1000, 1001, \ldots, 2000\}$
Output: array $q[1..n]$ contains a random permutation

{array of real numbers $x[1..4]$ stores the current values of the 4 PWLCMs}
for $i = 1$ to 4 **do**
  $x[i] \leftarrow x_0[i]$
end for

{the 4 PWLCMs are pre-iterated}
for $i = 1$ to 4 **do**
  for $j = 1$ to $m[i]$ **do**
    $x[i] \leftarrow PWLCM(r[i], x[i])$
  end for
end for

{$L[1..n]$ is a labeling array, i.e. $L[i] = 1$ if a value $i \in \{1, 2, \ldots, n\}$ has already used in permutation q, otherwise $L[i] = 0$}
for $i = 1$ to n **do**
  $L[i] \leftarrow 0$
end for

Algorithm 1 (continued)

{variable min stores the minimum unused value between 1 and n and variable i represents the index of the current element of the random permutation q}
$min \leftarrow 0$
$i \leftarrow 1$
while $i \leq n$ **do**
{the discretized and scaled real value of the current PWLCM is assigned to the current element of the permutation q - function $floor(x)$ returns the nearest integer less than or equal to the real number x}
$b \leftarrow false$
for $j = 1$ to 4 **do**
$q[i] \leftarrow 1 + floor(10^15 \times x[j]) \mod n$
if $L[q[i]] = 0$ **then**
$i \leftarrow i + 1$
$b \leftarrow true$
end if
end for

{if all the 4 random values generated by the PWLCMs were previously used in the permutation q, then the minimum unused value between 1 and n is assigned to $q[i]$}
if $b = false$ **then**
$j \leftarrow min + 1$
while $j \leq n$ and $L[j] = 1$ **do**
$j \leq j + 1$
end while
$min \leftarrow j$
$q[i] \leftarrow min$
$L[q[i]] \leftarrow 1$
$i \leftarrow i + 1$
end if

{all the 4 PWLCMs are iterated one time}
for $j = 1$ to 4 **do**
$x[j] \leftarrow PWLCM(r[j], x[j])$
end for
end while

It's obviously that Algorithm 1 has a maximum complexity of $O(n^2)$, but, due to the fact that PWLCMs are chaotic and ergodic, its average complexity will be close to $O(n)$. Moreover, the random permutations generated have a very small number of fixed points, proved by the fact that in the case of 1000000 generated permutations, with lengths between 1000 and 10000000, the maximum percent of fixed points was less than 0.01%.

In the encryption process, the pixels from the plain image will be shuffled using the permutation q into the auxiliary image A, while in the decryption process the pixels from the auxiliary image A will be un-shuffled using the inverse permutation

q^{-1}, both by an algorithm having an $O(n)$ complexity, so the overall complexity of the proposed algorithm remains the same.

3.3.2. *The confusion process.* Commonly, in an image encryption scheme, the confusion process is used to hide the correlations between the plain image and the encrypted image or the encryption key, through substitutions of the pixels' values in a deterministic manner ([2],[5],[7],[15],[16],[17],[18],[19],[20]).

In order to ensure a high level of security against differential attacks, in the confusion process we involved the sum of the pixels previously encrypted. Thus, assuming that the current pixel is c_i, we define the sum s_i of the pixels previously encrypted as it follows:

$$s_i = \sum_{j=0}^{i-1} c_j \tag{2}$$

for any $i \in \{1, \ldots, n-1\}$ and $s_0 = 0$.

The confusion process is based on the same 4 PWLCM maps, whose current values are stored in the same array $x[1..4]$ used in the diffusion process, thereby:

$$\begin{cases} c_0 = a_0 \oplus IV \oplus floor(10^{15} \times x[1 + IV \bmod 4\ ]) \\ c_i = (((a_i \oplus u_i) + s_i) \bmod 256) \oplus c_{i-1} \oplus v_i, 1 \leq i \leq n-1 \end{cases} \tag{3}$$

where the auxiliary image $A = \{a_0, a_1, \ldots, a_{n-1}\}$ contains the pixels of the plain image P after the shuffling process, $u_i = floor(10^{15} \times x[1 + s_i \bmod 4])$ and $v_i = floor(10^{15} \times x[4 - s_i \bmod 4])$. After the encryption of each pixel, all the 4 PWLCM maps are iterated one more time.

3.4. The decryption process. Due to the fact that the proposed cryptosystem is a symmetric one, in the decryption process is used the same secret key, so the decryption process will consist in the next two steps:
Step 1: Apply the inverse transformation of (3) upon the pixels of the encrypted image C, obtaining the auxiliary image A:

$$\begin{cases} a_0 = c_0 \oplus IV \oplus floor(10^{15} \times x[1 + IV \bmod 4\ ]) \\ a_i = (((c_i \oplus c_{i-1} \oplus v_i) + 256 - s_i) \bmod 256\ ) \oplus u_i, 1 \leq i \leq n-1 \end{cases} \tag{4}$$

where s_i, u_i and v_i are defined as above.
Step 2: Apply the inverse q^{-1} of the permutation q upon the pixels of the auxiliary image A, obtaining the plain image P.

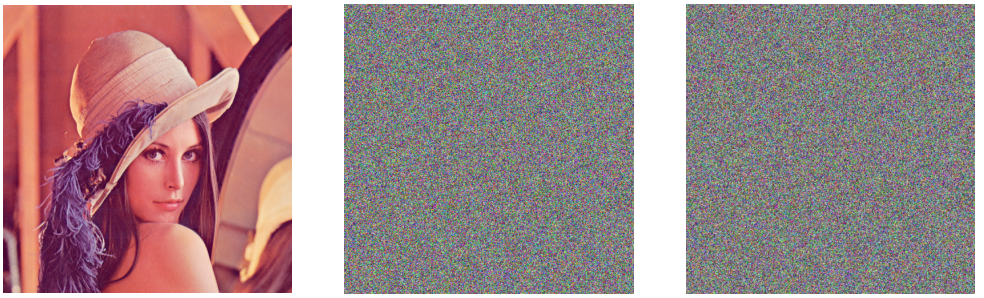
4. Performances of the proposed cryptosystem

4.1. Security analysis of the proposed cryptosystem. A strong encryption scheme should resist against known cryptanalytic attacks, such as brute-force attack, statistical attack, known-plain-text attack, differential attack etc. Thus, for the proposed image encryption scheme, we performed some standard security analysis: key space analysis, statistical analysis and differential analysis. Hence, we performed several specific statistical tests, such as image pixels distribution, the correlation between adjacent pixels of the image encrypted, entropy, NPCR and UACI. The analysis process involved 10 various standard test images from USC-SIPI Image Database ([21]), Kodak Digital Camera Sample Pictures ([22]), personal photos etc. All the pictures used were 24 bit-color bitmaps, with different dimensions varying from 256×256 to

3000×4000 pixels. All tests were performed for each RGB channel, in order to achieve a rigorous and detailed analysis of the performances of the proposed scheme.

4.2. Key space analysis. A secure image encryption algorithm should have an enough large key space in order to resist to brute-force attacks. The secret key of the proposed cryptosystem consists from 8 real numbers and 5 unsigned integers. The real numbers must to be handled using a real data type with high precision, in order to prevent the negative effects caused by the discretization. If the implementation of the cryptosystem is done using a programming language that complies with IEEE Standard 754-2008 ([23]), then it's recommended to use the double data type, which stores real numbers on 8 bytes, with an accuracy of 15 decimal places. Thus, the length of the secret key will be 672 bits, which means that the size of the secret key's space will be equal to $2^{672} \approx 1.9510^{202}$, a enough large value to prevent guessing the secret key in a reasonable time, using a brute-force algorithm.

4.3. Key sensitivity analysis. High key sensitive is required by secure cryptosystem, which means that the use of two secret keys which are very small different one from another lead to two completely different encrypted images. This ensure that encrypted image cannot be decrypted correctly although there is only a slight difference between encryption keys. In Figure 2b and Figure 2c are shown 2 images obtained by encrypting Lena image (Figure 2a) using 2 secret keys which has only a double data type component different by 10^{-12}. Note that the images are totally different from the plain-image Lena and, moreover, the decrypted image seems to be a noise.



(a) Lena image

(b) Lena image encrypted with the first secret key

(c) Lena image encrypted with the second secret key

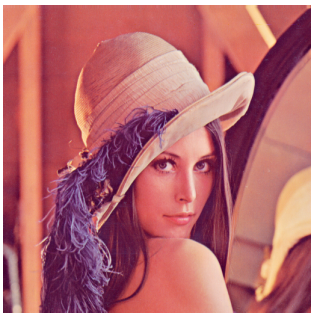
FIGURE 2. Images resulted by encrypting the Lena image using two secret keys which differ by 10^{-12} on a single double data type component

In this testing process of sensitivity analysis, we considered 10 plain images and 10 corresponding secret keys, too. Each plain image was encrypted using the corresponding secret key and the encrypted image obtained was decrypted using 13 secret keys, which differ from the initial one by 10^{-12} on components of double data type or by one bit on components of unsigned integer data type, and all the 13 decrypted images obtained were compared with the initial plain image. In all the 130 cases, we obtained a correlation coefficient very close to 0, which confirms that the decrypted images are completely different from the initial plain images.

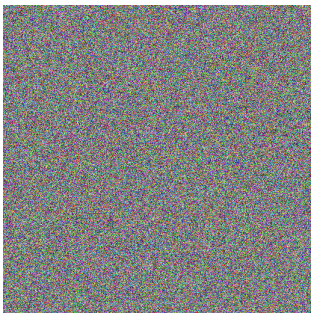
So, we can conclude that the proposed encryption algorithm is very sensitive to the key, due to that small change in the secret key will generate a completely different decrypted image from the original one.

4.4. Statistical analysis. In order to prove that the proposed image encryption system has a superior confusion and diffusion properties, we performed tests on the histograms and entropies of the encrypted images, along with a statistical test on the correlations between adjacent pixels in the encrypted image.

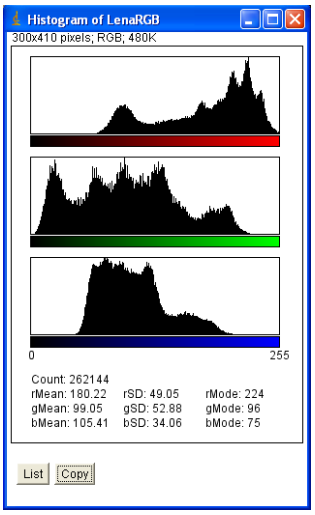
4.4.1. *Histogram of the encrypted images.* In order to hide the uneven distribution from the plain image, a cryptosystem with high security level needs to produce encrypted images with a uniform distribution of pixels in each color channel. To study the distribution of pixel values of a color image, the most often used tools for a visual analysis is the color histogram. Here, the pixel values frequencies are plotted separately for each color channel. Figure 3a and Figure 3b contains a pair of plain image - encrypted image, along with the associated color histograms (Figure 3c and Figure 3d).



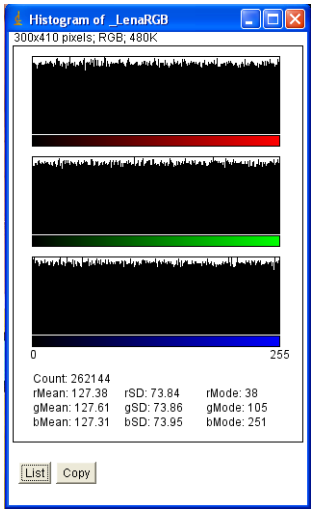
(a) Lena image



(b) Encrypted Lena image



(c) Histogram of the Lena image



(d) Histogram of the encrypted Lena image

FIGURE 3. Analysis of the standard image Lena

Note that the plain image Lena has a strong color uneven distribution (Figure 3c), but after the encryption, we obtained an image with a uniform distribution pixels values (Figure 3d) for each color channel. Thus, am attacker cannot extract statistical information about the plain image or about the encryption key used.

To analyze the distribution of pixel values for a large number of encrypted images, we used the χ^2 test ([24]). The value of the χ^2 test for an encrypted image of dimension $m \times n$ pixels is given by the following formula:

$$\chi^2 = \sum_{i=0}^{255} \frac{(v_i - v_0)^2}{v_0} \tag{5}$$

where v_i is the observed frequency of a pixel value i ($0 \leq 255$) and v_0 is the expected frequency of a pixel value i, so $v_0 = \frac{m \cdot n}{256}$.

The results obtained by applying the χ^2 test on 10 encrypted images can be summarized as it follows: for 9 images the values of the χ^2 test obtained were lower than the critical value $\chi^2_{255,0.05} = 293.25$ and for one image the value obtained was 294.07,

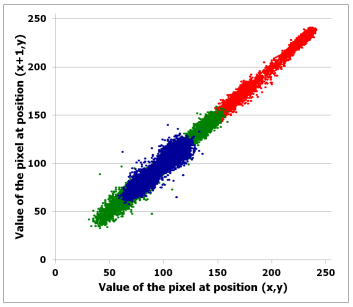
which is very close to the critical value of 293.25. Thus, we conclude that the distribution of pixel values is uniform in the encrypted image, which demonstrates that the proposed cryptosystem is able to resist against statistical attacks.

4.4.2. *Correlation of adjacent pixels.* In order to evaluate the encryption quality of the proposed encryption algorithm, the correlation coefficient is used. Firstly, we randomly selected 10000 pairs of two adjacent in pixels from plain/encrypted image, and, using the values from each color channel RGB, we constructed two series of data $X = \{x_1, x_2, \ldots, x_{10000}\}$ and $Y = \{y_1, y_2, \ldots, y_{10000}\}$. Then, we calculated the correlation coefficient between X and Y using the following formula:

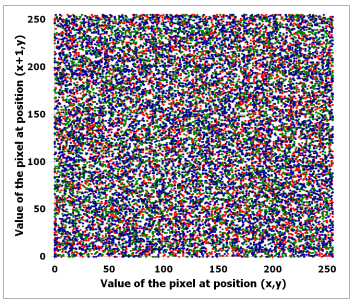
$$p_c(X, Y) = \frac{cov(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \tag{6}$$

where C is the colour channel, $D(\cdot)$ denotes the variance of a random variable and $cov(\cdot, \cdot)$ denotes the covariance of two random variables ([24]).

In Figure 3a we plotted the value of the pixel at the position (x, y) versus the value of the pixel at the position $(x + 1, y)$ from the Lena image, while in Figure 3b we plotted them from the encrypted Lena image. We repeated the same plotting for vertically adjacent pixels (Figure 3c and 3d), respectively for diagonally adjacent pixels (Figure 3e and Figure 3f).



(a) Correlation of horizontally adjacent pixels in the original image

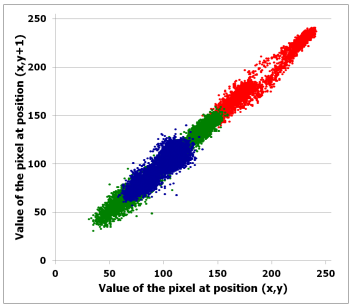


(b) Correlation of horizontally adjacent pixels in the encrypted image

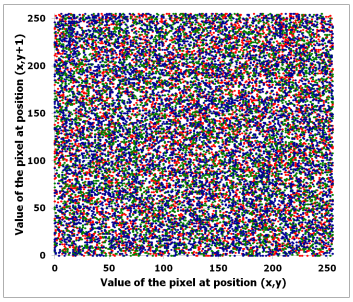
For all the 10 pairs of plain/encrypted test we obtained average values of the correlation coefficient from the interval $[-0.01538, 0.02613]$, so very close to 0, which confirms that the encryption process eliminates the inherent strong correlation existing between the pixels of the plain image.

4.5. Differential attack analysis. Usually, an opponent, trying to find out a meaningful relationship between the plain image and the encryption image, would make a slight change (e.g. modify only one pixel) of the encrypted image to observe the change of the corresponding results. If one minor change in the plain image can lead to a significant change in the encrypted image, with respect both to the confusion and diffusion, then the differential attacks may become inefficient.

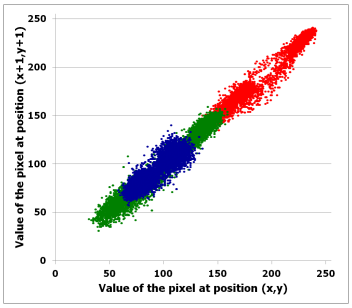
So, to test the resistance against differential attacks is necessary to evaluate how a minor change in the plain image is reflected in the encrypted image. In this sense, to common measures may be used: *Number of Pixels Change Rate (NPCR)* and *Unified Average Changing Intensity (UACI)* ([20]).



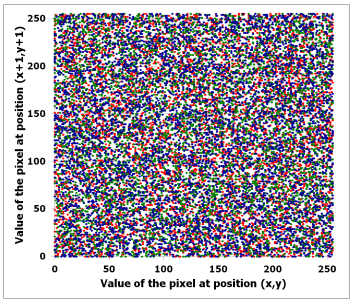
(c) Correlation of vertically adjacent pixels in the original image



(d) Correlation of vertically adjacent pixels in the encrypted image



(e) Correlation of diagonally adjacent pixels in the original image



(f) Correlation of diagonally adjacent pixels in the encrypted image

FIGURE 3. Correlation of adjacent pixels from Lena plain/encrypted image

We consider two plain images $P_1 = \{p_0^{(1)}, p_1^{(1)}, \ldots, p_{n-1}^{(1)}\}$ and $P_2 = \{p_0^{(2)}, p_1^{(2)}, \ldots, p_{n-1}^{(2)}\}$ which differ by the value of a single pixel in a RGB color channel and their corresponding encrypted images $C_1 = \{c_0^{(1)}, c_1^{(1)}, \ldots, c_{n-1}^{(1)}\}$ and $C_2 = \{c_0^{(2)}, c_1^{(2)}, \ldots, c_{n-1}^{(2)}\}$.

The NPCR indicator measures the percentage of different pixel numbers between the encrypted images C_1 and C_2 and it's defined as it follows:

$$NPCR = \left(\frac{1}{n} \sum_{i=0}^{n-1} s_i \right) \times 100\% \tag{7}$$

where $s_i = 0$ if $c_i^{(1)} = c_i^{(2)}$ and $s_i = 1$ if $c_i^{(1)} \neq c_i^{(2)}$ for any $i \in \{0, 1, \ldots, n-1\}$.

The UACI indicator measures the average intensity of differences between the encrypted images C_1 and C_2 and it's defined as it follows:

$$UACI = \left(\frac{1}{n} \sum_{i=0}^{n-1} \frac{\left| c_i^{(1)} - c_i^{(2)} \right|}{255} \right) \times 100\% \tag{8}$$

Considering two random images, the maximum expected value of NPCR is found to be 99.609375%, while the maximum expected value of UACI is 33.463541% ([20]). Using the proposed cryptosystem were performed 10 tests, achieving values of the NPCR indicator between 97.34% and 99.08% and between 31.66% and 33.12% for

the UACI indicator, which confirms that the proposed cryptosystem will withstand to the differential attacks.

4.6. Quality of the decryption process. Among the evaluation of the quality of the encryption process, it should be checked the quality of the decryption process, too. Basically, this consists in testing that the image obtained after decryption process coincides with the plain one. For this, we evaluated the Mean Squared Error (MSE) between a plain image $P = \{p_0, p_1, \ldots, p_{n-1}\}$ and the corresponding decrypted one $D = \{d_0, d_1, \ldots, d_{n-1}\}$, on each RGB color channel, using the following formula ([24]):

$$MSE(P, D) = \frac{1}{n} \sum_{i=0}^{n-1} (p_i - d_i)^2 \tag{9}$$

A value close to 0 of MSE indicates a good quality of the decryption process, while other values indicate the occurrence of errors in this process.

In all the 10 tests performed, the value of MSE was 0 for each RGB color channel, which indicates that decryption is carried out without loss of information.

4.7. Speed performance. An encryption scheme is considered to be efficient also in respect to its speed. In this sense, we run the proposed algorithm implemented in C language (MinGW compiler) under Windows 7, using a PC with Intel(R) Core(TM) I3 @2.53GHz CPU and 3GB RAM. We used 10 standard test bitmaps with sizes of $256 \times 256, 512 \times 512, 1024 \times 1024$ and 3000×4000 ([21],[22]). The mean speeds obtained are summarized in Table 1:

TABLE 1. Speed performance of the proposed cryptosystem (encryption/decryption)

Image size (pixels)	Image Size (MB)	Mean Time (s)	Mean Speed (MB/s)
256×256	0.19	0.09	2.11
512×512	0.75	0.38	1.97
1024×1024	3.00	1.37	2.19
3000×4000	34.33	16.13	2.13

Analyzing the mean speeds from Table 1, we can conclude that the proposed algorithm is fast, having a mean encryption/decryption speed about 2 MB/s.

5. Conclusions

In this paper we proposed a new image encryption scheme based on four PWLCMs. The pixels from the plain image are shuffled with a random permutation obtained by a new efficient algorithm. To make the encryption scheme more robust against cryptanalytic attacks, we altered the pixels' values using an efficient new XOR-scheme, based on a keystream obtained by discretizing the values extracted from the orbits of the 4 PWLCMs implied. Finally, we proved the very good cryptographic performances of the proposed image encryption scheme through an extensive analysis, performed with respect to the latest methodology from this field.

References

[1] J.C. Yen and J.I. Guo, A new chaotic key-based design for image encryption and decryption, *IEEE International Symposium on Circuits and Systems* (ISCAS), Geneva, Switzerland, May 28-31, 2000, IEEE Press, Lausanne (2000), 49-52.

[2] G. Chen, Y. Mao and C.K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, *Chaos, Solitons and Fractals* **21** (2004), 749–761.

[3] T. Gao and Z. Chen, A new image encryption algorithm based on hyper-chaos, *Physics Letters A* **372** (2008), no. 4, 394–400.

[4] X. Wang and G. He, Cryptanalysis on a novel image encryption method based on total shuffling scheme, *Optics Communications* **284** (2011), no. 24, 5804–5807.

[5] H. Yang, X. Lia, K.W. Wong, W. Zhang and P. Wei, A new cryptosystem based on chaotic map and operations algebraic, *Chaos, Solitons and Fractals* **40** (2009), no. 5, 2520–2531.

[6] Y.B. Mao, G. Chen, S.G. Lian, A novel fast image encryption scheme based on the 3D chaotic baker map, *International Journal on Bifurcation and Chaos* **14** (2004), no. 10, 3613–3624.

[7] A. Cheddad, J. Condell, K. Curran and P. McKevitt, A hash-based image encryption algorithm, *Optics Communications* **283** (2010), 879–893.

[8] F. Riaz, S. Hameed, I. Shafi, R. Kausar and A. Ahmed, Enhanced Image Encryption Techniques Using Modified Advanced Encryption Standard, *Communications in Computer and Information Science* **281** (2012) 385–396.

[9] E. Solak, C. Çokal, O. T. Yildiz and T. Biyikoglu, Cryptanalysis of Fridrich's chaotic image encryption, *International Journal on Bifurcation and Chaos* **20** (2010), no. 5, 1405–1413.

[10] C. Li, Y. Liu, T. Xie and M.Z.Q. Chen, Breaking a novel image encryption scheme based on improved hyperchaotic sequences, *Nonlinear Dynamics* **73** (2013), no. 3, 2083–2089.

[11] S. Li, G. Chen and X. Mou, On the security of the Yi-Tan-Siew chaotic cipher, *IEEE Transactions on Circuits and Systems II: Express Briefs* **51** (2004), no. 12, 665–669.

[12] G. Alvarez and S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, *International Journal of Bifurcation and Chaos* **16** (2006), 2129–2151.

[13] S. Li, G. Chen and X. Mou, On the Dynamical Degradation of Digital Piecewise Linear Chaotic Map, *International Journal on Bifurcation and Chaos* **15** (2005), no. 10, 3119–3151.

[14] Z.G. Xu, Q. Tian and L. Tian, Theorem to Generate Independently and Uniformly Distributed Chaotic Key Stream via Topologically Conjugated Maps of Tent Map, *Mathematical Problems in Engineering* **2012** (2012), Article ID: 619257.

[15] K.W. Wong, B.S.H. Kwok and W.S. Law, A fast image encryption scheme based on chaotic standard map, *Physics Letters A* **372** (2008), no. 15, 2645–2652.

[16] G. Ye and K.W. Wong, An efficient chaotic image encryption algorithm based on a generalized Arnold map, *Nonlinear Dynamics* **69** (2012), no. 4, 2079–2087.

[17] G. Ye and K.W. Wong, An image encryption scheme based on time-delay and hyperchaotic system, *Nonlinear Dynamics* **71** (2013), no. 1-2, 259–267.

[18] A.V. Diaconu and K. Loukhaoukha, An Improved Secure Image Encryption Algorithm Based on Rubik's Cube Principle and Digital Chaotic Cipher, *Mathematical Problems in Engineering* **2013** (2013), Article ID: 848392.

[19] C. Zhu, A novel image encryption scheme based on improved hyperchaotic sequences, *Optics Communications* **285** (2012), no. 1, 29–37.

[20] H.S. Kwok and W.K.S. Tang, A fast image encryption system based on chaotic maps with finite precision representation, *Chaos, Solitons and Fractals* **32** (2007), no. 4, 1518–1529.

[21] USC-SIPI Image Database, University of South California, Signal and Image Processing Institute: `http://sipi.usc.edu/database/database.php`.

[22] Kodak Digital Camera Sample Pictures, `http://www.kodak.com/digitalImaging/samples/classic.shtml`.

[23] IEEE Standard for Floating-Point Arithmetic, *IEEE Std* **754-2008** (2008) 1-70.

[24] D. Freedman, R. Pisani and R. Purves, *Statistics* (Fourth Edition), W. W. Norton and Company, 2007.

(Ana Cristina DĂSCĂLESCU, Radu BORIGA, Marius Iulian MIHĂILESCU) Department of Computer Science, Titu Maiorescu University, Vacaresti Avenue 187, Bucharest, Romania

E-mail address: `cristina.dascalescu@prof.utm.ro, radu.boriga@prof.utm.ro, marius.mihailescu@prof.utm.ro`