Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 848392, 10 pages
http://dx.doi.org/10.1155/2013/848392

Research Article

An Improved Secure Image Encryption Algorithm Based on Rubik's Cube Principle and Digital Chaotic Cipher

Adrian-Viorel Diaconu[1,2] and Khaled Loukhaoukha[3]

[1] *University Politehnica of Bucharest, ETTI Faculty, Bucharest 061071, Romania*
[2] *Lumina—The University of South-East Europe, IT & C Department, Bucharest 021187, Romania*
[3] *Laval University, Department of Electrical and Computer Engineering, QC, Canada G1K 7P4*

Correspondence should be addressed to Adrian-Viorel Diaconu; adrian.diaconu@lumina.org

Received 28 December 2012; Revised 22 January 2013; Accepted 4 February 2013

Academic Editor: Jun-Juh Yan

A recently proposed secure image encryption scheme has drawn attention to the limited security offered by chaos-based image encryption schemes (mainly due to their relatively small key space) proposing a highly robust approach, based on Rubik's cube principle. This paper aims to study a newly designed image cryptosystem that uses the Rubik's cube principle in conjunction with a digital chaotic cipher. Thus, the original image is shuffled on Rubik's cube principle (due to its proven confusion properties), and then XOR operator is applied to rows and columns of the scrambled image using a chaos-based cipher (due to its proven diffusion properties). Finally, the experimental results and security analysis show that the newly proposed image encryption scheme not only can achieve good encryption and perfect hiding ability but also can resist any cryptanalytic attacks (e.g., exhaustive attack, differential attack, statistical attack, etc.).

1. Introduction

In secure communication, image encryption schemes transform clear images into unintelligible others. The fundamental techniques used to encrypt a block of pixels are substitution and permutation.

Recent years' battle focuses on designing of highly robust encryption schemes (i.e., which provide good confusion and diffusion properties, to ensure coveted security factor), either using peculiar pixel shuffling methods [1–4], or using innovative digital chaos-based ciphers [5–25], or by making justified compositions between these different pixel shuffling and ciphering techniques [26, 27].

In this paper, we take advantage of results enforced in our past research articles [1, 28] (i.e., with images encryption schemes and chaotic ciphers generation), in order to design and analyze a new secure image encryption algorithm (based, in principle, on classic permutation-substitution architecture).

The rest of this paper is organized as follows. Section 2 presents the design of image cryptosystem (based on Rubik's cube principle and chaotic cipher). Section 3 makes detailed and comprehensive security assessments on the proposed scheme, while presenting experimental results. Finally, Section 4 concludes the work carried out.

2. Modified Rubik's Cube Image Encryption Algorithm

As the title suggests, this work is based on a previous work about images encryption algorithm, which at its turn is founded on Rubik's cube principle [1], but with slightly different approaches:

(1) on pixels' shuffling procedure:

   (a) rows and columns are alternatively shuffled (i.e., a row followed by a column, at a time),
   (b) each row's and column's circular-shifting direction and number of steps is derived from their intrinsic properties, using different modulo operators;

(2) on pixels' ciphering procedure:

(a) maintaining the rule of alternating rows and columns, the shuffled image is doubly ciphered, using two different ciphering matrices,

(b) ciphering matrices are built using bitstreams generated by chaotic systems, in conjunction with a multilevel discretization method [28].

In the following three subsections, we will describe the new modified encryption procedure.

2.1. Encryption Algorithm. With I_0 representing the pixels values matrix of an 8-bit gray level scale image of the size $m \times m$, the steps of newly proposed encryption algorithm are as follows.

(1) Randomly generate $\alpha_{\max}$, number of iterations, and β_1, β_2, rows' and columns' modulo operators; initialize α to zero.

(2) For $i = 1 : m$,

(a) compute the sum of all elements in row i

$$S_{\text{row\_}i} = \sum_{j=1}^{m} I_0(i, j), \tag{1}$$

(b) compute modulo β_1 of $S_{\text{row\_}i}$

$$M_{\text{row\_}i} = S_{\text{row\_}i} \pmod{\beta_1}, \tag{2}$$

(c) compute modulo 2 of $M_{\text{row\_}i}$

$$\omega_{\text{row\_}i} = M_{\text{row\_}i} \pmod 2, \tag{3}$$

(d) if $\omega_{\text{row\_}i} = 0 \rightarrow$ row i is right circular-shifted, with $M_{\text{row\_}i}$ steps; else, if $\omega_{\text{row\_}i} = 1 \rightarrow$ row i is left circular-shifted, with $M_{\text{row\_}i}$ steps

(e) compute the sum of all elements in column i

$$S_{\text{col\_}i} = \sum_{j=1}^{m} I_0(j, i), \tag{4}$$

(f) compute modulo β_2 of $S_{\text{col\_}i}$

$$M_{\text{col\_}i} = S_{\text{col\_}i} \pmod{\beta_2}, \tag{5}$$

(g) compute modulo 2 of $M_{\text{col\_}i}$

$$\omega_{\text{col\_}i} = M_{\text{col\_}i} \pmod 2, \tag{6}$$

(h) if $\omega_{\text{col\_}i} = 0 \rightarrow$ column i is up circular shifted, with $M_{\text{col\_}i}$ steps; else, if $\omega_{\text{col\_}i} = 1 \rightarrow$ column i is down circular shifted, with $M_{\text{col\_}i}$ steps.

(3) Increment α; if $\alpha \leq \alpha_{\max}$, go to the previous step; else, go to the next step.

Steps (2)-(3) will modify matrix I_0, generating a newly one denoted as I_{HVPS} (i.e., with horizontal and vertical pixels shuffled).

(4) Compute ciphering matrices $I_{\text{cipher\_col}}$ and $I_{\text{cipher\_row}}$

(5) For $i = 1 : m$,

(a) cipher row i

$$I_{\text{HVPS}}(i, :) = I_{\text{HVPS}}(i, :) \oplus I_{\text{cipher\_row}}(i, :), \tag{7}$$

(b) cipher column i

$$I_{\text{HVPS}}(:, i) = I_{\text{HVPS}}(:, i) \oplus I_{\text{cipher\_col}}(i, :)'. \tag{8}$$

Step (5) will modify the matrix I_{HVPS}, generating a newly one denoted as I_{ENC} (i.e., representing the pixels values matrix of the encrypted image).

2.2. Decryption Algorithm. With I_{ENC} representing the pixels values matrix of an 8-bit gray level scale encrypted image of size $m \times m$, with the correct key $K = (\alpha, \beta_1, \beta_2, x, r)$, the original image I_0 is recovered as follows.

(1) Compute ciphering matrices $I_{\text{cipher\_col}}$ and $I_{\text{cipher\_row}}$

(2) For $i = 1 : m$,

(a) decipher row i

$$I_{\text{ENC}}(i, :) = I_{\text{ENC}}(i, :) \oplus I_{\text{cipher\_row}}(i, :), \tag{9}$$

(b) decipher column i

$$I_{\text{ENC}}(:, i) = I_{\text{ENC}}(:, i) \oplus I_{\text{cipher\_col}}(i, :)'. \tag{10}$$

Step (2) will modify matrix I_{ENC}, generating a newly one denoted as I_{DEC} (i.e., representing the pixels values matrix of the deciphered image).

(3) Initialize α to zero.

(4) For $i = 1 : m$,

(a) compute the sum of all elements in column i

$$S_{\text{col\_}i} = \sum_{j=1}^{m} I_{\text{DEC}}(j, (m + 1 - i)), \tag{11}$$

(b) compute modulo β_2 of $S_{\text{col\_}i}$

$$M_{\text{col\_}i} = S_{\text{col\_}i} \pmod{\beta_2}, \tag{12}$$

(c) compute modulo 2 of $M_{\text{col\_}i}$

$$\omega_{\text{col\_}i} = M_{\text{col\_}i} \pmod 2, \tag{13}$$

(d) if $\omega_{\text{col\_}i} = 1 \rightarrow$ column i is up circular shifted, with $M_{\text{col\_}i}$ steps; else, if $\omega_{\text{col\_}i} = 0 \rightarrow$ column i is down circular shifted, with $M_{\text{col\_}i}$ steps,

(e) compute the sum of all elements in row i

$$S_{\text{row\_}i} = \sum_{j=1}^{m} I_{\text{DEC}}\left((m+1-i), j\right), \tag{14}$$

(f) compute modulo β_1 of $S_{\text{row\_}i}$

$$M_{\text{row\_}i} = S_{\text{row\_}i} \left(\text{mod } \beta_1\right), \tag{15}$$

(g) compute modulo 2 of $M_{\text{row\_}i}$

$$\omega_{\text{row\_}i} = M_{\text{row\_}i} \left(\text{mod } 2\right), \tag{16}$$

(h) if $\omega_{\text{row\_}i} = 1 \rightarrow$ row i is right circular shifted, with $M_{\text{row\_}i}$ steps; else, if $\omega_{\text{row\_}i} = 0 \rightarrow$ row i is left circular shifted, with $M_{\text{row\_}i}$ steps.

(5) Increment α; if $\alpha \leq \alpha_{\max}$, go to the previous step; else, the decryption process is done.

Steps (4)-(5) will modify the matrix I_{DEC}, generating a new one denoted as I_0 (i.e., representing the pixels values matrix of the deshuffled image ~ the original image).

2.3. Ciphering Matrices Computation. Given any chaotic map (e.g., a tent map (17), denoted as f_T), using random sequences of real numbers generated by its orbits in conjunction with a multilevel discretization method [28] (e.g., with four thresholds, i.e., 2-bit encoding of each interval), resulted di-bits are spread into two separate files (i.e., "Bits_A.txt," containing di-bit's first bit; resp., "Bits_B.txt," containing di-bit's second bit). Consider

$$\begin{aligned} f_T &: [0,1] \longrightarrow [0,1], \\ f_T(x) &= r\left(1 - |1 - 2x|\right), \quad r \in (0,1) \end{aligned} \tag{17}$$

$m \cdot m \cdot 8$ di-bit pairs have been generated (i.e., 2.097.152 bits were written in each file), this number being, as seen, directly proportional to the image dimensions.

f_T's initial seeding points were $r = 1.99831378796143$ and $x = 0.687754925117697$.

Under the previous circumstances, ciphering matrices are computed as follows.

(1) Open and read "Bits_A.txt" and "Bits_B.txt" files; initialize a temporary counter C to zero.

(2) Initialize $I_{\text{cipher\_col}}$ and $I_{\text{cipher\_row}}$, where

$$I_{\text{cipher\_col}} = I_{\text{cipher\_row}} = \text{zeros}\ (m, m). \tag{18}$$

(3) For $i = 1 : m$, for $j = 1 : m$,

(a) take eight consecutive bits from each file

$$\begin{aligned} &\text{Byte\_A} = \text{strcat}\left(\text{Bits\_A.txt}\left(C + k\right)\right), \quad k = \overline{1,8}, \\ &\text{Byte\_B} = \text{strcat}\left(\text{Bits\_B.txt}\left(C + k\right)\right), \quad k = \overline{1,8}; \end{aligned} \tag{19}$$

(b) update $I_{\text{cipher\_col}}$ and $I_{\text{cipher\_row}}$

$$\begin{aligned} I_{\text{cipher\_row}}(i,j) &= \text{bin2dec}\left(\text{Byte\_A}\right), \\ I_{\text{cipher\_col}}(i,j) &= \text{bin2dec}\left(\text{Byte\_B}\right); \end{aligned} \tag{20}$$

(c) update the temporary counter

$$\text{C} = \text{C} + 8. \tag{21}$$

Steps (1)–(3) will produce the ciphering matrices.

3. Experimental Results

In order to assess the efficiency and security of the newly resulted image encryption algorithm, various analyses were conducted (e.g., statistical, strength against some cryptographic attacks, etc.). This section deals with the presentation of tests and observations over the results achieved (performances).

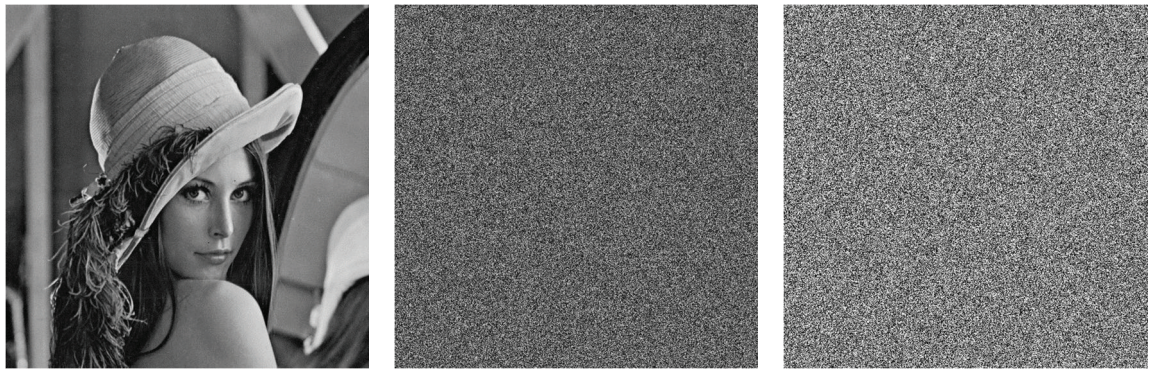
3.1. Visual Testing. The purpose of visual testing is to highlight presence of similarities between plain-image and shuffled and (or) ciphered image (i.e., if the scrambled and (or) ciphered image does or does not contain any features of the plain-image).

Visual testing was performed on the 512×512 pixels, 8-bit, gray-level, Lena standard test image. Figure 1 depicts the test image (a), along with its shuffled (b) and ciphered (c) versions. By comparing them, one can say that there is no perceptual similarity (i.e., no visual information can be observed in processed versions of plain-image).

3.2. Security Assessment by Statistical Analysis. The main two statistical analyses, generally performed when it comes to showcase confusion and diffusion properties of an image shuffling and (or) ciphering algorithm, are the histograms analysis and analysis of the correlation coefficient between adjacent pixels. [1, 2, 5–8, 26].

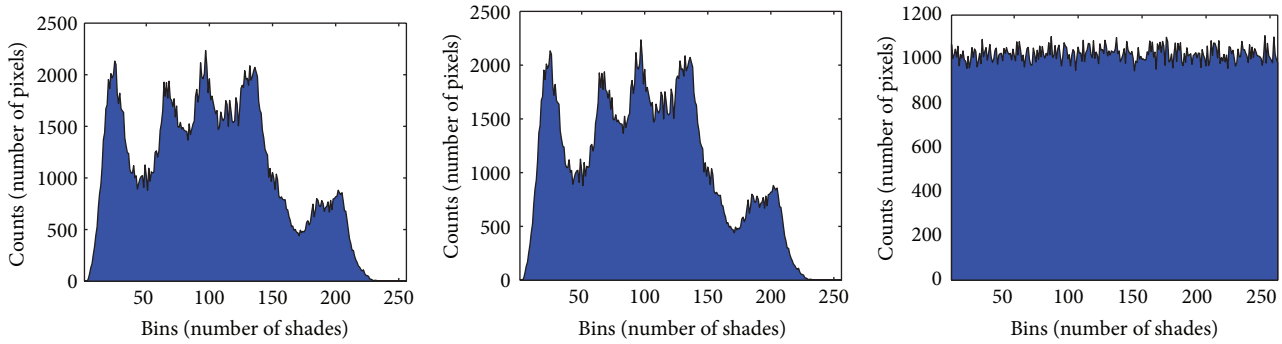
3.2.1. Histogram Analysis. Histogram analysis depicts pixels' distribution within an image, by representing their number relative to each intensity level. Figure 2 represents histograms of original (a), shuffled (b), and ciphered (c) images. As expected, although a superior diffusion effect is achieved through pixels' shuffling procedure, image histogram is not modified thus making it vulnerable to statistical attacks. But after algorithm's second phase (i.e., ciphering procedure) completion, the histogram gains a uniform distribution, meaningfully different than the one of original image (which contains large sharp rises followed by sharp declines), and thus with no statistical similarity in appearance, resulted image does not provide any clue for statistical attacks.

3.2.2. Adjacent Pixels Correlation Coefficients. It is well known that, generally in plain-images, any arbitrarily chosen pixel is strongly correlated with its adjacent pixels (either they are diagonally, vertically, or horizontally oriented) [3, 9].



(a) (b) (c)

Figure 1: Original, shuffled, and encrypted images. (a) Original image, (b) shuffled image and (c) encrypted image. (Test conditions: $\alpha = 10$, $\beta_1 = 167$, $\beta_2 = 202$, $r = 1.998313787961430$, and $x = 0.687754925117697$.)



(a) (b) (c)

Figure 2: Histograms of original, shuffled, and encrypted images. (a) Histogram of original image, (b) histogram of shuffled image, and (c) histogram of encrypted image. (Test conditions: $\alpha = 10$, $\beta_1 = 167$, $\beta_2 = 202$, $r = 1.998313787961430$, and $x = 0.687754925117697$.)

Figure 3 shows the correlation distributions of the horizontal adjacent pixels for the original (a), shuffled (b), and encrypted images (c), while the correlation distributions of the vertically and diagonally adjacent pixels are shown in Figure 4, respectively, Figure 5. At the same time, all the correlation coefficient values (computed over 10.000 pairs of adjacent pixels, randomly selected, for each of the testing directions) are summarized in Table 1. It can be easily noticed that neighboring pixels in the original image are highly correlated; that is, correlation coefficient values are too high, very close to one. In contrary, in cases of shuffled and encrypted images, those values are close to zero, meaning that all neighboring pixels considered in tests are weakly correlated, which is the expected result [4, 10–12, 29].

Overall, results are with an order of better magnitude, compared with results of other proposed algorithms, as proven in [13–16].

3.3. Security Assessment by Differential Analysis. Differential cryptanalysis assumes that the attacker is able to create small changes in the input plain-image and examine outputs (i.e., original image's processed versions). In doing this, shuffling and (or) encryption key(s) and (or) the meaningful relationship between original images and its shuffled or encrypted versions can be derived. Therefore, a desirable property of the proposed modified algorithm is to be sensitive to the small changes in plain-image. Thus, the differential attack can actually lose its efficiency and become practically useless, if one small change in the original image can cause a significant change in its processed versions.

Table 1: Correlation coefficients of adjacent pairs of pixels.

Image	Adjacent pixels orientation		
	Vertical	Horizontal	Diagonal
Original	0.9696	0.9820	0.9577
Shuffled	0.0019	0.0090	0.0104
Encrypted	0.0002	0.0006	0.0043

Security assessment by differential analysis is based on three measures: NPCR (number of pixels change rate), UACI (unified average changing intensity), and MAE (i.e., mean absolute error). In order to approach the performances of ideal image encryption algorithms, MAE and NPCR must be as large as possible (i.e., close to unity, at least for NPCR), while UACI values must be around 33% [1, 2, 5, 7].

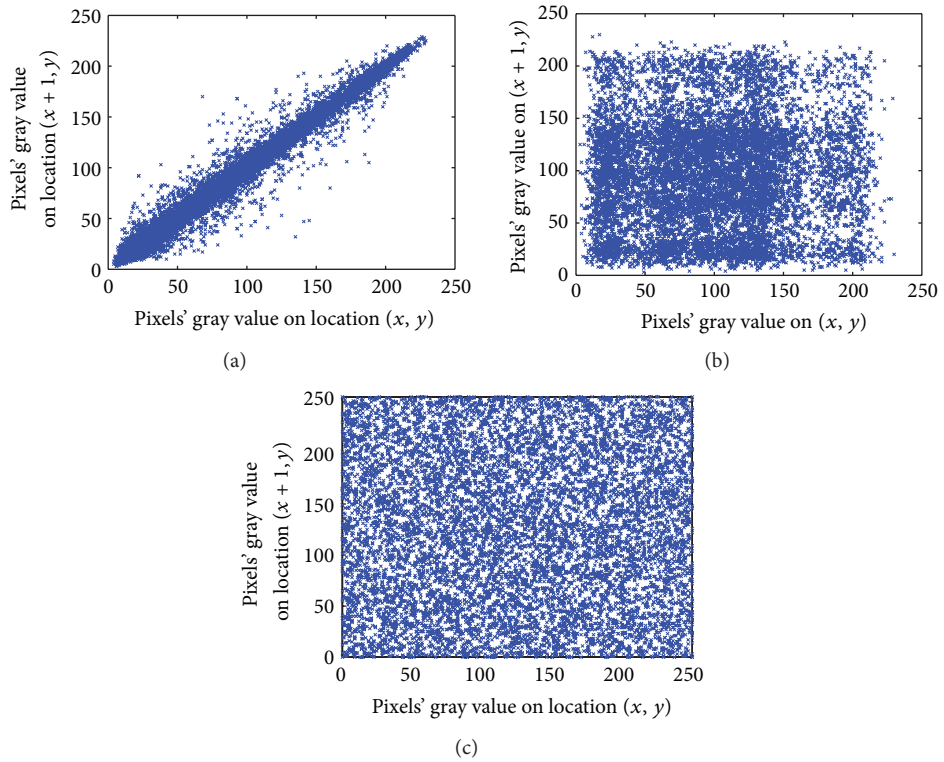


Figure 3: Correlation distribution of horizontally adjacent pixels. (a) Original image, (b) shuffled image, and (c) encrypted image. (Test conditions: $\alpha = 10$, $\beta_1 = 167$, $\beta_2 = 202$, $r = 1.998313787961430$, and $x = 0.687754925117697$.)

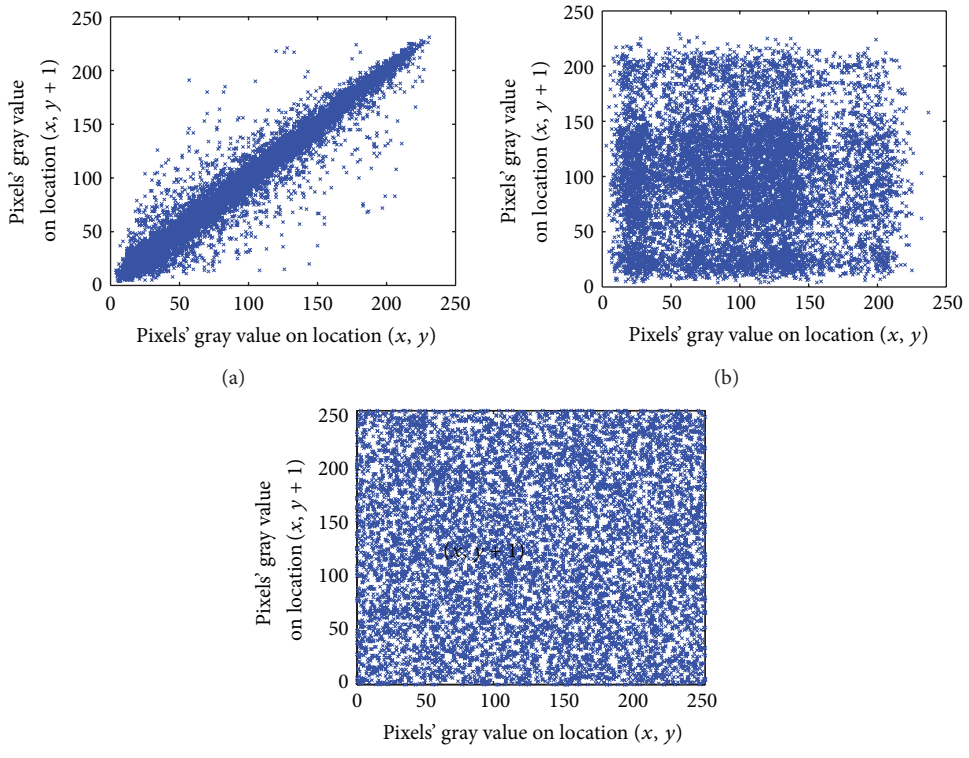


Figure 4: Correlation distribution of vertically adjacent pixels. (a) Original image, (b) shuffled image, and (c) encrypted image. (Test conditions: $\alpha = 10$, $\beta_1 = 167$, $\beta_2 = 202$, $r = 1.998313787961430$, and $x = 0.687754925117697$.)

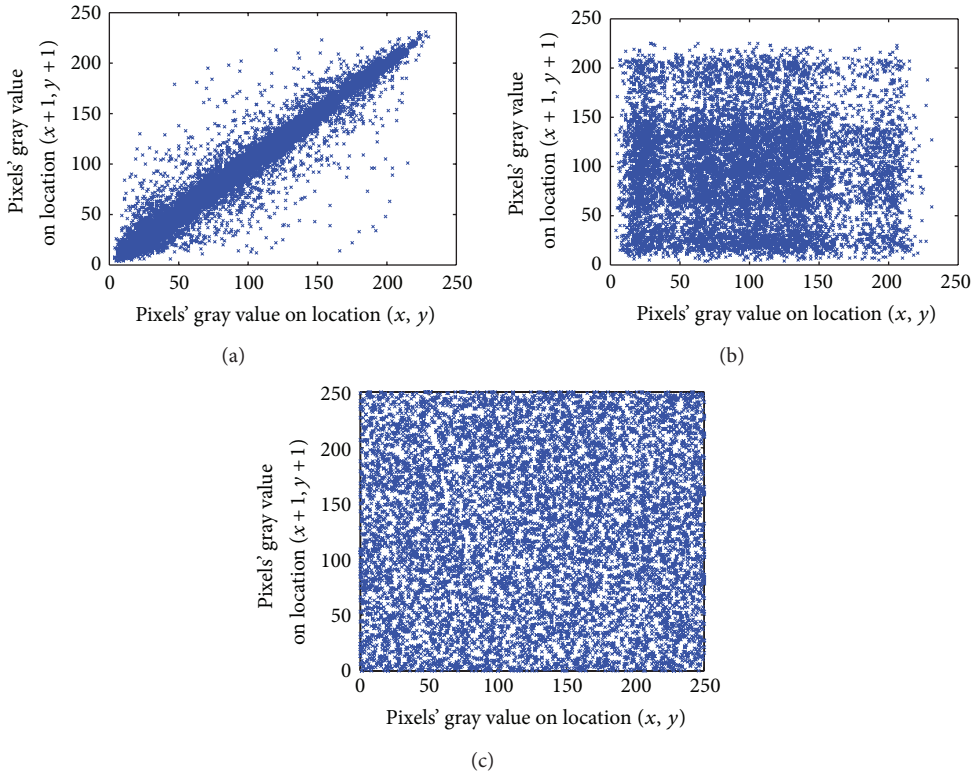


FIGURE 5: Correlation distribution of diagonally adjacent pixels. (a) Original image, (b) shuffled image, and (c) encrypted image. (Test conditions: $\alpha = 10$, $\beta_1 = 167$, $\beta_2 = 202$, $r = 1.998313787961430$, and $x = 0.687754925117697$.)

NPCR, UACI, and MAE values, summarized in Table 2, demonstrate that swiftly changes in the original image will result in negligible changes in its shuffled and encrypted versions and also that choosing higher values of α, β_1, and β_2 will result in higher values of these coefficients, thus a better strength against any differential cryptanalysis.

Overall, results are, on average, about 1.29%–28.09% (in case of UACI), 62.61%–78.11% (in case of NPCR), and over 42.89% (in case of MAE) better, compared with results of other proposed algorithms [13–16] for the following test conditions: $\alpha, \beta_1, \beta_2 \in [10^2, 10^3]$. Similar results, especially in terms of NPCR's value, are proven in [1, 5].

3.4. Security Assessment by Entropy Analysis. An algorithm for encryption of images should give a ciphered image having equiprobable gray levels (i.e., the entropy of this ciphered image should be, at least theoretically, equal to 8 bits, for gray-scale images of 256 levels). Actually, in practice, the resulted entropy is smaller than the ideal one. The smaller the resulted entropy, the greater the degree of predictability, a fact which threatens encryption system's security.

Computing the entropy value of the encrypted image, which is very close to the ideal value of 8 Sh (more accurately 7.9992 Sh, much more closer than values resulted under different algorithms [21–23, 30]), we can say that the proposed encryption algorithm is highly robust against entropy attacks.

3.5. Security Assessment by Key Analysis. This security assessment targets the key space and sensibility.

3.5.1. Key Sensibility. In order to approach the performances of an ideal image encryption algorithm, the proposed one should have high sensibility to encryption key (i.e., any small change in the key should lead to a significant change in the shuffled and encrypted or deshuffled and decrypted images).

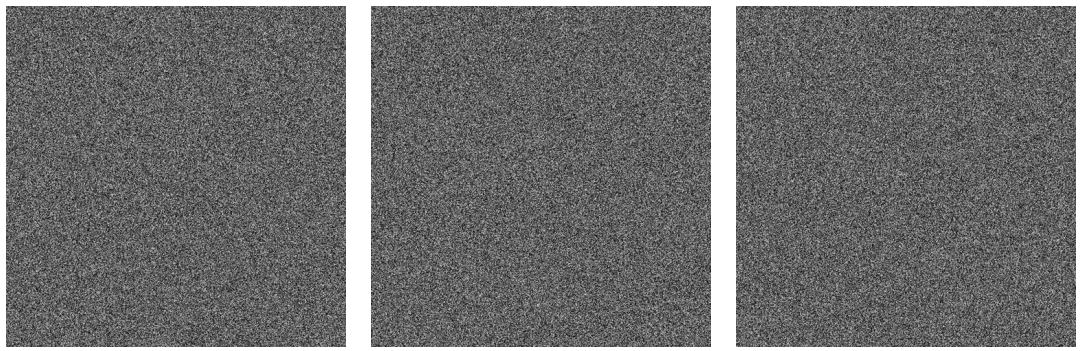
For the newly proposed scheme the encryption key is $K = (\alpha, \beta_1, \beta_2, x, r)$. Various tests were performed in order to highlight the impact of key changes in the image deshuffling and decryption processes. Results are presented in Figures 6, 7, and 8. Clearly, deshuffling and (or) decryption using wrong key does not succeed.

3.5.2. Key Space. A large key space is very important for an encryption algorithm to repel the brute-force attack. In theory, the proposed scheme can accommodate an infinite key space, taking into account α, β_1, and β_2; but, due to actual computational limitations, α, β_1, and β_2 key elements are restrained to 64-bit representation, and, at first sight, the key space size is 2^{192}. Furthermore, the two seed points of the chaotic generator (viz., x and r), with 10^{-15} precision, extend the key space by 10^{30}. Therefore, a total 10^{88} key space is large enough to face an exhaustive attack [6, 17–20, 27].

3.6. Security Assessment by Different Attacks. Few attacks, such as cropping and additive noise, can be performed by an attacker who intercepts encrypted image, in order to modify it so that, after decryption, the legitimate user cannot understand and (or) use the original message (i.e., in our case, the image).

Table 2: Difference measures between original and shuffled or encrypted images.

Image	NPCR [%]	UACI [%]	MAE [%]	Test conditions		
				α	β_1	β_2
Shuffled	$\ll 80$	$\ll 5$	$\ll 10$	1	$0 < \beta_1 < 10$	$0 < \beta_2 < 10$
	97.7104	13.3053	33.9286		$10 \leq \cdots \leq 10^2$	$10 \leq \cdots \leq 10^2$
	99.2668	22.7348	57.9737		$10^2 < \cdots \leq 10^3$	$10^2 < \cdots \leq 10^3$
	99.3950	23.4347	59.7585		$10^3 < \cdots \leq 10^4$	$10^3 < \cdots \leq 10^4$
	99.4068	23.4454	59.7858		$10^4 < \cdots \leq 10^5$	$10^4 < \cdots \leq 10^5$
	99.4476	23.7236	60.4951	$10 < \alpha \leq 10^2$	$10^2 < \cdots \leq 10^3$	$10^2 < \cdots \leq 10^3$
	99.4541	23.7657	60.6026	$10^2 < \cdots \leq 10^3$		
	99.4629	23.7697	60.7253	$10^3 < \cdots \leq 10^4$		
	99.4673	23.7701	60.7284	$10^4 < \cdots \leq 10^5$		
Encrypted	99.6078	30.5903	78.0053	1	$10^2 < \cdots \leq 10^3$	$10^2 < \cdots \leq 10^3$
	99.6090	30.5699	77.9533	$10 < \alpha \leq 10^2$		
	99.6120	30.5997	78.0292	$10^2 < \cdots \leq 10^3$		



(a) (b) (c)

Figure 6: Algorithm key sensibility image deshuffled using (a) wrong key β_1 (+1 LSB variation), (b) wrong key β_2 (−1 LSB variation), (c) wrong keys β_1, β_2 (±1 LSB variations). (Test conditions: image successfully decrypted; $r = 1.998313787961430$, $x = 0.687754925117697$, and $\alpha = 10$ known.)

The following two subsections aim to analyze proposed scheme's performances against such attacks.

3.6.1. Additive Noise Attack. An additive noise attack consists in adding random noise to the intercepted encrypted image. Two types of noises are considered: salt and pepper noise and speckle noise. To measure robustness of the proposed image encryption scheme against these attacks, MSE measure is used.

Figures 9 and 10 illustrate decrypted images, in cases where their encrypted version was attacked by salt and pepper noise with 0.0001 densities (a) and by speckle noise with 0.0001 variance (b), but under different test conditions.

As expected, due to the fact that each column's and row's intrinsic properties were used to compute the number of their circular shifts, the proposed image encryption scheme has no immunity to this kind of attacks, statement strengthened by MSE measure's values ($2.515 \cdot 10^3$ in Figure 9(a) and $2.372 \cdot 10^3$ in Figure 9(b), resp., $5.599 \cdot 10^3$ in Figure 10(a) and $5.697 \cdot 10^3$ in Figure 10(b)).

3.6.2. Cropping Attacks. Cropping attacks consist of modifying the intercepted encrypted image by deleting one, or several of its areas. To measure robustness of proposed encryption scheme against this kind of attacks, MSE measure is used also.

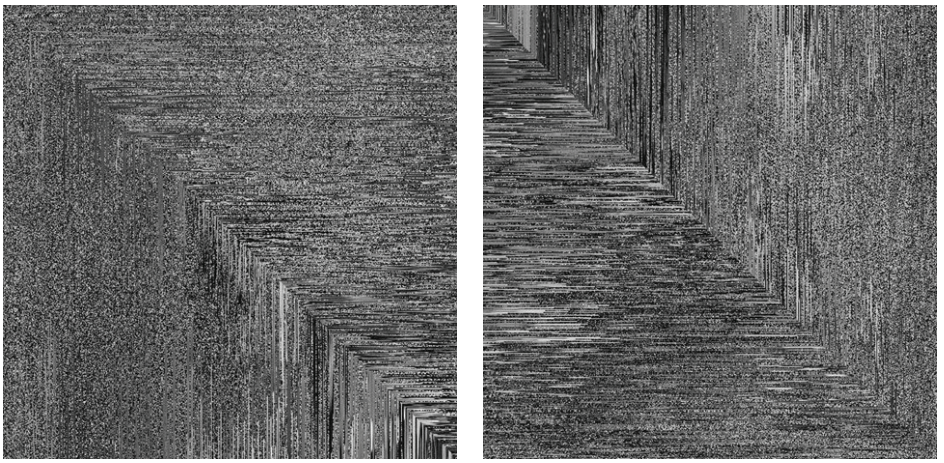
Figures 9 and 10 illustrate decrypted image, in cases where their encrypted version was attacked by cropping one 1/8 of its area (c).

Same problem is raised by the shuffling procedure mode of operation, and proposed image encryption scheme cannot handle such type of attacks, statement strengthened by MSE measure's values ($3.031 \cdot 10^3$ in Figure 9(c) and $6.043 \cdot 10^3$ in Figure 10(c)).

From the previous results, we can conclude that random noise attacks, seriously affect decrypted images. As a proposal, if it is known that the proposed encryption scheme will be used within a communication system susceptible to perturbations, algorithms (or systems) to ensure data integrity should be used, as the one shown in [31].

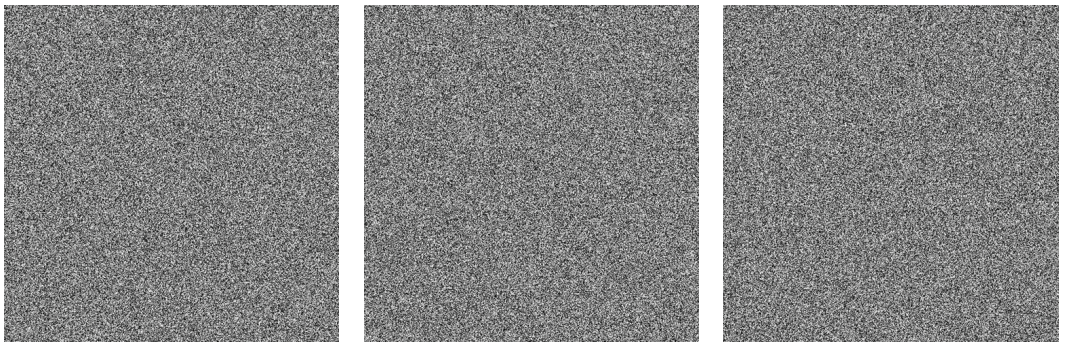
3.7. Speed Test. Another important matter taken into account when new image encryption techniques are proposed is the actual algorithm execution speed.

Regardless of β_1 and β_2 variations, shuffling time is of 0.2158 s while the deshuffling time is of 0.1987 s. Obviously,



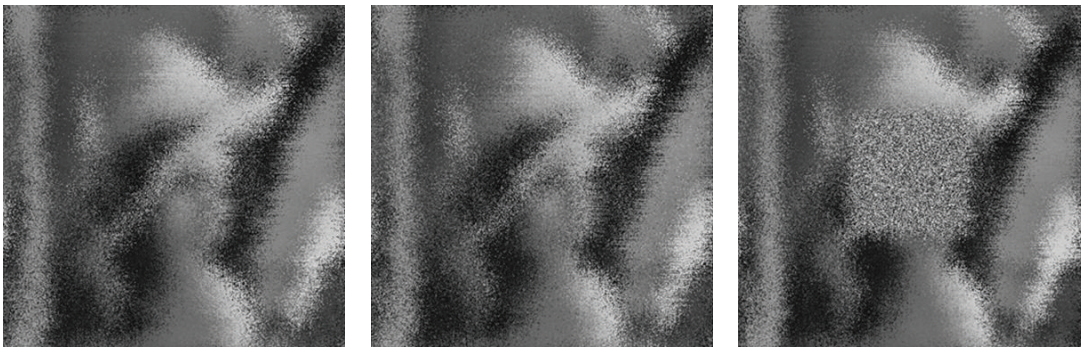
(a) (b)

Figure 7: Algorithm key sensibility image deshuffled using (a) wrong number of iterations ($\alpha + 1$ LSB variation) and (b) wrong number of iterations ($\alpha - 1$ LSB variation). (Test conditions: image successfully decrypted; $r = 1.998313787961430$, $x = 0.687754925117697$, $\beta_1 = 167$, & $\beta_2 = 202$ known.)



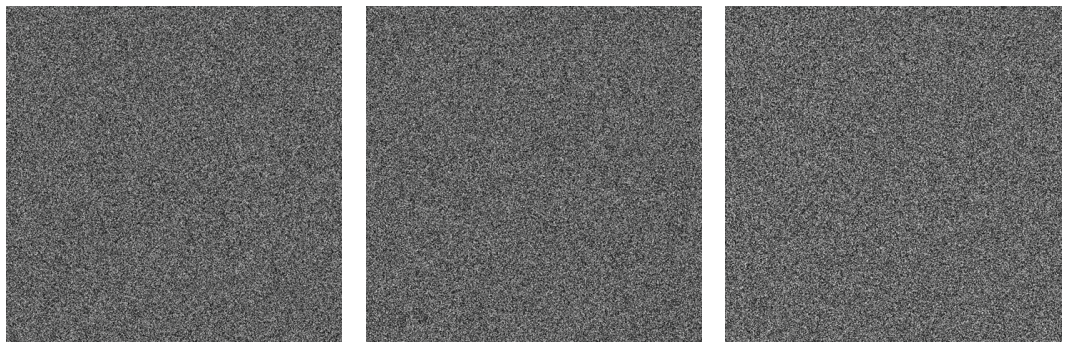
(a) (b) (c)

Figure 8: Algorithm key sensibility image decrypted using (a) wrong key x ($\pm 1 \cdot 10^{-15}$ variations), (b) wrong key r ($\pm 1 \cdot 10^{-15}$ variations), and (c) wrong keys x and r ($\pm 1 \cdot 10^{-15}$ variations).



(a) (b) (c)

Figure 9: Decrypted images (a) under salt and pepper noise attack, (b) under speckle noise attack, and (c) under cropping attack. (Test conditions: $\alpha = 3$, $\beta_1 = 10$, and $\beta_2 = 10$.)



(a) (b) (c)

Figure 10: Decrypted images (a) under salt and pepper noise attack, (b) under speckle noise attack, and (c) under cropping attack. (Test conditions: $\alpha = 125$, $\beta_1 = 107$, and $\beta_2 = 202$.)

these timings are increasing proportionally with α number of iterations. With an actual ciphering time of 0.1826 s and a deciphering time of 0.1701 s, proposed scheme's encryption and decryption speeds are smaller than 0.40 s (more accurately, 0.3984 s and 0.3688 s). Achieved speed is superior, in comparison with those of other algorithms [1, 5, 13–16].

The speed test was performed on the 512×512 pixels, 8-bit, gray-level, Lena standard test image.

All the previous results (i.e., numerical results presented in the entire Section 3) were generated with proposed encryption algorithm's scripts for (de)shuffling and (de)ciphering written on MATLAB 7.3.0 and ran on an Intel Pentium Dual CPU T3200 @ 2.00 GHz personal computer.

4. Conclusions

This work develops novel permutation—substitution image encryption architecture, based on Rubik's cube principle and digital chaotic cipher.

The proposed encryption system includes two major parts, chaotic pixels substitution (in order to achieve desired diffusion factor) and Rubik's cube, principle based, pixels permutation (in order to achieve desired confusion factor).

Different keys were used for shuffling and ciphering procedures, and while a tent map was used to generate image's ciphering matrices, each row's and column's intrinsic properties were used to compute the number of their circular shifts.

Comprehensive experimental tests have been carried out, and numerical analyses have shown robustness of the proposed algorithm against several cryptanalytic attacks. Likewise, the performance assessment tests attest that the proposed image encryption scheme is fast and highly secure. Although a much smaller key space is used, but still large enough to face against exhaustive attack, with a smaller key size, the proposed encryption scheme presents better results, compared to those of previously proposed ones.

As future work, an actual implementation on FPGA (focused on the optimization of proposed algorithm, for parallel computing) as well as the investigation of lossless image compression and (or) noise protection schemes (i.e., taking into consideration shortcoming of "how to use" proposal [31]) is concerned.

References

[1] K. Loukhaoukha, J.-Y. Chouinard, and A. Berdai, "A secure image encryption algorithm based on Rubik's cube principle," *Journal of Electrical and Computer Engineering*, vol. 2012, Article ID 173931, 13 pages, 2012.

[2] S. Rakesh, A. A. Kaller, B. C. Shadakshari et al., "Image encryption using block based uniform scrambling and chaotic logistic mapping," *International Journal on Cryptography and Information Security*, vol. 2, no. 1, pp. 49–57, 2012.

[3] S. Borujeni and M. Eshghi, "Design and simulation of encryption system based on PRNG and tompkins-paige permutation algorithm using VHDL," in *Proceedings of the International Conference on Robotics, Vision, Information and Signal Processing*, pp. 63–67, Penang, Malaysia, November 2007.

[4] A. Mitra, Y. V. Subba Rao, S. R. M. Prasanna et al., "A new image encryption approach using combinational permutation techniques," *International Journal of Electrical and Computer Engineering*, vol. 1, no. 2, pp. 127–131, 2006.

[5] A. Jolfael and A. Mirghadri, "An image encryption approach using chaos and stream cipher," *Journal of Theoretical and Applied Information Technology*, vol. 19, no. 2, pp. 117–125, 2010.

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

[7] S. Etemadi Borujeni and M. Eshghi, "Chaotic image encryption design using Tompkins-Paige algorithm," *Mathematical Problems in Engineering*, vol. 2009, Article ID 762652, 22 pages, 2009.

[8] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[9] A. N. Pisarchik and M. Zanin, "Image encryption with chaotically coupled chaotic maps," *Physica D*, vol. 237, no. 20, pp. 2638–2648, 2008.

[10] H.-P. Xiao and G.-J. Zhag, "An image encryption scheme based on chaotic systems," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 2707–2711, Dalian, China, August 2006.

[11] J. Zou, D. Qi, R. K. Ward et al., "The application of chaotic maps in image encryption," in *Proceedings of the 3rd International*

IEEE Northeast Workshop on Circuits and Systems, pp. 331–334, Quebec City, Canada, June 2005.

[12] S. Lian, J. Sun, Z. Wang et al., "Security analysis of a chaos-based image encryption algorithm," *Physica A*, vol. 351, no. 2–4, pp. 645–661, 2005.

[13] Y. Mao, G. Chen, and S. Lian, "A novel fast image encryption scheme based on 3D chaotic Baker maps," *International Journal of Bifurcation and Chaos*, vol. 14, no. 10, pp. 3613–3624, 2004.

[14] L. Zhang, X. Liao, and X. Wang, "An image encryption approach based on chaotic maps," *Chaos, Solitons & Fractals*, vol. 24, no. 3, pp. 759–765, 2005.

[15] H. Gao, Y. Zhang, S. Liang et al., "A new chaotic algorithm for image encryption," *Chaos, Solitons & Fractals*, vol. 29, no. 2, pp. 393–399, 2006.

[16] F. Zhou, G. Cao, and B. Li, "Design of digital image encryption algorithm based on compound chaotic system," *Journal of Harbin Institute of Technology*, vol. 14, no. 2, pp. 30–33, 2007.

[17] Z. Liu, L. Xu, C. Lin et al., "Image encryption scheme by using random phase encoding in gyrator transform domains," *Optics and Lasers in Engineering*, vol. 49, no. 4, pp. 542–546, 2011.

[18] C. K. Huang, H.-H Nien, S.-K. Changchien et al., "Image encryption with chaotic random codes by grey relational grade and Taguchi method," *Optics Communications*, vol. 280, no. 2, pp. 300–310, 2007.

[19] S. Mazloom and A. M. Efekhari-Moghadam, "Color image encryption based on coupled nonlinear chaotic map," *Chaos, Solitons & Fractals*, vol. 42, no. 3, pp. 1745–1754, 2009.

[20] Y. Tang, Z. Wang, and J.-A. Fang, "Image encryption using chaotic coupled map lattices with time-varying delays," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2456–2468, 2010.

[21] K.-W. Wong, S.-W. Ho, and C.-K. Yung, "A chaotic cryptography scheme for generating short ciphertext," *Physics Letters A*, vol. 310, no. 1, pp. 67–73, 2003.

[22] T. Xiang, X. Liao, G. Tang et al., "A novel block cryptosystem based on iterating a chaotic map," *Physics Letters A*, vol. 349, no. 1–4, pp. 109–115, 2006.

[23] Z. Lin and H. Wang, "Efficient image encryption using a chaos-based PWL memristor," *IETE Technical Review*, vol. 27, no. 4, pp. 318–325, 2010.

[24] D. A. Cristina, B. Radu, and R. Ciprian, "A new pseudorandom bit generator using compounded chaotic tent maps," in *Proceedings of the 9th International Conference on Communications (COMM '12)*, pp. 339–342, Bucharest, Romania, June 2012.

[25] B. Radu and D. A. Cristina, "A novel pseudo-random bit generator based on some transcendental chaotic systems," *Ovidius University Annals*, vol. 11, no. 1, pp. 208–212, 2011.

[26] A. Musheer and F. Omar, "A multi-level blocks scrambling based chaotic image cipher," in *Proceedings of the 3rd International Conference on Contemporary Computing (IC3 '10)*, pp. 171–182, Noida, India, August 2010.

[27] C. K. Huang and H. H. Nien, "Multi chaotic system based pixel shuffle for image encryption," *Optics Communications*, vol. 282, no. 11, pp. 2123–2127, 2009.

[28] A.-V. Diaconu, "Multiple bitstreams generation using chaotic sequences," *The Annals of "Dunarea De Jos" University of Galati—Fascicle III*, vol. 35, no. 1, pp. 37–42, 2012.

[29] D. R. Stinson, *Cryptography: Theory and Practice*, CRC Press Series on Discrete Mathematics and Its Applications, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2nd edition, 2002.

[30] M. S. Baptista, "Cryptography with chaos," *Physics Letters A*, vol. 240, no. 1-2, pp. 50–54, 1998.

[31] A.-V. Diaconu, "Hardware implementation, on a FPGA, of Golay (23.12.7) error-correcting code—for usage within WSNs," *University of Pitești Scientific Bulletin*, vol. 12, no. 1, pp. 25–34, 2012.