Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Image encryption using block cipher and chaotic sequences ☆

José A.P. Artiles *, Daniel P.B. Chaves, Cecilio Pimentel

Department of Electronics and Systems, Federal University of Pernambuco, Recife, PE, 50711-970, Brazil

ARTICLE INFO

Keywords:
Advanced encryption standard
Block cipher
Chaotic maps
Image encryption
Security

ABSTRACT

Encryption of data with high correlation, such as images, is a challenge for block ciphers, since patterns of the original image may remain after encryption. This is due to the deterministic mapping performed by the cipher. To overcome this limitation, a block cipher is used in an adequate mode of operation, such as cipher block chaining, counter mode. It is presented in this work randomized block ciphers inspired by the Rijndael architecture employing chaotic maps as an entropy source. It is shown that the proposal achieves good security and robustness indicators with fewer rounds compared to that obtained with the Rijndael algorithm.

1. Introduction

The National Institute of Standards and Technology (NIST) selected the Rijndael algorithm with block size of 128 bits, the Advanced Encryption Standard (AES), as its current recommendation for the symmetric key encryption algorithm [1]. Rijndael supports block and key sizes of 128, 192, and 256 bits and the number of rounds may be 10, 12, or 14, depending on the block and key sizes. This cipher has four units in each round: SubBytes, ShiftRows, MixColumns, AddRoundKey [2]. The SubBytes unit provides confusion in the ciphertext [3] and is implemented by S-boxes. An S-box performs a bijective mapping from an input byte to an output byte defined by operations in Galois field GF(2^8).

The application of Rijndael directly as an encryption unit is equivalent to the electronic code book (ECB) mode of operation. There are some weaknesses associated with this mode, since encryption is performed deterministically [2]. This means that identical plaintext blocks are encrypted into identical ciphertext blocks when the same key is used. An ECB-encrypted ciphertext leaks information about repetitive patterns in the plaintext. This feature can be avoided by using probabilistic (or nondeterministic) modes of operation [4], such as, cipher block chaining (CBC), cipher feedback (CFB), output feedback (OFB), counter mode (CTR). These nondeterministic modes of operation are secure under chosen-plaintext attack and allow efficient and fast software implementation [5]. The analysis of an image encryption scheme based on the CBC is conducted in [6].

Several chaos-based image encryption algorithms have been recently proposed [7–14]. Chaos has suitable properties for image encryption, such as, aperiodicity, ergodicity, sensibility to initial conditions [15]. Chaotic maps are used to design some transformations like scrambling, permutation. These maps are also employed to generate pseudo-random numbers used in encryption algorithms. However, by exploiting the structure of the algorithms some chaos-based image encryption schemes have been broken by either differential or known/chosen plaintext attacks, see for example [5,16–18].

The objective of this work is to propose symmetric key cipher architectures based on a randomized version of the Rijndael S-box in order to encrypt images using the ECB mode of operation. These are less complex alternatives to the AES modes of operation. In this context, there are three main contributions in this article: (i) propose a technique to add a source of entropy to the S-boxes employing chaotic maps in a way compatible with the original Rijndael algorithm; (ii) show that the proposal achieves with fewer rounds the same level of security of the AES modes of operation indicated by a series of security metrics; (iii) propose alternatives to the diffusion layer of the Rijndael that speed up the encryption process without compromising security.

The rest of this article is organized into four sections. In Section 2, the AES algorithm and the chaotic maps used in this work are described. Some security metrics commonly used in image encryption are revised in Section 3. In Section 4, the randomized S-box by a chaotic map is introduced and its security and time complexity are analyzed. A simplified algorithm is also proposed in this section. The conclusions of this work are presented in Section 5.

2. Preliminares

We consider the structure of the Rijndael algorithm adopted by AES with blocks of size 128 bits. Moreover, we consider 128-bit key and then 10 rounds. The encryption process begins by converting

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.image.2019.08.014.

* Corresponding author.

E-mail addresses: joseantonio.artiles@ufpe.br (J.A.P. Artiles), daniel.chaves@ufpe.br (D.P.B. Chaves), cecilio@ufpe.br (C. Pimentel).

https://doi.org/10.1016/j.image.2019.08.014
Received 27 February 2019; Received in revised form 20 May 2019; Accepted 31 August 2019
Available online 4 September 2019

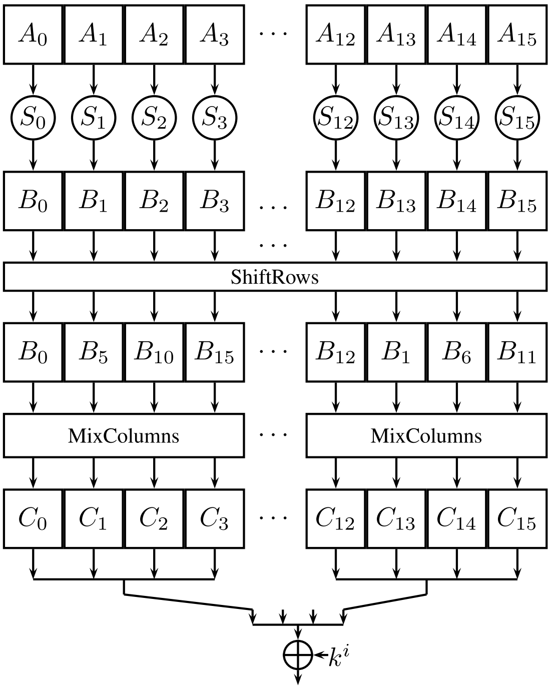

Fig. 1. Block diagram of the AES algorithm per round.

the information block of 128 bits into a 4 × 4 array, denoted by state matrix, where each element is a byte. Similarly, the key is also converted into a 4 × 4 array. The four AES units, shown in the block diagram of Fig. 1, are briefly described next.

(1) SubBytes: This unit has 16 identical S-boxes, denoted by $S_0, \dots, S_{15}$ in Fig. 1, which operate in parallel, with input byte A_i and output byte B_i. In each S-Box, one byte of the state matrix is replaced by one byte obtained by operations in GF (2^8) generated by the primitive polynomial $P(x) = x^8 + x^4 + x^3 + x + 1$. For the ith S-box, the input and output bytes are related as $B_i = S(A_i) = R \times A_i^{-1} + Q$, where R is a matrix, A_i^{-1} is the multiplicative inverse of A_i in GF (2^8) and Q is a non-zero constant vector.

(2) ShiftRows: This unit cyclically shifts the ith row of the state matrix at the output of the SubBytes unit by i bytes to the left, $i = 0, 1, 2, 3$.

(3) MixColumns: This process combines the columns of the state matrix using a linear transformation. This is the major diffusion element in the cipher. The four bytes of each column are multiplied by a 4 × 4 matrix as is illustrated as follows for one column

$$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{bmatrix} \tag{1}$$

where the entries of the 4 × 4 matrix are in hexadecimal notation, for example, 01 refers to a polynomial in GF(2^8) with coefficients 0000 0001.

(4) AddRoundKey: The state matrix after the MixColumns unit is added to the sub-key matrix, forming a new state matrix used in the next round. The sub-keys used in each round $k^i, i = 1, 2, \dots, 10$, are obtained by operations on the original key k^0 involving 4 S-boxes.

The nondeterministic modes of operation (CBC, OFB, CFB, CTR) are briefly described in the appendix [2,4]. Fig. 2 shows the results of the encryption of a gray-scale image with these modes of operation with 10 rounds. Except for the ECB mode, which reveals patterns of the original image, the mondeterministic modes produce random-like encrypted images.

2.1. Chaotic maps

A one-dimensional chaotic map generates a discrete-time series $\{x_n\}_{n=0}^{\infty}$ by the iteration of a suitable nonlinear and noninvertible function $f(x)$, with initial condition x_0, given by [15]

$$x_n = f(x_{n-1}), \quad n = 1, 2, 3, \dots \tag{2}$$

Thus, $\{x_n\}_{n=0}^{\infty} = \{x_0, f(x_0), f(f(x_0)), \dots\}$ is called an orbit of $f(x)$ starting at x_0. The first 200 samples are eliminated due to the transient behavior. The value of x_0 is given from a random integer of 128 bits, possibly generated from the original key k^0. Examples of chaotic maps include the cubic map (CM) $f(x) = 4x^3 - 3x$ [19], and the logistic map (LM) $f(x) = rx(1-x)$ [15], where r is a control parameter. New one-dimensional chaotic maps with enhanced chaos complexity have been developed by applying nonlinear transforms to the output of existing chaotic maps (called seed maps) [20,21]. An example of this map is the enhanced logistic map (ELM) [20] that uses the sine function as the nonlinear transform and is defined as [20]

$$f(x) = \sin(\pi a x(1-x)), \quad 0 \leq x \leq 1 \tag{3}$$

where the control parameter a is in the range $[0, \infty]$. Chaotic maps with complex behavior are also obtained by applying the cosine function to a combination of two seed maps. For example, the tent-logistic-cosine (TLC) map is given by (4) [21], shown in Box I, where the control parameter r is within the interval $[0, 1]$.

Chaotic maps are known to generate uncorrelated, noise-like, aperiodic sequences [15]. An important property of chaotic systems is that they are deeply sensitive on the initial condition, meaning that nearby trajectories separate exponentially fast. A widely used metric to measure this sensitivity on initial conditions is the Lyapunov exponent. A chaotic system has positive Lyapunov exponent [15].

The balanced binary sequence $\{z_n\}$, henceforth denoted by the chaotic binary sequence, is generated from $\{x_n\}$ from a partition of the map domain into two regions $\mathcal{R}_0$ and $\mathcal{R}_1$ satisfying $\Pr(x_n \in \mathcal{R}_0) = \Pr(x_n \in \mathcal{R}_1) = 1/2$ [22]. Thus, if $x_n \in \mathcal{R}_0$ then $z_n = 0$, or if $x_n \in \mathcal{R}_1$ then $z_n = 1$. Alternative discretization methods have been proposed, see, for example, [20].

3. Security metrics

This section describes security metrics commonly used in the literature to evaluate the ability of an image encryption scheme to resist statistical attacks [12–14,23,24]. In this work, we consider monochrome images of size 512 × 512 pixels with $L = 256$ gray levels.

(1) Shannon entropy: A measure of randomness is given by the Shannon entropy, defined as

$$H = -\sum_{i=1}^{L} \Pr(m_i) \log_2 \Pr(m_i) \tag{5}$$

where $\Pr(m_i)$ is the probability that a pixel has a gray value m_i and L is the number of gray values. For $L = 256$, the maximum value of H is 8 bits.

(2) Local Shannon entropy: The Shannon entropy in (5) is calculated over the entire image. An alternative measure, called (k, T_B)-local Shannon entropy, considers the randomness of local image blocks and is defined as [25]

$$H_{(k,T_B)} = \sum_{i=1}^{k} \frac{H_i}{k} \tag{6}$$

where H_i, $i = 1, \dots, k$, is the Shannon entropy of randomly selected non-overlapping blocks of the image with T_B pixels. The parameters used in [25] are $k = 30$ and $T_B = 1936$. It is provided in [25] a statistical criterion, based on the hypothesis test with significance level α, to decide if a cipher algorithm is approved in this test. This occurs if $H_{(30,1936)}$ lies within the interval $[7.9019, 7.9030]$ for $\alpha = 0.05$, $[7.9017, 7.9032]$ for $\alpha = 0.01$, $[7.9015, 7.9034]$ for $\alpha = 0.001$.

$$f(x)=\begin{cases}\cos(\pi(2rx+4(1-r)x(1-x)-0.5)), & \text{for } 0\le x\le 0.5;\\ \cos(\pi(2r(1-x)+4(1-r)x(1-x)-0.5)), & \text{for } 0.5<x\le 1\end{cases}\tag{4}$$

Box I.

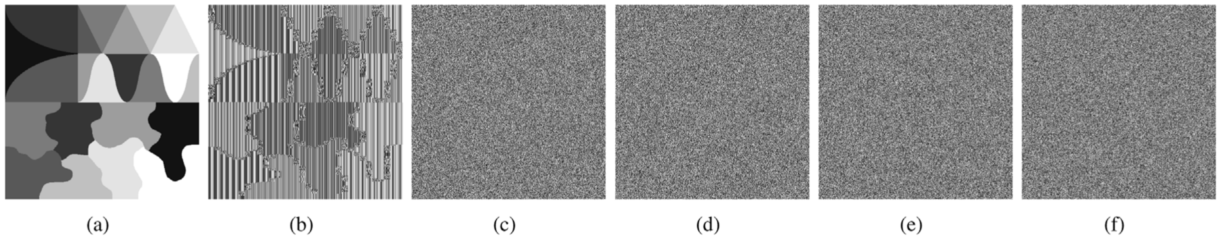
(a) (b) (c) (d) (e) (f)

Fig. 2. Original image (a). Modes of operation with 10 rounds: ECB (b), CBC (c), OFB (d), CFB (e), CTR (f).

(3) NPCR and UACI: The strength of an image encryption algorithm with respect to differential attacks is commonly evaluated by the number of pixel change rate (NPCR) and the unified average change intensity (UACI). Let C_1 and C_2 be two encrypted images of dimension $M \times N$, such that their original images differ by one pixel at a random position. The gray values (ranging from 0 to 255) at position (i,j) of C_1 and C_2 are denoted as $C_1(i,j)$ and $C_2(i,j)$, respectively. The NPCR and UACI between C_1 and C_2 are defined as

$$\text{NPCR}=\frac{\sum_{ij}D(i,j)}{N\times M}\times 100\%\tag{7}$$

$$\text{UACI}=\frac{1}{N\times M}\sum_{ij}\frac{|C_1(i,j)-C_2(i,j)|}{255}\times 100\%\tag{8}$$

where $D(i,j)$ is equal to 1 if $C_1(i,j) = C_2(i,j)$, or is equal to 0 if $C_1(i,j) \neq C_2(i,j)$. The maximum NPCR score is 100. In this work, 10 000 tests are performed with pairs of images C_1 and C_2 and the average values of NPCR and UACI are tabulated. Based on the statistical criterion provided in [26], a cipher algorithm is approved in this test (with significance level α) if the value of NPCR is greater than a critical value $N^{\star}_{\alpha}$. For images of size 512 × 512 pixels, these thresholds for three values of α are $N^{\star}_{0.05} = 99.5893$, $N^{\star}_{0.01} = 99.5810$, $N^{\star}_{0.001} = 99.5717$. A statistical criterion is also provided in [26] for passing the UACI test. In this case, this metric must be within the following ranges: $[33.3730, 33.5541]$ for $\alpha = 0.05$, $[33.3445, 33.5826]$ for $\alpha = 0.01$, $[33.3115, 33.6156]$ for $\alpha = 0.001$.

A good cipher must have high sensitivity with respect to changes in the original key k^0. Flipping a single bit in this key should lead to a widely different encrypted image. The NPCR can be used to validate this property. In this case, an original image is encrypted with two keys that differ by one bit at a random position, generating two encrypted images C_1 and C_2. This test is denoted by k-NPCR.

(4) NIST: The NIST test suit (version 800-22) [27] is used to test if a sequence is suitable for cryptographic applications. This includes 15 statistical tests focusing on distinct types of nonrandomness that could exist in a sequence. The tests are used to determine the acceptance or rejection of the hypothesis of ideal randomness with significance level β. We adopt in this work $\beta = 0.01$, a common value of β used in cryptography [27]. In the simulations, a binary sequence representing an encrypted image is the input to the NIST test suit.

(5) Correlation analysis: The correlation coefficient ρ measures the correlation between two adjacent pixels in horizontal, vertical and diagonal directions. We randomly select J pairs of adjacent pixels (x_i, y_i) in a given direction and calculate ρ as

$$\rho=\frac{\text{cov}(x,y)}{\sqrt{\text{Var(x)}}\sqrt{\text{Var(y)}}}$$

where

$$\text{cov}(x,y)=\frac{1}{J}\sum_{i=1}^{J}(x_i-\eta_x)(y_i-\eta_y)$$

$$\text{Var}(x)=\frac{1}{J}\sum_{i=1}^{J}(x_i-\eta_x)^2$$

and

$$\eta_x=\frac{1}{J}\sum_{i=1}^{J}x_i.$$

In a highly correlated image, the value of ρ is close to 1, while the ideal value of this parameter for an encrypted image is 0. We consider in this work $J = 10\,000$.

3.1. Security evaluation

We use 141 original images from the USC-SIPI image database (http://sipi.usc.edu/database). This database contains color and grayscale images of different sizes (such as 256 × 256 pixels, 512 × 512 pixels, 1024 × 1024 pixels) and we resize some images so that a new database contains images of the same size 512 × 512 pixels. Additionally, some color images are converted to grayscale images with 256 gray levels. In summary, all simulations in this work use a database with 141 grayscale images of size 512 × 512 pixels in bmp format (each image of size 768.1 kB).

Table 1 shows the minimum and maximum values of each security metric for all images in the database for each AES mode of operation with 10 rounds. The minimum value of the entropy for the ECB mode is not close to 8 bits, even in the tenth round and a significant correlation between adjacent pixels is also observed. For the nondeterministic modes of operation, the values of these metrics are significantly improved and all images pass the NIST test. Additionally, the minimum values of NPCR, UACI, k-NPCR and local entropy for these modes pass the corresponding statistical tests according to the criteria described in Section 3. The best results are obtained with the CTR mode.

4. AES modified with the chaotic map

The nondeterministic modes of operation use an initialization vector to break the deterministic behavior of the ECB mode. In this section, we present the algorithm ARQ1 that introduces a source of entropy in the AES S-box from the iterations of a chaotic map. The output byte of each S-box is added to a random byte generated by an evolution of a chaotic map while maintaining the ECB mode of operation. This proposal uses a fixed number of chaotic bits in each S-box, with an observed improvement in the robustness against cryptanalysis when

Table 1
Minimum and maximum values of the security metrics for the AES modes of operation.

Modes of operation	Entropy		Correlation analysis					
	Min	Max	Hrt		Vrt		Dgn	
			Min	Max	Min	Max	Min	Max
ECB	6.3872	7.9999	−0.0212	0.3146	0.0189	−0.2973	−0.0218	−0.3208
CBC	7.9982	7.9999	0.0109	−0.0214	−0.0093	−0.0228	0.0115	0.0191
OFB	7.9958	7.9999	−0.0116	−0.0249	−0.0113	0.0218	0.0111	−0.0276
CFB	7.9979	7.9999	−0.0109	0.01498	0.0111	−0.01839	−0.0097	0.0204
CTR	7.9988	7.9999	0.0087	0.0107	−0.0084	−0.0119	−0.0087	−0.0138
Modes of operation	NPCR		UACI		k-NPCR		Entropy local	
	Min	Max	Min	Max	Min	Max	Min	Max
ECB	0.003	0.006	0.0022	0.0035	99.5780	99.5918	6.8860	7.9025
CBC	99.599	99.617	33.395	33.503	99.5986	99.6060	7.9020	7.9029
OFB	99.589	99.611	33.382	33.445	99.5892	99.6066	7.9020	7.9030
CFB	99.590	99.609	33.383	33.443	99.5895	99.6084	7.9020	7.9030
CTR	99.601	99.621	33.397	33.509	99.6018	99.6117	7.9020	7.9030

Table 2
Minimum and maximum values of the security metrics for the ARQ1 algorithm per round. Cubic chaotic map.

Round	Entropy		Correlation analysis					
	Min	Max	Hrt		Vrt		Dgn	
			Min	Max	Min	Max	Min	Max
1	7.9538	7.9876	0.1012	0.1424	−0.0861	0.1652	−0.0871	0.1714
2	7.9826	7.9968	−0.0547	0.1048	−0.0343	0.0741	0.0311	0.0891
3	7.9867	7.9985	0.0223	−0.0424	0.0237	−0.0391	−0.0196	−0.0351
4	7.9868	7.9989	0.0165	−0.0283	−0.0163	0.0275	−0.0147	0.0287
5	7.9954	7.9995	−0.0128	0.0211	0.0117	−0.0197	0.0109	0.0203
6	7.9977	7.9999	−0.0101	−0.0184	0.0097	0.0174	0.0093	−0.1056
7	**7.9988**	**7.9999**	**0.0095**	**−0.0112**	**0.0092**	**0.0101**	**−0.0088**	**−0.0098**
8	7.9989	7.9999	−0.0077	0.0094	0.0085	−0.0091	0.0082	0.0083
9	7.9999	7.9999	0.0071	0.0088	−0.0075	0.0078	−0.0075	−0.0076
10	7.9999	7.9999	0.0053	−0.0081	0.0064	−0.0072	0.0058	0.0069
Round	NPCR		UACI		k-NPCR		Local entropy	
	Min	Max	Min	Max	Min	Max	Min	Max
1	99.3936	99.5981	33.2125	33.4827	99.5894	99.6074	7.8697	7.9010
2	99.5924	99.6180	33.4218	33.4981	99.5980	99.6122	7.9017	7.9024
3	99.6017	99.6213	33.4247	33.5105	99.5996	99.6126	7.9019	7.9025
4	99.6096	99.6216	33.4314	33.5109	99.6042	99.6131	7.9020	7.9026
5	99.6112	99.6224	33.4315	33.5110	99.6050	99.6160	7.9020	7.9029
6	99.6116	99.6238	33.4320	33.5123	99.6055	99.6185	7.9020	7.9029
7	**99.6118**	**99.6259**	**33.4341**	**33.5146**	**99.6074**	**99.6193**	**7.9020**	**7.9030**
8	99.6122	99.6263	33.4350	33.5170	99.6080	99.6198	7.9020	7.9030
9	99.6123	99.6268	33.4352	33.5181	99.6085	99.6201	7.9020	7.9030
10	99.6123	99.6268	33.4373	33.5217	99.6093	99.6226	7.9020	7.9030

this number of chaotic bits is increased. In this work, we set this number of bits to 3. For each plaintext block of 128 bits, the binary chaotic sequence $\{z_n\}$, generated as described in Section 2.1, is sectioned into blocks of 60 bits. The first 48 bits are used in the 16 S-boxes of the SubBytes unit, while the remaining 12 bits are used in the 4 S-boxes of the sub-key generation unit.

Three chaotic bits (z_j, z_{j+1}, z_{j+2}) are used in each S-Box, with polynomial representation $c(x) = z_j x^2 + z_{j+1} x + z_{j+2}$. Since the operations in each S-box are defined in GF (2^8), $c(x)$ is multiplied by a primitive polynomial $p(x)$ in GF (2^5), obtaining the polynomial $h(x) = c(x)p(x) \bmod P(x)$, where $P(x)$ is given in Section 2. The coefficients of this polynomial form a byte called **h**. The choice of $p(x)$ is made so that **h** is as balanced as possible in terms of zeros and ones. In this work, the polynomial $p(x) = x^5 + x^4 + x^3 + x + 1$ is chosen via exhaustive search. The byte **h** is bitwise added (module 2) to the output byte of the S-box in the SubBytes unit.

In the second round, the 48 chaotic bits used in the first round of the SubBytes unit are cyclically shifted to the right by the base ten value of the three bits used in the last S-box. For example, if the polynomial of the last S-box is $c(x) = x^2+x+1$, the shift is by seven bits. This procedure is repeated in each round. The purpose of this shift is to randomize the output of the SubBytes unit while keeping the same 48 chaotic bits per plaintext block. A similar procedure is performed in the S-boxes used to obtain the sub-keys $k^i, i = 1, \dots, 10$.

The performance of the ARQ1 algorithm (minimum and maximum values of each security metric for all images in the database) is shown in Table 2 for each round using the CM chaotic map. Comparing the minimum values of NPCR, UACI, k-NPCR, and local entropy with the thresholds defined in Section 3, we conclude that ARQ1 algorithm passes the corresponding statistical tests after the third round. These results indicate that the ARQ1 algorithm is highly sensitive to small changes either in the original image or in the original key. It is observed that after the seventh round the values of entropy and correlation coefficient reach the corresponding values achieved by the CTR mode of operation in Table 1 (this round is highlighted in bold in Table 2). This is also valid for the other chaotic maps used in this work. This is a good indication that the proposed algorithm is robust against statistical attacks. Furthermore, all images are approved in the NIST test suit after the first round. We conclude that the modified S-boxes with the inclusion of chaotic bits (both in the data path and in the sub-key generation) reach good indicators of randomness and robustness against cryptanalysis with fewer rounds. Fig. 3 shows the histogram and correlation coefficient in each direction of an original image and an encrypted image by the ARQ1 algorithm with seven rounds. A visual inspection reveals that the histogram of the encrypted image

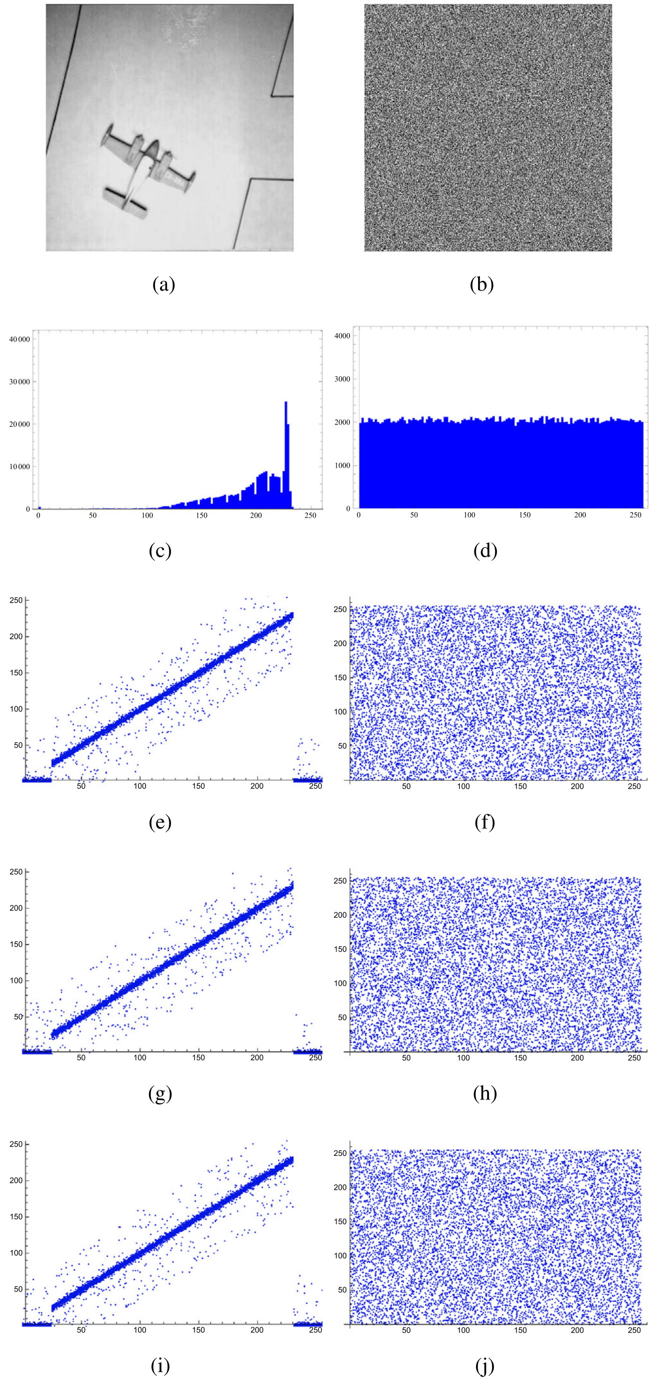

Fig. 3. Original image (a), encrypted image (b), histogram of the original image (c), histogram of the encrypted image (d), horizontal correlation of the original image (e), horizontal correlation of the encrypted image (f), vertical correlation of the original image (g), vertical correlation of the encrypted image (h), diagonal correlation of the original image (i), horizontal correlation of the encrypted image (j).

is uniformly distributed and the ARQ1 breaks the high correlation between the adjacent pixels of the original image.

For a comparison with some existing cipher algorithms, we consider 16 grayscale images from the USC-SIPI "Miscellaneous" dataset of size 512 × 512 pixels. Tables 3 and 4 display the NPCR and UACI values obtained by the ARQ1 algorithm with 7 rounds for a given image using three chaotic maps (CM, ELM with $a = 4$, TLC with $r = 0.5$) as well as the results reported in [21] and [28] for this dataset. It is observed that the ARQ1 algorithm passes the NPCR and UACI tests for all images and for the three chaotic maps (the critical values have been calculated under the significance level α equal to 0.05, that is NPCR ≥ 99.5893, UACI ∈ [33.3730,33.5541]). For some images, the ARQ1 slightly outperforms the ciphers considered in this comparison. The same trend is observed for the local entropy as shown in Table 5.

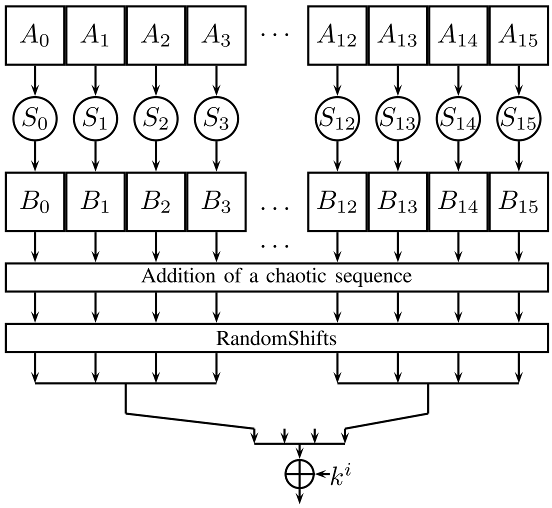

Fig. 4. Block diagram of the ARQ2 algorithm per round.

Table 3
NPCR values of the USC-SIPI "Miscellaneous" dataset of size 512 × 512 pixels.

File name	CM	ELM	TLC	[21]	[28]
5.2.08	99.6212	99.6107	99.6201	99.6250	99.5934
5.2.09	99.6119	99.6203	99.6134	99.6292	99.6002
5.2.10	99.6189	99.6205	99.6244	99.6212	99.6365
7.1.01	99.6244	99.6193	99.6127	99.6208	99.6147
7.1.02	99.6220	99.6117	99.6270	99.6025	99.6075
7.1.03	99.6142	99.6255	99.6119	99.6078	99.6048
7.1.04	99.6121	99.6140	99.6281	99.6082	99.6162
7.1.05	99.6091	99.6239	99.6153	99.6014	99.6326
7.1.06	99.6257	99.6131	99.6188	99.6063	99.6086
7.1.07	99.6150	99.6014	99.6248	99.5964	99.6185
7.1.08	99.6202	99.6240	99.6077	99.5953	99.6140
7.1.09	99.6106	99.6287	99.6212	99.6094	99.6002
7.1.10	99.6162	99.6077	99.6127	99.6078	99.6174
Boat.512	99.6113	99.6015	99.6099	99.6181	99.6151
Gray.512	99.6185	99.6107	99.6129	99.6029	99.5998
Ruler.512	99.6188	99.6177	99.6150	99.6033	99.6181

Table 4
UACI values of the USC-SIPI "Miscellaneous" dataset of size 512 × 512 pixels.

File name	CM	ELM	TLC	[21]	[28]
5.2.08	33.4410	33.4025	33.4528	33.4973	33.4642
5.2.09	33.4873	33.4806	33.5103	33.4778	33.5235
5.2.10	33.5014	33.5439	33.4621	34.4327	33.4638
7.1.01	33.3974	33.3927	33.4981	33.4154	33.4310
7.1.02	33.4715	33.4208	33.4015	33.4694	33.5172
7.1.03	33.4823	33.5119	33.4717	33.4632	33.4884
7.1.04	33.5319	33,4890	33.5021	33.4996	33.4156
7.1.05	33.4748	33.3914	33.4691	33.4647	33.4123
7.1.06	33.4509	33.4821	33.4770	33.4416	33.5102
7.1.07	33.3918	33.3872	33.4938	33.3906	33.4759
7.1.08	33.4782	33.4753	33.5092	33.4029	33.4500
7.1.09	33.4431	33.4392	33.4429	33.4686	33.4552
7.1.10	33.5211	33.4827	33.4632	33.4434	33.4374
Boat.512	33.4928	33.4718	33.4498	33.4472	33.4068
Gray.512	33.4701	33.4792	33.4972	33.4781	33.4598
Ruler.512	33.5017	33.5452	33.5289	33.3883	33.5389

Table 5
Local entropy values of the USC-SIPI "Miscellaneous" dataset of size 512 × 512 pixels. The numbers in bold fail to pass the local entropy test for $\alpha = 0.05$, that is, they are not within the interval [7.9019,9.9030].

File name	MC	SCM	TLC	[29]	[28]
5.2.08	7.9027	7.9024	7.9025	7.9025	7.9020
5.2.09	7.9026	7.9024	7.9028	7.9027	7.9023
5.2.10	7,9030	7.9028	7.9022	7.9022	7.9030
7.1.01	7.9027	7.9030	7.9028	7.9027	7.9022
7.1.02	7.9023	7.9024	7.9028	**7.8935**	7.9028
7.1.03	7.9029	7.9026	7.9027	**7.9007**	7.9020
7.1.04	7.9024	7.9028	7.9027	7.9022	7.9022
7.1.05	7.9027	7.9022	7.9026	7.9022	7.9025
7.1.06	7.9029	7.9024	7.9030	7.9030	7.9023
7.1.07	7.9030	7.9028	7.9027	7.9028	7.9028
7.1.08	7.9026	7.9028	7.9029	7.9024	7.9024
7.1.09	7.9020	7.9026	7.9022	7.9027	7.9030
7.1.10	7.9026	7.9023	7.9024	7.9027	7.9028
Boat.512	7.9029	7.9025	7.9022	7.9025	7.9024
Gray.512	7.9023	7.9029	7.9025	**7.8871**	7.9029
Ruler.512	7.9028	7.9022	7.9029	**7.8987**	7.9028

4.1. Algorithm ARQ2

The uncertainty introduced by the binary chaotic sequence allows simplifications of the ARQ1 algorithm in order to reduce its complexity. The MixColumns unit is responsible for the diffusion of bits and requires a large number of operations addition module 2. The number of operations can be counted from (1) [1]. The multiplication by 01 results in the same value as the input. The multiplication by 02 is implemented as a left shift of B_i followed by a conditional bitwise addition to 00011011 if the leftmost bit of B_i (before the shift) is 1. The multiplication $03 \times B_i$ is equivalent to $02 \times B_i + B_i$. Each $C_i, i = 1, \dots, 4$, requires (on average) 40 bitwise additions. In each round, these operations are repeated for four columns, so there are 640 bitwise additions per round to implement this unit (on average).

The proposed variation of ARQ1, named ARQ2, is shown in Fig. 4. In this algorithm, the ShiftRows and MixColumns units are replaced by a new unit, called RandomShifts. In each round, this unit performs a cyclic shift on the 128 bits obtained after the addition to a chaotic sequence by m positions ($m = 0, \dots, 127$), where m is the base ten value of the first 7 chaotic bits of the 48 bits used in the data path. Table 6 presents the performance of the ARQ2 algorithm for each round. The results of the ARQ2 with 6 rounds are similar to those of ARQ1 with 7 rounds, indicating the viability of the former algorithm. A comparison of the histogram and correlation coefficients for the original and encrypted images is provided in Fig. 5 for the ARQ2 with 6 rounds.

4.2. Encryption speed

We compare in this subsection the time required to encrypt 512 × 512 grayscale images for the algorithms considered in this work. These are implemented with the Mathematica v.10 program on the Linux operating system running on a personal computer with 3.70 GHz Intel Xeon E5-1620 CPU and 64 GB RAM. The test images are all 141 images in the database described in Section 3.1. The implementation is sequential and no parallel processing is performed. A table stores all possible input and output values of an S-box, performing a mapping between the input A_i and the corresponding output B_i. The chaotic binary sequence used in ARQ1 and ARQ2 are previously generated and stored in a vector.

Table 7 shows the average encryption time (in seconds) as well as the rate (in M bytes/s) of each algorithm. The ARQ1 is implemented with 7 rounds, and the ARQ2 with 6 rounds. The AES modes of operation have 10 rounds, as specified by the standard. The ARQ2 algorithm, which uses the RandomShifts unit with a reduced number of rounds in ECB mode, is the fastest algorithm.

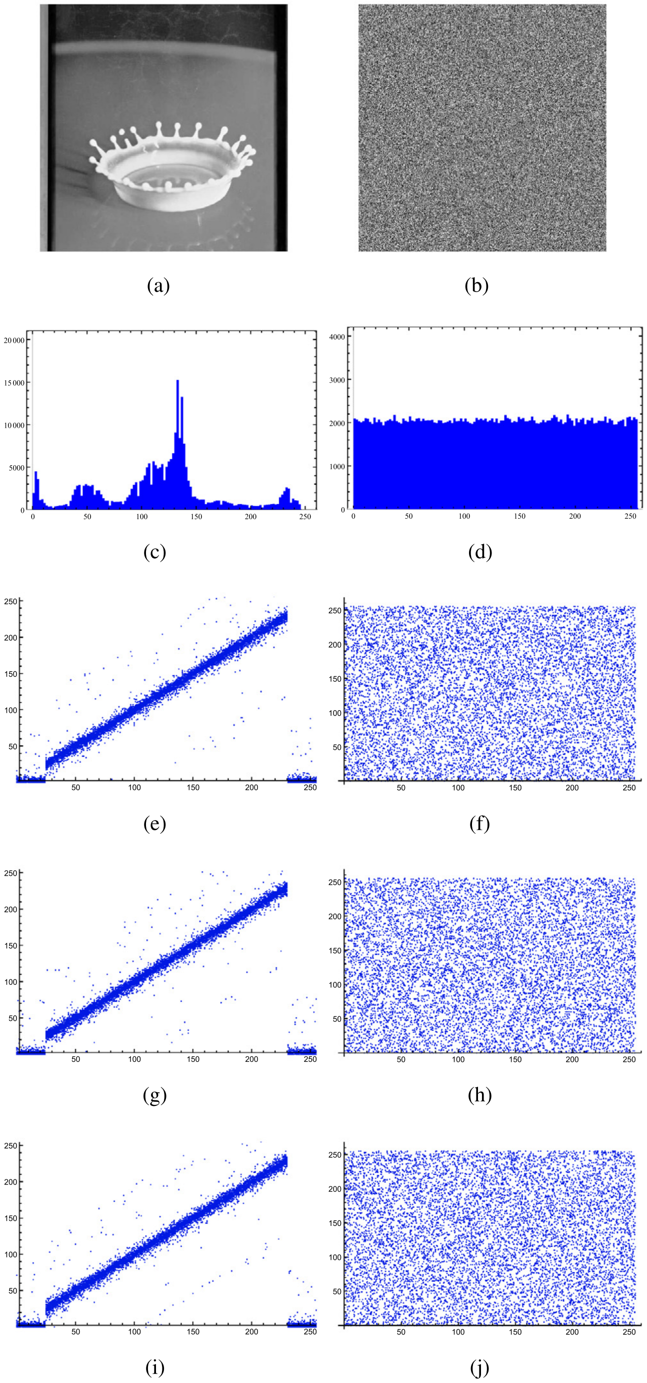

Fig. 5. Original image (a), encrypted image (b), histogram of the original image (c), histogram of the encrypted image (d), horizontal correlation of the original image (e), horizontal correlation of the encrypted image (f), vertical correlation of the original image (g), vertical correlation of the encrypted image (h), diagonal correlation of the original image (i), horizontal correlation of the encrypted image (j).

4.3. Sensitivity of the sub-key generation process

It is important to analyze the sensitivity of the sub-key generation process to a slight variation of the original key. Given an original key k^0 and the corresponding sub-keys $k^1, \dots, k^{10}$, let $\hat{k}^0$ be a key that differs from k^0 by a single bit and let the corresponding sub-keys be $\hat{k}^1, \dots, \hat{k}^{10}$. We randomly select 10 000 original keys and calculate how many bits are changed between k^i and $\hat{k}^i$ in the ith round of the algorithm. The average value is shown in Table 8. For a good diffusion, this should

Table 6
Minimum and maximum values of the security metrics for the ARQ2 algorithm per round. Cubic chaotic map.

Round	Entropy		Correlation analysis					
	Min	Max	Hrt		Vrt		Dgn	
			Min	Max	Min	Max	Min	Max
1	7.9534	7.9164	0.0527	0.0976	−0.0485	0.0956	−0.0556	−0.0824
2	7.9746	7.9868	−0.0256	−0.0424	0.0244	−0.0417	0.0234	0.0437
3	7.9899	7.9938	−0.0152	0.0182	−0.0122	−0.0173	0.0142	−0.0168
4	7.9965	7.9989	0.0105	−0.0118	−0.0098	0.0122	−0.0108	0.0113
5	7.9982	7.9993	0.0076	−0.0099	0.0087	−0.0101	−0.0099	0.0109
6	**7.9986**	**7.9999**	**0.0079**	**0.0092**	**0.0084**	**−0.0098**	**−0.0074**	**−0.0095**
7	7.9997	7.9999	−0.0068	−0.0085	0.0068	0.0093	0.0067	−0.0086
8	7.9999	7.9999	−0.0057	0.0077	−0.0058	−0.0079	0.0059	0.0081
9	7.9999	7.9999	−0.0054	0.0073	0.0056	0.0074	−0.0058	−0.0069
10	7.9999	7.9999	0.0050	−0.0058	−0.0051	0.0064	0.0056	0.0059

Round	NPCR		UACI		k-NPCR		Local entropy	
	Min	Max	Min	Max	Min	Max	Min	Max
1	99.5853	99.6091	33.3844	33.4437	99.6029	99.6116	7.8790	7.9019
2	99.6077	99.6174	33.3880	33.4562	99.6117	99.6170	7.9019	7.9021
3	99.6109	99.6216	33.4186	33.4813	99.6120	99.6175	7.9020	7.9026
4	99.6134	99.6220	33.4337	33.4508	99.6128	99.6178	7.9020	7.9028
5	99.6143	99.6254	33.4429	33.4514	99.6133	99.6194	7.9020	7.9029
6	**99.6147**	**99.6286**	**33.4462**	**33.4533**	**99.6142**	**99.6202**	**7.9020**	**7.9030**
7	99.6154	99.6289	33.4490	33.4551	99.6146	99.6245	7.9020	7.9030
8	99.6160	99.6295	33.4494	33.4554	99.6153	99.6303	7.9021	7.9030
9	99.6167	99.6300	33.4498	33.4580	99.6158	99.6311	7.9022	7.9030
10	99.6182	99.6347	33.4499	33.4587	99.6174	99.6338	7.9022	7.9030

Table 7
Comparison of the encryption time of different algorithms.

Algorithm	Time (s)	M bytes/s
ARQ1	10.42	0.0738
ARQ2	4.02	0.1419
CBC	15.41	0.0499
OFB	14.08	0.0546
CBC	14.01	0.0549
CTR	12.52	0.0614

Table 8
Sensitivity of the sub-keys.

Round	AES	ARQ1
1	8.8927	49.9378
2	24.3097	49.9921
3	28.6164	49.9938
4	36.7991	49.9958
5	39.9431	49.9967
6	49.9385	49.9981
7	49.9917	49.9984
8	49.9924	49.9985
9	49.9956	49.9991
10	49.9962	49.9992

be close to 50% in each round. For the AES algorithm, this number converges to 49.99% in the seventh round while the proposed algorithm achieves this level of diffusion in the second round, due to the binary chaotic sequence used in the sub-key generation unit.

5. Conclusions

A randomized encryption scheme was proposed with the introduction of a binary chaotic sequence in the SubBytes and sub-key generation units of the AES algorithm. We showed that a well known block cipher algorithm, when properly associated with chaotic maps acting as a source of entropy, presents results that point to two directions: reducing the number of rounds and simplifying the encryption algorithm. As a consequence, such a strategy could lead to lower energy costs and hardware simplification. An interesting future work is to extend the analysis to other block ciphers as well as consider cryptanalytic attacks, like differential attack, to better understand the effect of chaotic bits in the proposed encryption process.

Acknowledgments

This work was supported in part by the Brazilian National Council for Scientific and Technological Development (CNPq), Brazil and by the Foundation for Support of Science and Technology of the State of Pernambuco (FACEPE), Brazil.

Appendix

(1) Cipher Block Chaining Mode (CBC): In this mode, the current ciphertext block does not depend only on the current plaintext block, but also on the previous ciphertext blocks. Let $\mathbf{x}_k$ and $\mathbf{y}_k$ be the kth plaintext and ciphertext blocks (both of 128 bits), respectively. The kth input block to the AES is $\mathbf{x}_k + \mathbf{y}_{k-1}$ generating an output $\mathbf{y}_k$, where $\mathbf{y}_0$ is a random initialization vector (commonly referred to as IV). The encryption operation is sequential and is non-parallelizable. This scheme is shown in Fig. 6.

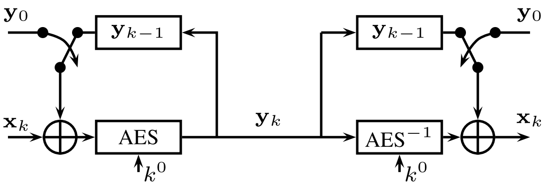

Fig. 6. CBC mode of operation.

(2) Output feedback Mode (OFB): This mode uses a block cipher to create a stream cipher. The ciphertext block is given by $\mathbf{y}_k = \mathbf{x}_k + \mathbf{s}_k$, where $\mathbf{s}_k$ is the AES output block when the input block is $\mathbf{s}_{k-1}$, and $\mathbf{s}_0$ is the initialization vector, as shown in Fig. 7. Due to the sequential operation, encryption and decryption are non-parallelizable.

(3) Cipher Feedback Mode (CFB): This mode also uses a block cipher to construct a stream cipher. Its operation is similar to the OFB mode, but the AES input block is $\mathbf{y}_{k-1}$, where $\mathbf{y}_0$ is a random initialization vector, as shown in Fig. 8.

(4) Counter Mode (CTR): As in the previous mode, a block cipher is used to construct a stream cipher. An initialization vector of length

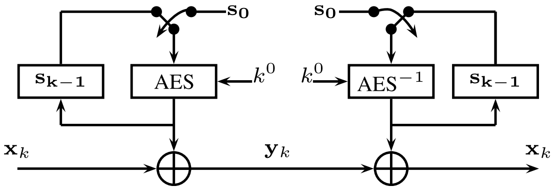

Fig. 7. OFB mode of operation.

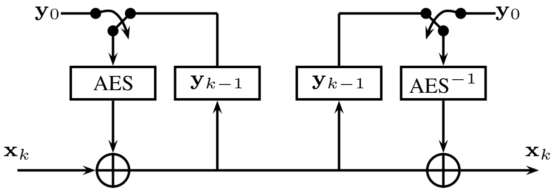

Fig. 8. CFB mode of operation.

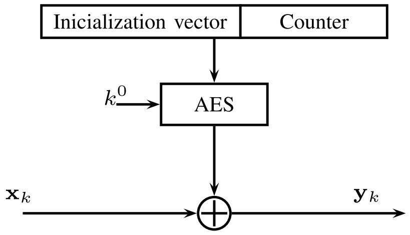

Fig. 9. CTR mode of operation.

smaller than the plaintext block (e.g. 96 bits) is concatenated to a counter (CTR). Whenever a block is encrypted, the counter is incremented, but the initialization vector stays the case. Both encryption and decryption can be performed in parallel. The scheme is shown in Fig. 9.

References

[1] F.I.P.S. (FIPS), Advanced encryption standard (AES), NIST, Tech. Rep. FIPS 197, Nov. 2001.
[2] C. Paar, J. Pelzl, Understanding Cryptography, in: A Textbook for Students and Practitioners, Springer, 2010.
[3] H. Li, Efficient and flexible architecture for AES, IEE Proc., Circuits Devices Syst. 153 (6) (2006) 533–538.
[4] S. Almuhammadi, I. Al-Hejri, A comparative analysis of AES common modes of operation, in: Proc. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, Canada, 2017.
[5] M. Preishuber, T. Htter, S. Katzenbeisser, A. Uhl, Depreciating motivation and empirical security analysis of chaos-based image and video encryption, IEEE Trans. Inform. Forensics Secur. 13 (9) (2018) 2137–2150.
[6] Y. Zhang, X. Li, W. Hou, A fast image encryption scheme based on AES, in: Proc. 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, 2017, pp. 2624–2628.
[7] L. Wenhao, S. Kehui, Z. Congxu, A fast image encryption algorithm based on chaotic map, Opt. Lasers Eng. 84 (2016) 26–36.
[8] B. Yang, X. Liao, Some properties of the logistic map over the finite field and its application, Signal Process. 153 (2018) 231–242.
[9] X. Wu, B. Zhu, Y. Hu, Y. Ran, A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps, IEEE Access 5 (2017) 6429–6436.
[10] J. Guo, D. Riyono, H. Prasetyo, Improved beta chaotic image encryption for multiple secret sharing, IEEE Access 6 (2018) 46297–46321.
[11] X. Wang, C. Liu, A novel and effective image encryption algorithm based on chaos and DNA encoding, Multimedia Tools Appl. 76 (5) (2017) 6229–6245.
[12] Z. Tang, J. Song, X. Zhang, R. Sun, Multiple-image encryption with bit-plane decomposition and chaotic maps, Opt. Lasers Eng. 80 (2016) 1–11.
[13] R. Lan, J. He, S. Wang, T. Gu, X. Luo, Integrated chaotic systems for image encryption, Signal Process. 147 (2018) 133–145.
[14] L. Xu, Z. Li, J. Li, W. Hua, A novel bit-level image encryption algorithm based on chaotic maps, Opt. Lasers Eng. 78 (2016) 17–25.
[15] S. Strogatz, Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering, in: Studies in Nonlinearity Series, Westview Press, 2001.
[16] Y. Liu, L. Zhang, J. Wang, Y. Zhang, K. Wong, Chosen-plaintext attack of an image encryption scheme based on modified permutation- diffusion structure, Nonlinear Dynam. 84 (4) (2016) 2241–2250.
[17] C. Li, Cracking a hierarchical chaotic image encryption algorithm based on permutation, Signal Process. 118 (2016) 203–210.
[18] C. Zhu, K. Sun, Cryptanalyzing and improving a novel color image encryption algorithm using RT-enhanced chaotic tent maps, IEEE Access 6 (2018) 18759–18770.
[19] F. Lau, C. Tse, Chaos-based Digital Communication Systems, in: Engineering online library, Springer, 2010.
[20] Z. Hua, B. Zhou, Yicong Zhou, Sine chaotification model for enhancing chaos and its hardware implementation, IEEE Trans. Ind. Electron. 66 (2) (2019) 1273–1284.
[21] Z. Hua, Y. Zhou, H. Huang, Cosine-transformation-based chaotic system for image encryption, Inf. Sci. 409 (2019) 403–419.
[22] J.V. Evangelista, J.A. Artiles, D.P. Chaves, C. Pimentel, Emitter-coupled pair chaotic generator circuit, AEU-Int. J. Electron. Commun. 77 (2017) 112–117.
[23] B. Wang, X. Yingjie, Z. Changjun, Z. Shihua, Z. Xuedong, Evaluating the permutation and diffusion operations used in image encryption based on chaotic maps, Optik 127 (7) (2016) 3541–3545.
[24] Z. Zhenjun, et al., Multiple-image encryption with bit-plane decomposition and chaotic maps, Opt. Lasers Eng. 80 (2016) 1–11.
[25] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. Noonan, P. Natarajan, Local shannon entropy measure with statistical tests for image random- ness, Inform. Sci. 222 (2013) 323–342.
[26] Y. Wu, J. Noonan, S. Agaian, NPCR and UACI randomness tests for image encryption, Cybern. J.: Multidiscip. J. Sci. Technol., J. Sel. Areas Telecommun. (JSAT) (2011) 31–38.
[27] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, Statistical test suite for random and pseudo random number generators for cryptographic applications, in: Special Publication 800-22 Revision 1a, National Institute of Standards and Technology, 2010.
[28] M. Wang, X. Wang, Y. Zhang, Z. Gao, A novel chaotic encryption scheme based on image segmentation and multiple diffusion models, Opt. Laser Technol. 108 (2018) 558–573.
[29] M. Alawida, A. Samsudin, J.S. Teh, R.S. Alkhawaldeh, A new hybrid digital chaotic system with application in image encryption, Signal Process. 160 (2019) 45–58.