Journal of Electronic Imaging

JElectronicImaging.org

Cipher image damage and decisions in real time

Victor Manuel Silva-García
Rolando Flores-Carapia
Carlos Rentería-Márquez
Benjamín Luna-Benoso
Cesar Antonio Jiménez-Vázquez
Marlon David González-Ramírez

SPIE. IS&T imaging.org

Cipher image damage and decisions in real time

Victor Manuel Silva-García,[a,*] Rolando Flores-Carapia,[a] Carlos Rentería-Márquez,[b] Benjamín Luna-Benoso,[c] Cesar Antonio Jiménez-Vázquez,[c] and Marlon David González-Ramírez[a]
[a]Instituto Politécnico Nacional, CIDETEC, Computer Security, Av. "Juan de Dios Bátiz" s/n esq. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, Del. Gustavo A. Madero, México D.F., Mexico
[b]Instituto Politécnico Nacional, ESFM, Code Theory, Av. Instituto Politécnico Nacional Edificio 9, Unidad Profesional Adolfo López Mateos, Zacatenco, Del. Gustavo A. Madero, C.P. 07738, México D.F., Mexico
[c]Instituto Politécnico Nacional, ESCOM, Pattern Recognition and Image Analysis, Av. Juan de Dios Bátiz esq. Av. Miguel Othón de Mendizábal, Col. Lindavista. Del. Gustavo A. Madero, C.P. 07738, México D.F., Mexico

Abstract. This paper proposes a method for constructing permutations on m position arrangements. Our objective is to encrypt color images using advanced encryption standard (AES), using variable permutations means a different one for each 128-bit block in the first round after the x-or operation is applied. Furthermore, this research offers the possibility of knowing the original image when the encrypted figure suffered a failure from either an attack or not. This is achieved by permuting the original image pixel positions before being encrypted with AES variable permutations, which means building a pseudorandom permutation of 250,000 position arrays or more. To this end, an algorithm that defines a bijective function between the nonnegative integer and permutation sets is built. From this algorithm, the way to build permutations on the $0, 1, \ldots, m-1$ array, knowing $m-1$ constants, is presented. The transcendental numbers are used to select these $m-1$ constants in a pseudorandom way. The quality of the proposed encryption according to the following criteria is evaluated: the correlation coefficient, the entropy, and the discrete Fourier transform. A goodness-of-fit test for each basic color image is proposed to measure the bits randomness degree of the encrypted figure. On the other hand, cipher images are obtained in a loss-less encryption way, i.e., no JPEG file formats are used. [DOI: 10.1117/1.JEI.24.1.013012]

Keywords: advanced encryption standard; transcendental numbers; encrypted images; variable permutations; decision in real time; damage in encrypted images.

Paper 14469 received Aug. 8, 2014; accepted for publication Dec. 2, 2014; published online Jan. 9, 2015.

1 Introduction

With the development of communications, the security problem for confidential information has emerged. Also, there is theft or damage to data repositories to which there is no free access. Therefore, there have appeared several encryption procedures for information, in particular, images.[1] There are some new methods using the Hilbert transform,[2] chaos,[3] the hyper-chaos,[4] or even the advanced encryption standard (AES) cryptosystem[5] with the CBC encryption mode,[6] although this way is a sequential encryption. The first two are fast, but they have a robustness problem,[6] in that it is not specifically mentioned what the number of elements in the key set is. The hyper-chaos method[4] is a key size of 2^{167} and they only speak of brute force attacks and no other type of analysis, for example, differential[7] and linear[8] attacks. By the way, the AES cryptosystem has no problem with any of them yet.[9] Moreover, the AES key set can reach 2^{256} elements. It is also important to note that the AES algorithm uses the substitution operation through a box. The substitution operation gives nonlinearity to the encryption process,[10] and none of the aforementioned algorithms. In fact, the nonlinearity of the AES box is superior to data encryption standard (DES) and triple-DES boxes.[11] Also, there are encryptions for color images using the transformed Fourier,[12] the gyrator and Arnold transforms.[13] However, in these two latest investigations, the specific algorithm complexity is not shown. On the other hand, there are some color image encryption investigations[14] where they do not specify the set key size. Other optical papers[15–17] demonsrrate cipher images in an original way, but they do not clearly show the key set size. There is also an important paper in image encryption,[18] although this alters the original image when it is decrypted, and this is important because in some countries it is not allowed. There is an interesting investigation using a chaotic map to encrypt images,[19] but that paper does not employ a test of NIST Special Publication 800-22 to measure the randomness of the encrypted images.

We decided to use the AES algorithm for images encryption for the following reasons: it is a recent symmetric encryption system, and it is also the International Standard at this time.[5] This makes the AES algorithm one of the most studied worldwide. However, an efficient method for breaking it has not yet been found.[20] The encryption "quality" of a figure concerns the randomness degree in the image bits distribution. Several methods have been utilized to measure the randomness degree.[21] In this work, the following are used: correlation coefficients; horizontal, vertical, and diagonal,[22] entropy and discrete Fourier transform (DFT). The latter measures the periodicity degree of the zeroes and ones string, i.e., whether or not a pattern is followed. Furthermore, a different way is proposed to measure the randomness degree of the encrypted figure bits using a "goodness-of-fit test."[22]

*Address all correspondence to: Victor Manuel Silva-García, E-mail: vsilvag@ipn.mx

This investigation does not use compression on images because there are some countries whose security areas do not allow compression in the encryption image.[23] In other words, it does not utilize the process: compression–encryption → decryption–decompression. It is only employs encryption → decryption. Five images as prototypes to be encrypted are used, of which four are in most papers on encryption figures; these images are the following: baboon, Barbara, Lena, and peppers. A criterion to select the fifth figure to be encrypted is proposed. All the images have a different difficulty degree to be encrypted when a symmetric cryptosystem is used. This difficulty depends on the randomness degree of the figure bits to be encrypted, since an image with a high randomness degree in its bits distribution is easier to encrypt than a figure that has low randomness, when ECB encryption mode is used. This paper is organized as follows: this section presents a very synthetic state of art and Sec. 2 shows the basic concepts to be used in this investigation. The bijective function is addressed in Sec. 3, and is also demonstrated. Section 4 approaches the AES algorithm with variable permutations, and constructs a permutation in the whole image. Ways to measure the randomness degree are presented in Sec. 5. Section 6 shows the encrypted figures analysis with damage, and the outcome of the randomness degree in the encrypted figures is shown in Sec. 7. Section 8 discusses the results. Finally, the conclusions are given in Sec. 9.

2 Preliminary Concept

Even though the AES algorithm is the most studied in the world, an efficient way to solve it is not known. Another important issue to clarify is that AES is a symmetric algorithm, which makes it a very fast method with which to encrypt data. However, if a figure with a low randomness degree in its bits distribution is encrypted using the AES algorithm without any modification, perhaps the image encrypted could provide information, i.e., the distribution of the different shades of basic colors follow a certain pattern. So, it is necessary to employ an additional element in the algorithm. Thus, this paper proposes applying a different permutation in each 128 bits block. This permutation is applied in the first round after the x-or operation. The reason why the permutation is not used at the first round entrance as is triple-DES[11] is because some images have areas with the same color, such as black or white. In this situation, whatever permutation applied to a zeroes or ones string would not make any change in the chain. If the permutation is used after the x-or operation, then it allows for changing some string bits. Another question is why in the first round? Well, it is understood that the information is mixed in each round, so any change carried out by the permutation at the first round will have more opportunity to mix the information. Therefore, at the end of the encryption process the zeroes and ones string will be random.

As is known, the transcendental numbers are not the solution for any polynomial whose form is $a_n x^n + a_{(n-1)}x^{(n-1)} + \ldots + a_0$ with $a_i \ \epsilon \ Z$, and besides, all the numbers after the decimal point have the property of not following any periodicity,[24] making them good candidates to be used as pseudorandom numbers.[25] In fact, for this feature, the irrational numbers are employed in Has-Sha functions.[26] The transcendent number used in this investigation is pi, because it has been studied for a long time.[27] On the other hand, the permutations generated depend on the AES key in accordance with the following procedure: denoted as l the integer which represents the 128 bits chain AES key. Then the product $(pi)l$ is also a transcendental number. Thus, using this last number it is possible to get the constants to build the permutations.

The entropy is measured according to the formula: $-\sum_{x\varepsilon X} P_r(x)\log_2 P_r(x)$. When working with each basic color of the images—red, green, or blue—each one can be described as 1 byte, i.e., 256 levels are sufficient for each. So, if each basic color has a uniform distribution, all points are equally likely, and the entropy value is 8.[28] This means the information is completely random. However, in practice, this is not so. Then values are sought as close to 8 as possible, in the basic colors' distribution red, green, and blue of the encrypted figure. If the image is mono-colored, the procedure is basically the same, since only 1 byte is used to describe the gray color, i.e., 256 different gray levels, following the same reasoning as the color images.

A statistic test to evaluate the chain bits randomness is formulated by means of a null hypothesis H_0 versus an alternative H_a. The null hypothesis establishes that the bits sequence is random and the alternative hypothesis is the opposite. To accept or reject the null hypothesis, a statistic and a threshold are used. If the statistic based on the data has a value bigger than the threshold, it implies that the null hypothesis is accepted, otherwise H_0 is rejected. In any hypothesis test scheme there are two errors, namely, type I and type II errors. The type I error is committed when H_0 is rejected when this hypothesis is true and the type II error is committed when H_0 is accepted and it is false. The type I error can be controlled, because it is supposed that H_0 is the more important of the two hypotheses. The amount used in this research for type I error is $\alpha = 0.01$, although the value $\alpha = 0.001$ can be used.[21] The error α is also called the significance level.

The probability distributions used in the randomness tests are: Chi-square χ^2 and complementary error function $\mathrm{erfc}(z) = (2/\sqrt{pi}) \int_z^\infty e^{-u^2}\mathrm{d}u$.[29] It is possible to express the $\mathrm{erfc}(z)$ function in terms of a normal standard cumulative distribution according to the following reasoning:

$$\Phi(z) = \left(\frac{1}{\sqrt{2pi}}\right) \int_{-\infty}^{z} e^{-\frac{u^2}{2}}\mathrm{d}u. \tag{1}$$

The normal standard cumulative distribution, and

$$\mathrm{erfc}(z) = \left(\frac{2}{\sqrt{pi}}\right) \int_{z}^{\infty} e^{-u^2}\mathrm{d}u \tag{2}$$

the complementary error function.

The next variable change is proposed for Eq. (2), $u = v/\sqrt{2}$ and $du = dv/\sqrt{2}$.

Therefore, $\mathrm{erfc}(z) = (2/\sqrt{pi})\int_{\sqrt{2}z}^{\infty} e^{-v^2/2}\mathrm{d}v$ then, this last expression is written thus: $\mathrm{erfc}(z = w/\sqrt{2}) = 2/\sqrt{2pi}(\int_w^\infty e^{-v^2/2}\mathrm{d}v)$, thus

$$\operatorname{erfc}\left(z=\frac{w}{\sqrt{2}}\right)=2\left(1-\frac{1}{\sqrt{2pi}}\int_{-\infty}^{w}e^{-\frac{v^2}{2}}\mathrm{d}v\right)$$
$$=2(1-\Phi(w)). \quad (3)$$

Regarding the concept of a real-time decision relates to the following: it is known that there are important decisions for which there is a short time to make them, therefore, the procedure expressed here contributes to the process of making a timely decision.

3 Bijective Function

Let us have the following considerations: given a natural $m \geq 2$ the sets $N_m = \{n\varepsilon N | 0 \leq n \leq m! - 1\}$ and $\Pi_m = \{\pi\}$ can be defined, such that π is a permutation of the $0, 1, \ldots, m-1$ array. According to the Euclid division algorithm,[30] $\forall\ n\varepsilon N_m$, this one can be written in a unique way as follows:

$$n = C_0(m-1)! + C_1(m-2)! + \ldots + C_{m-2}(1)! + C_{m-1}(0)!. \quad (4)$$

Note that for a given m, $(m-1)!, (m-2)!, \ldots 1!, 0!$ are fixed. It will be seen in the algorithm description that the constant $C_{m-1} = 0$. Also, it is easy to prove that

$$0 \leq C_i < (m-i) \quad \text{with} \quad 0 \leq i \leq (m-2). \quad (5)$$

When the constants $C_0, C_1 \ldots C_{m-2}$ are calculated, the following algorithm can be constructed:

Step 0.
An array in ascending order can be defined as follows: $X[0] = 0$, $X[1] = 1 \ldots X[m-1] = m-1$.

Step 1.
According to Eq. (5), $C_0 < m$; so $X[C_0]$ is an element from the step 0 array. $X[C_0]$ is removed from the step 0 arrangement and instead is replaced by $X[m-1]$, i.e., the last element of the array. Note that only two operations are performed, removal and replacement, in fact, the other array elements remain unchanged.

Step 2.
Again using Eq. (5), $C_1 < m-1$; thus $X[C_1]$ is an array element from step 1. In the same way as in the previous step, $X[C_1]$ is removed and is replaced by the last element step 1 array.

Step $m-1$.
If this process is repeated at the end, the result will have the following: $X[C_{m-2}]$ and $X[C_{m-1}] = k$ with $0 \leq k \leq m-1$. The number $X[C_{m-1}]$ automatically appears as it is the last element, i.e., $C_{m-1} = 0$ because it has position zero.

The arrangement of positive integers $X[C_0]$, $X[C_1] \ldots X[C_{m-2}]$ and $X[C_{m-1}]$ is a permutation of the $0,1 \ldots m-1$ array. This procedure is made in $m-1$ steps. Regarding the complexity to implement this algorithm is $O(m)$ because at every step a removal and replacement of an item is made. The remainder is unchanged.

Clearly, in the case of images with 250,000 pixel files or more the algorithm with a $O(m)$ complexity represents an important advantage. It is clear that the algorithm presented above defines a function that goes from N_m to Π_m; denoted as I_m. Next I_m: $N_m \to \Pi_m$ is demonstrated as a bijective function.

Theorem 1 Let us have the sets $N_m = \{n\varepsilon N | 0 \leq n \leq m! - 1\}$ and $\Pi_m = \{\pi\}$, such that $m \geq 2$ and π is a permutation of $0, 1, 2 \ldots m-1$ array. Then the function I_m: $N_m \to \Pi_m$ is bijective.

Proof. First, it is shown that I_m is a one to one function. It is used reductio ad absurdum as a demonstration method. In this vein, suppose that $n_1 \neq n_2\ \varepsilon N_m \Rightarrow I_m(n_1) = I_m(n_2)$; but according to Eq. (4) the positive integers n_1, n_2 can be written as follows:

$$n_1 = C_{0,1}(m-1)! + C_{1,1}(m-2)! + \ldots + C_{m-2,1}(1)!$$
$$n_2 = C_{0,2}(m-1)! + C_{1,2}(m-2)! + \ldots + C_{m-2,2}(1)!$$

But it is supposed that $I_m(n_1) = I_m(n_2)$, which means that the elements of both permutations were selected in the same way, therefore, it must be true that $C_{0,1} = C_{0,2}$, $C_{1,1} = C_{1,2} \cdots C_{m-2,1} = C_{m-2,2}$. If this is so, then $n_1 = n_2$. So, the latter contradicts the hypothesis and concludes that if $n_1 \neq n_2\ \varepsilon N_m \Rightarrow I_m(n_1) \neq I_m(n_2)$. At the moment, this proves that I_m is a one to one function. The test that I_m function is surjective is simple, since the number of elements in the sets N_m and Π_m is equal. □

4 Advanced Encryption Standard with Variable Permutations

At this point, the way to use the algorithm presented above to construct a pseudorandom permutation over m positions array and then how to apply this tool in image encryption is shown. In this vein, it proceeds as follows: If the algorithm developed in the previous section is observed, knowing the values C_i for $i = 0, 1, \ldots, m-1$ a permutation can be constructed. Note that it is not important to know the n number with all its digits, fortunately, because it is impossible to work with integers around $250{,}000! - 1$ or higher since they are huge.

In this sense, the quantities $(m-1)!$, $(m-2)!\ldots$ are used only as marks, so it is not important to write all the digits. Then, the next question to address is how to choose pseudorandom values C_i for $i = 0, 1, \ldots m-1$. First, the permutations are built on 128 position arrangements using the pi number thus:

1. The symmetric cryptosystem key AES is a string of zeroes and ones, which in turn represents a positive integer, that is, if the key is 128 bits length, then the integer associated to the bits string has the form $(c_{m-1})2^m + (c_{m-1})2^{m-1} + \cdots + c_0$, where $c_{m-i} = 0, 1$ for $m = 127$ and $i = 0, 1, \ldots, 127$. So this integer can be denoted as l, then this paper proposes that l multiplies pi, such that the product is itself a transcendent number.

   Particularly, in this research, the AES-128 symmetric cryptosystem is used, although there is the possibility of using keys up to 256 bits.

2. After making the multiplication $l * pi$, it is taken to the right of the decimal point in 8-bit blocks. These blocks are also positive integers and they are denoted as $a_0, a_1, \ldots, a_{126}$. C_i can be defined as $C_i = a_i$ mod. $128 - i$, for $i = 0, 1, \ldots, 126$ and $C_{127} = 0$. If in the above procedure each 127-bytes block is taken one after the other, the required number of bytes can be very large. For example, for an image of 7,372,800 bits, 57,600 128-bit blocks are required. Then the necessary bytes are $57,600 * 127 = 7,315,200$. This amount may be reduced if the procedure is as follows: the first permutation can be built from byte 0 to 126, the second permutation from byte 1 to 127, the third from 2 to 128, and so on until the required number of permutations is reached. If this procedure is made in this way, the bytes number used for the above example is $126 + 57,600 - 1 = 57,725$; as can be seen, this is a significant reduction.

For the whole image, the number of items to permute, m, can have 250,000 or more elements. For a situation like this, the procedure is the same as when $m = 128$. There are some differences, namely, the size of the blocks in this case a_i is 24 bits. This is because many of the current images do not exceed 2^{24} bits in the spatial resolution.

Clearly, these blocks also represent positive integers. So, the C_i can be defined as $C_i = a_i$ mod. $m - i$, for $i = 0, 1, \ldots, m - 2$ and $C_{m-1} = 0$.

Sometimes, some bytes can be subtracted from the image to be encrypted according to the next criterion: if $24(m)$ mod. $128 \not\equiv 0$ where m is the pixels number of the image, then the minimum amount of bytes required is subtracted, say n, such that $24(m) - 8(n)$ mod.$128 \equiv 0$. It is important to point out that $8n < 128$ and this number of bits, $8n$, is not encrypted. Once the values $C_0, C_1, \ldots, C_{m-2}$ are known, the π_m permutations on m elements array can be calculated according to the procedure described in Sec. 3. In the case of AES with variable permutations the constant sets that are necessary according to the image size are calculated.

In a particular situation such as: a secure communication scheme such as public key infrastructure (PKI),[31] where the AES-128 and Elgamal[32] cryptosystems are used, the procedure is as follows:

1. In a random way, the AES-128 cryptosystem key is chosen, that is, a chain of 128 bits.
2. The bits string is converted into a positive integer, which is denoted as l, and later the multiplication $l * pi$ is performed.
3. The $C_0, C_1, \ldots, C_{m-2}$ values are calculated as described above to get the π_m permutations; one over $m = 128$ positions and another for the whole picture.
4. The π_m permutation to the entire image is applied and the image permutated is encrypted with an AES-128 system with variable permutations.
5. The sender encrypts the AES-128 key with the addressee's public key using the asymmetric cryptosystem Elgamal. Subsequently, the receptor can find the AES-128 key using its private key.

In this research, the signature and nonrepudiation as part of the structure of PKI secure communication are not mentioned[33] since they are not within the scope of this work.

Figures 1 and 2 flowcharts are shown; namely, the first illustrates how the permutation for the entire image is obtained. The second exemplifies how the variable permutations of 128 positions are developed.

Regarding the security of this cryptosystem, the following can be mentioned: the worst that can happen is that the AES key could be known, because if this encryption scheme is used, the permutation over the whole image and the variable permutations applied in each block can be calculated. So, in this situation, the maximum security is 2^{128}. However, if we want to find the key, taking as a plaintext the permuted image using the brute force attack, then this could be a problem with a complexity of 2^{843}; because in a previous work[34] a strong evidence was given that the DES algorithm with variable permutation has a complexity of $2^{56} \times (64!)$, where the Monte Carlo method was used. Clearly, if the problem is to find the key from the initial image, the solution to the problem would be more complex. Cases of plaintext are chosen as differential and linear attacks, which are not applicable to the AES cryptosystem due to the way the substitution box is constructed.[6]

5 Randomness Analysis

5.1 *Correlation Coefficient, Entropy, and DFT*

In this section, randomness using the following tests will be analyzed: correlation coefficient of horizontal, vertical, and diagonal directions, entropy, and DFT. It is also pointed out that the image encryption process is performed without compression, specifically, loss-less of information. In any image encryption process, it is important that the bits distribution be random in order to avoid bias that could lead to attacks for finding the key or plaintext.

Adjacent pixels are considered in three directions, namely, horizontal, vertical, and diagonal. Furthermore, it is said that a picture is "well encrypted," if the correlation coefficient between adjacent pixels is a number close to zero.[35] The process of calculating the correlation coefficient between two random variables X, Y is carried out as follows: in a random way a pixel of the encrypted image is chosen. This pixel has a level of red, green, and blue which is denoted as x_r, x_g, and x_b, i.e., the analysis is performed for each primary color. After selecting a pixel in a random way, the next pixel in the adjacent directions, horizontal, vertical, or diagonal, is taken. The adjacent pixel has a level of red, green, and blue. These levels are denoted as follows: y_r, y_g, and y_b.

Now, suppose that M pairs of pixels x, y are chosen randomly. It is possible to calculate the correlation coefficients in the three directions for the three basic colors. The equatio for calculating the correlation coefficient in the horizontal direction and a basic color c is thus

$$r_{h;x_c,y_c} = \frac{\sum_{i=1}^{M}(x_{h;i,c} - \overline{x}_{h,c})(y_{h;i,c} - \overline{y}_{h,c})}{\sqrt{[\sum_{i=1}^{M}(x_{h;i,c} - \overline{x}_{h,c})^2][\sum_{i=1}^{M}(y_{h;i,c} - \overline{y}_{h,c})^2]}}, \tag{6}$$

where $\overline{x}_{h,c}$ and $\overline{y}_{h,c}$ are presented next:

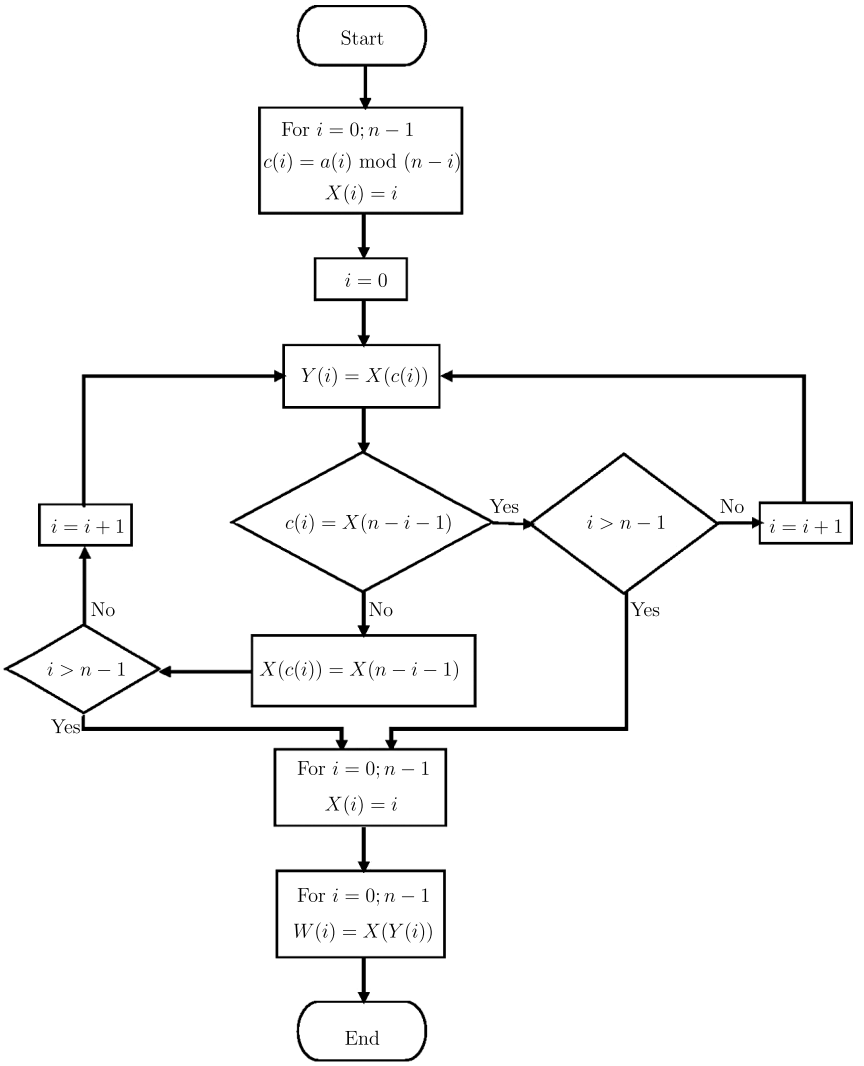

Fig. 1 Permutation for the entire image. The *n* is the number of pixels. The product $(I) * (\pi)$, where *I* is the integer associated to the key. The $a(i)$ is the block number *i* of 24 bits, after the decimal point of product $(I) * (\pi)$. The constant $c(i)$ is the number *i* to obtain the permutation $Y(i)$. The $W(i)$ is the element number *i* of the permuted image.

$$\bar{x}_{h,c} = \frac{1}{M}\sum_{i=1}^{M} x_{h;i,c} \quad \text{and} \quad \overline{y}_{h,c} = \frac{1}{M}\sum_{i=1}^{M} y_{h;i,c}. \tag{7}$$

Clearly, the vertical and diagonal expressions are the same.

In case of a mono-color image, the method is only for 256 gray levels.

The entropy analysis of the image pixels is performed for each basic color apart. In this sense, any basic color, red, green, or blue, requires only 1 byte to express the entropy, i.e., 256 levels. In this vein, it is said that if the distribution of the pixels is completely random then the entropy of any of the basic colors is 8. Measuring the randomness in practical cases to strings of zeroes and ones has the following reasoning: when it is close to 8 this means that the string of zeroes and ones is random, otherwise it would mean the opposite.

The pixels are separated in their primary colors to calculate the entropy. Assume that the bits string has the basic color c, which is divided into blocks of 8 bits. It follows that there are 256 possible values. The frequencies are recorded in a table of 256 classes according to their order of appearance. Then, each class has a frequency f_i for $i = 0, 1, \ldots, 225$, so that an estimate of the probabilities of each of the classes is $P[x_i] = 1/f_i$. Therefore, the entropy for basic color c is calculated as follows: $H_c = -\sum_{x_i \varepsilon X} P_c(x_i)\log_2[P_c(x_i)]$.

The DFT measures the randomness degree of a string of zeroes and ones i.e., there is no periodicity—repetitive patterns—one after another in the string of zeroes and ones. In addition, the following elements in the calculation of the statistic test are shown:

N_0 is the theoretical amount expected; $(0.95)(n/2)$, where n is the chain length.

N_1 is the number of values less than a threshold h, which depends on the length n in the string.

The value $f_j = \sum_{k=1}^{n} x_k e^{\frac{2(pi)j(k-1)i}{n}}$, where $i = \sqrt{-1}$ and $j = 1, 2, \ldots, \frac{n}{2} - 1$.

If n is odd, the last chain bit is suppressed. Clearly, f_j has real and complex parts. Then, the module $\|f_j\|$ is calculated, which is real, and is compared with h. If $\|f_j\| < h$, a one is added to the value of N_1. Otherwise, N_1 stays at

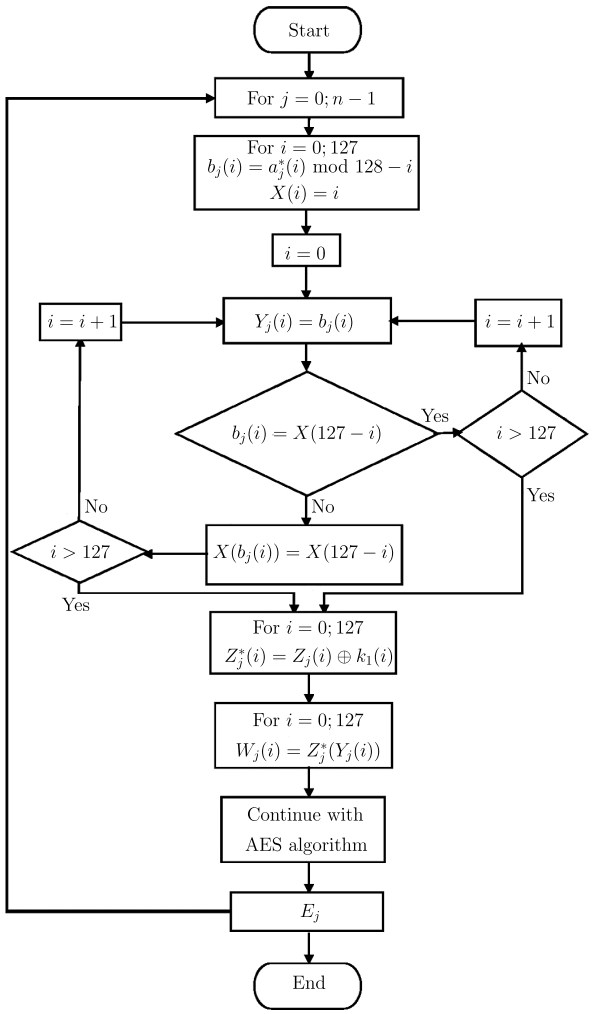

Fig. 2 Permutations of 128 positions in the cipher of the image. The n is the number of blocks of 128 bits of the permutated image. The $a_j^*(i)$ is the byte number i of the product $(I) * (\pi)$, after the decimal point. The $Z_j(i)$ is the bit of position i of block j of 128 bits to be ciphered. The k_1 is the first key of the schedule of keys. The E_j is the block number j of 128 bits ciphered. The $Z_j^*(i)$ is the result of the operation x-or between Z_j and k_1. The W_j is the block that results after the application of permutation Y_j.

its previous value. With this data, the quantities $d = N_1 - N_0/\sqrt{\frac{n(0.95)(0.05)}{4}}$; the statistic $P\text{-value} = \text{erfc}(d/\sqrt{2})$ and $\text{erfc}(\frac{d}{\sqrt{2}}) = 2(1-\Phi(d))$ are calculated. The decision rule is: if the P-value is less than 0.01, the null hypothesis is rejected, otherwise, it is accepted. The null and alternative hypotheses were defined in Sec. 2. The three tests are illustrated in Sec. 7 with the particular value $k =$ 2F9A68D501CB 57F3A4E80B9A417AD254 key of 128 bits.

5.2 *Proposed Test*

Working with images, a test of randomness based on the way the bits are arranged in an encrypted figure is proposed and the statistical, Chi-square, $\chi^2 = \sum_{i=1}^{k}[(o_i - e_i)^2/e_i]$ is used for each of the basic colors. The amounts o_i and e_i are the observed and expected values number i, respectively. Using statistical χ^2 it is possible to quantify the freedom degree that has the distribution of the primary colors: red, green, and blue. All the NIST 800-22 tests do not have this type of proof, that is, the randomness of the basic color distribution of an encrypted image is not measured. As in some of the NIST 800-22 standard tests, in this proposal the goodness-of-fit test is applied, using the statistical, χ^2, which has a probability distribution Chi-square with $n-1$ freedom degrees.[22] The freedom degrees are obtained in the following way: the shades of each color of an image can be displayed as a histogram whose abscissa has 256 divisions. Then, the degrees of freedom are 255. Moreover, the random variable χ^2 can be approximated to the normal distribution according to the central limit theorem.[36] Thus, the mean and standard deviation of the statistical χ^2 are: $\mu = 255$ and $\sigma = \sqrt{2(255)} = 22.5831$. With this information, it is simple to calculate the thresholds for significance levels $\alpha = 0.01$ and $\alpha = 0.001$, considering that both boundaries are on the right side of the normal distribution. The threshold for the level of significance $\alpha = 0.01$ is 307.61 and when $\alpha = 0.001$ it is 324.78. Therefore, the process for making the decision to accept or reject the null hypothesis according to a specific bits chain is as follows:

a. The statistical $\chi^2 = \sum_{i=1}^{k}[(o_i - e_i)^2/e_i]$ is calculated for specific values, considering that o_i and e_i are the observed and expected values number i.
b. The probability after the χ^2 value is computed, i.e., the area down the normal curve after χ^2 is calculated. If this probability is greater than or equal to 0.01 the null hypothesis is accepted, otherwise it is rejected. If the significance level is 0.001 the procedure is the same.

Of note, consider that an encrypted figure is rejected when some of the basic colors do not pass the proposed randomness test. Also, the size of the type I error used in this test is $\alpha = 0.01$. Then, taking into account that rejecting an encrypted image happens when at least one of the primary colors fails the randomness test and considering that the probability of rejecting any of them is $p = 0.01$, it follows that this situation can be described using the binomial model;[36] that is, $P(X = x) = \binom{3}{x}p^x(1-p)^{3-x}$ where x is the number of primary colors that do not pass the hypothesis test. So, the rejection happens when $x = 1, 2, 3$. Therefore, the probability of acceptance is approximately $(0.99)^3 = 0.9703$, and the probability of rejecting the encrypted image is about $p_1 = 0.0297$.

However, the probability of rejection can be reduced if the procedure is as follows: suppose that five keys are chosen in a pseudorandom way, independent of each other, then the probability that an encrypted image does not fulfil the randomness requirement is rejected as mentioned above, as happens when none of the five figures encrypted fulfill the randomness test for all the primary colors. This type of problem can also be solved using the binomial model, taking into account that $p_1 = 0.0297$. So, $P(X = x) = \binom{5}{x}p_1^x(1-p_1)^{5-x}$ where x is the number of encrypted images that are rejected because they do not fulfill the proposal test randomness criteria. Then, a setting rejection happens when $x = 5$, i.e., none of the figures encrypted have a randomness quality. In this case, the probability of not accepting any of them is $(0.0297)^5 = 0.000000023$, which means that for every 1000 millions that the latter process makes, about 23 are rejected. Another comment:

1. The probability of rejection can be decreased as much as you want by simply increasing the number of encryption figures.
2. The encryption time is not incremented much if parallel programming is used.[37]

Now try another question: assume that 100,000 different keys are taken and the image (c) of Fig. 6 is encrypted. Also assume that the figures encrypted with these keys pass the hypothesis test using the criterion given above. Then, in Sec. 7, the following results will be presented: the average entropy for each of the basic colors using these 100,000 encrypted figures. Regarding the correlation coefficient, the averages are also considered in three directions: horizontal, vertical, and diagonal, and for each of the three primary colors: red, green, and blue. Moreover, the furthest value of 8 for each of the basic colors in the case of entropy is reported, and the largest absolute value of the correlation coefficient is also shown in the three directions and the three basic colors.

5.3 *Criteria of Image Selection*

In Sec. 1, it was mentioned that a criterion would be presented to choose a fifth figure to be encrypted. This criterion is based on a characteristic of the goodness-of-fit test that tells us the following: if the tone distribution in each basic color was completely random, then $\chi^2 = 0$. In fact, this means that the color histogram follows a uniform distribution. However, if χ^2 has a large value for each basic color, it means they have a defined order. So the authors propose choosing an image that has a χ^2 as large as possible for each primary color. In this paper, a figure with the values of χ^2: $\chi^2_r = 106361059.17$, $\chi^2_g = 106647915.51$ and $\chi^2_b = 107366956.56$, for the red, green, and blue colors, respectively, is proposed. This image is a simulation of Latin text that is usually used in graphic design typographic demonstrations or drafts.[38] This image is shown in Fig. 3.

On the other hand, there is another reason for making the χ^2 very large since there are many images with relatively small values for each statistical χ^2 of the basic color, say less than half a million; in these cases the AES cryptosystem can be applied directly, i.e., it is not necessary to use variable permutations to encrypt an image and the result passes the above tests and even the proposal. For example, image (b) of Fig. 6—Barbara—has the quantities of χ^2 for the primary colors red, green, and blue: $\chi^2_r = 95086.89$, $\chi^2_g = 95086.89$, and $\chi^2_b = 95086.89$; these values are equal because it is a mono-colored image. In Fig. 4, Barbara and her encrypted image are presented without using variable permutations. By the way, Fig. 4(a) is ciphered considering the same gray level of each pixel in the three planes and is later encrypted in blocks of 128 bits.

For figures with a very large χ^2 for each different basic color, say more than 100 million, the variable permutations must be applied so that the encryption process is effective. Figure 5 shows when the encryption process can be ineffective in a simple way, since the χ^2 for the primary colors are: $\chi^2_r = 5691872.24$, $\chi^2_g = 5714519.25$, and $\chi^2_b = 5706278.92$ for red, green, and blue, respectively. Remember, it was noted above that the rejection region threshold is 324.78 for $\alpha = 0.001$ as maximum. Furthermore, the χ^2 values mentioned above are much higher, in fact, about 6 million.

Basically this is the reason why an image with an χ^2 as large as possible is proposed in order to verify that the cryptosystem presented is efficient. This section is suitable to show the five figures to be encrypted. The first one is presented in Fig. 3; the other four are shown in Fig. 6.

(a) (b)

Fig. 4 Comparing the original (a) and encrypted (b) images without variable permutation.

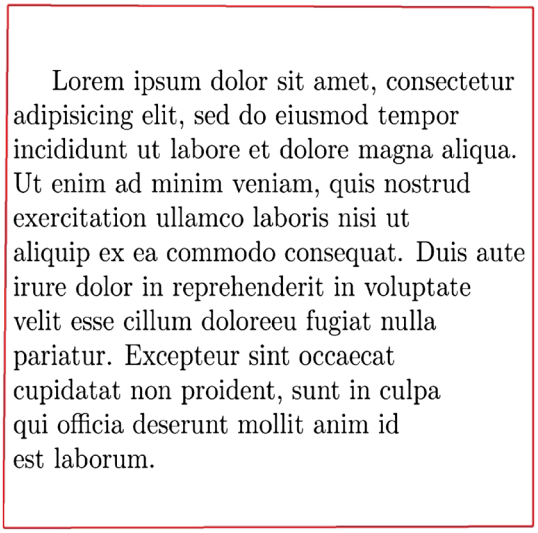

Fig. 3 Type image to be encrypted.

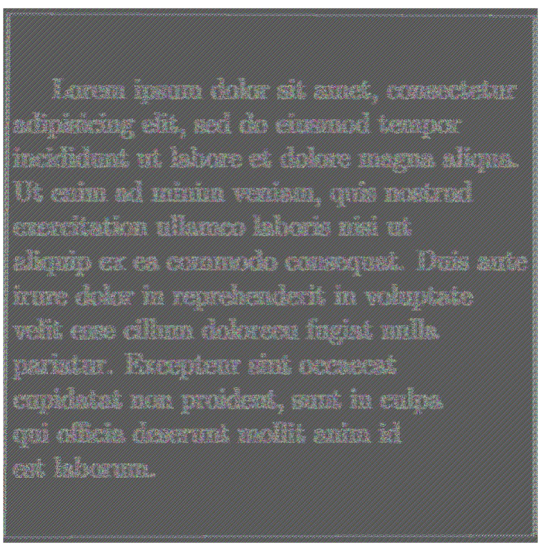

Fig. 5 Type image encrypted without variable permutation.

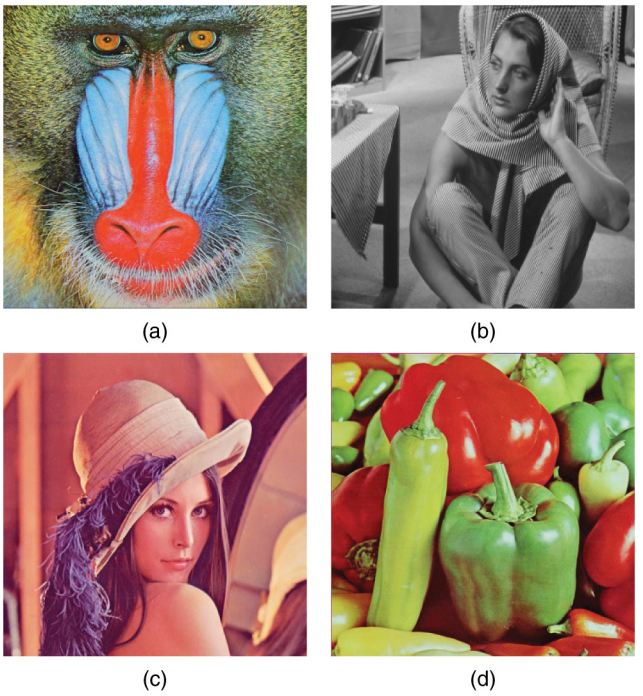

Fig. 6 The four type images to be encrypted, (a) baboon, (b) Barbara, (c) Lena, and (d) peppers.

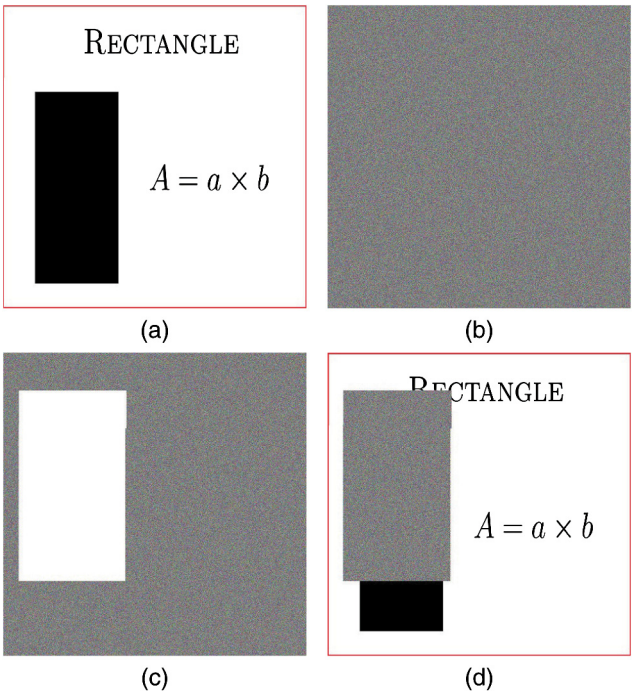

Fig. 7 (a) Original image, (b) image (a) encrypted without initial permutation, (c) image (b) with damage, and (d) image (c) deciphered.

6 Damage in the Encrypted Images

In this section, the figures ciphered with damage are treated, either accidentally or voluntarily. It is clear that the damage in the encrypted images is an attack because the message receiver cannot know what is it, therefore, the receiver does not make a decision or decisions that may be important. Time is a factor, too, because there are decisions that cannot wait. So, the decisions that concern us have the following characteristics: first, they are important for a state or corporation, and second, they have a very short time to make a decision, say less time that is required to again ask for the original message. In this research, a way to solve such problems is presented with some restrictions, of course. On the other hand, this investigation only analyzed occlusion = type damage, and did not take into account additive or multiplicative noise.

It starts by showing an original image, which is encrypted without applying a permutation over the whole figure before being encrypted. Later, the encrypted image is damaged and at the end is deciphered, see Fig. 7.

In Sec. 3, the way to generate a permutation for the whole image was presented. So it is possible to apply a permutation to an array of 250,000 or more positions; in this particular case, the pixels number of the original figure. The purpose of using a permutation in the original image before being encrypted has the objective to disperse the information, thus, when the figure encrypted with damage is decrypted, the damage is dispersed and it is possible to perceive the original picture. Of course, it also depends on the size of the damage. This paper proposes that the size damage is not greater than 40% of the cypher image. It is easy to realize that this amount may be higher or lower depending on the "sharpness" degree that is required in the decoded image. The Chi-square statistic for a basic color c of an encrypted image with damage is denoted as $\chi^2_{x\%,c}$ considering $x\%$ is the size of the failure. In this sense, the Chi-square of the original image to the same color c can be written as follows: $\chi^2_{0\%,c}$. Then, the ratio $\tau = \chi^2_{x\%,c}/\chi^2_{0\%,c}$ is the information percentage that is known of color c, with respect to the same original figure color pi. It will be noted in Sec. 7 that a failure less or equal to 40% has a ratio, as minimum, of about 34%. That is $\tau \geq 34\%$ for images of about 512×512 or higher. It is easy to see that this quantity depends on the size of the damage.

With regard to the manner by which the failure of the encrypted figure is made, it is carried out by means of concentric rectangles, see Fig. 8. The k key of 128 bits for the AES cryptosystem was written in Sec. 5.1. Figure 8 also presents an encrypted image with 40% damage and later the figure is decrypted. The image (b) of Fig. 6 is used to illustrate the point.

7 Results Presentation

7.1 *Randomness Results of the Encrypted Images Without Damage*

In this section, the correlation coefficient, entropy, DFT, and the proposed test are applied to five images, corresponding to Figs. 3 and 6. Furthermore, the results of this procedure are presented. The k key of 128 bits was written in Sec. 5.1, and as noted earlier, the k key is associated with a positive integer, and it is: $l = 63275455764858117084829320942584517204$. Then, if l is multiplied by pi, that is, the product $l * pi$, the result is also a transcendental number.

Taken to the right of the decimal point, the bits amount needed to calculate all the constant sets are used to encrypt the image, in addition, the permutation used for the whole image. Table 1 shows the results of the randomness test using DFT for the encrypted images of Figs. 3 and 6. This encryption is performed in two steps: in the

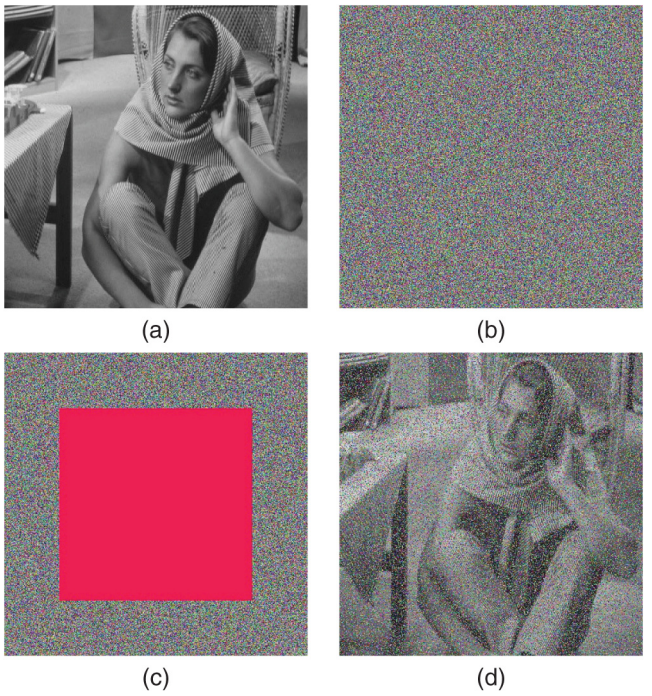

Fig. 8 Image (b) of Fig. 6 encrypted with prior permutation and decrypted with 40% of damage.

first, a permutation is applied over the whole original image; the second uses the AES cryptosystem with variable permutations.

The results for the proposed test are presented in Table 2. Note that the analysis for each primary color is included; it is also important to mention that for all the cases, the null hypothesis is accepted. Regarding entropy, analysis of randomness for the three basic colors of the images encrypted with the key k is performed. Again, the images that are encrypted correspond to Figs. 3 and 6. Furthermore, it is considered that for a figure that has been "well encrypted," entropy must be very close to 8. The results are presented in Table 3.

To calculate the correlation coefficient, a random sample of 3000 pixel pairs is taken in both original and encrypted images. The horizontal correlation coefficient of a basic color c is denoted as $r_{h,c}$ then, the vertical correlation coefficient and the diagonal for the primary color c are expressed as $r_{v,c}$ and $r_{d,c}$.

Table 1 The DFT test results applied to encrypted images of Figs. 3 and 6 ($\surd$ accepted and χ rejected).

Test name	Significance label $\alpha = 0.01$	*P*-value/decision				
			Figure 6			
		Figure 3	(a)	(b)	(c)	(d)
Spectral DTF	Red	0.33/$\surd$	0.37/$\surd$	0.54/$\surd$	0.90/$\surd$	0.95/$\surd$
	Green	0.14/$\surd$	0.56/$\surd$	0.99/$\surd$	0.95/$\surd$	0.89/$\surd$
	Blue	0.66/$\surd$	0.41/$\surd$	0.75/$\surd$	0.93/$\surd$	0.49/$\surd$

Table 2 The proposal test results applying to encrypt images of Figs. 3 and 6 ($\surd$ accepted and χ rejected).

Test name	Significance label $\alpha = 0.01$	*P*-value/decision				
			Figure 6			
		Figure 3	(a)	(b)	(c)	(d)
Proposal test	Red	0.33/$\surd$	0.53/$\surd$	0.18/$\surd$	0.12/$\surd$	0.52/$\surd$
	Green	0.37/$\surd$	0.91/$\surd$	0.37/$\surd$	0.08/$\surd$	0.33/$\surd$
	Blue	0.59/$\surd$	0.38/$\surd$	0.15/$\surd$	0.61/$\surd$	0.67/$\surd$

The correlation coefficient results for the images of the Figs. 3 and 6 are shown in Table 4, and the coefficients for the same images encrypted by k are shown in Table 5. The average values of the entropy and the furthest number from 8 are presented in Table 6, regarding number 8 as perfect randomness. Table 7 shows the average amounts of the correlation coefficient and the biggest absolute value or furthest from zero for the same coefficient in the three directions and for each basic color. The absolute value of the correlation coefficient furthest from zero means that it is the worst case.

7.2 *Randomness Results for the Encrypted Images with Damage*

The aspect of analyzing encrypted figures with failure, either voluntarily or not, is pending. In this sense, it is important to mention some aspects before proceeding. The first is with respect to the damage size. In this investigation, failures of 40% with respect to whole encrypted images are handled. The second point concerns how to make the damage. In this paper, the concentric rectangles are used, as presented in Fig. 8.

When an encrypted figure with damage is decrypted, such as that illustrated in Fig. 6, the decoded image with failure has a higher degree of disorder in its bits than the original image, that is, the Fig. 8 clause (c) has a higher randomness degree in its bits than the Fig. 8 clause (a). To measure the randomness degree, χ^2 is used. In this vein, the τ value fulfills the following inequality: $0 \leq \tau \leq 1$. Thus, if the pixels' randomness increases then χ^2 decreases, in fact, if the bits distribution was totally random, then $\chi^2 = 0$. So, the τ value for each basic color of the encrypted image with failure measures how much these colors are separated from the original image colors. In other words, how much information

Table 3 Entropy of encrypted images using the k key for Figs. 3 and 6.

Entropy	Figure 3	Figure 6			
		(a)	(b)	(c)	(d)
Red	7.99958	7.99930	7.99924	7.99922	7.99930
Green	7.99963	7.99938	7.99928	7.99920	7.99927
Blue	7.99965	7.99927	7.99923	7.99931	7.99932

Table 4 Correlation coefficients; horizontal, vertical, and diagonal of the three basic colors for images of Figs. 3 and 6.

Color	Correlation coefficient	Figure 3	Figure 6 (a)	Figure 6 (b)	Figure 6 (c)	Figure 6 (d)
Red	Horizontal	0.63	0.86	0.89	0.97	0.99
	Vertical	0.73	0.77	0.95	0.98	0.99
	Diagonal	0.55	0.74	0.88	0.96	0.98
Green	Horizontal	0.63	0.91	0.89	0.97	0.97
	Vertical	0.73	0.85	0.95	0.98	0.97
	Diagonal	0.54	0.84	0.88	0.96	0.96
Blue	Horizontal	0.54	0.91	0.89	0.95	0.97
	Vertical	0.73	0.87	0.95	0.97	0.97
	Diagonal	0.56	0.85	0.88	0.92	0.96

has the decrypted image with damage lost with respect to the original image.

Table 8 presents the results with 40% failure for the five images of Figs. 3 and 6. Moreover, the τ value of each basic color is reported. The correlation graphs, or linear relationships, in three directions are shown: horizontal, vertical, and diagonal for each primary color and for both the original image and the decoded image with damage. The latter is performed for Fig. 6 clause (c). In Fig. 9, these linear relationships are illustrated for original image and in Fig. 10, for decipher image with 40% damage.

Now, it is convenient to show what happens if as in the original image in Fig. 7, a permutation over whole image is applied before it is encrypted. Then, it is ciphered and later is damaged as shown in Fig. 7; at the end, it is decrypted with the failure. The result is illustrated in Fig. 11.

Table 5 Correlation coefficients for horizontal, vertical, and diagonal of the three basic colors for encrypted images using *k* key for Figs. 3 and 6.

Color	Correlation coefficient	Figure 3	Figure 6 (a)	Figure 6 (b)	Figure 6 (c)	Figure 6 (d)
Red	Horizontal	0.013	0.006	0.003	0.029	0.008
	Vertical	0.001	0.021	0.043	0.013	0.037
	Diagonal	0.023	0.016	0.038	0.024	0.013
Green	Horizontal	0.001	0.004	0.021	0.029	0.050
	Vertical	0.019	0.019	0.026	0.014	0.018
	Diagonal	0.031	0.002	0.013	0.006	0.012
Blue	Horizontal	0.019	0.015	0.018	0.014	0.016
	Vertical	0.028	0.001	0.011	0.019	0.002
	Diagonal	0.027	0.006	0.011	0.013	0.003

Table 6 Average and the furthest value from 8 for the entropy using image (c) of Fig. 6.

Entropy	Average value	The furthest value of 8
Red	7.99929	7.99915
Green	7.99929	7.99914
Blue	7.99929	7.99914

8 Results Discussion

This section carried out the results analysis, separated them into two parts, namely, in the first part, the encrypted images randomness used the two-steps procedure in question. That is, in the first, a permutation was applied to the whole image and in the second, the figure permuted image was encrypted with AES cryptosystem with variable permutations. The second part will address the results' analysis when the encrypted figures are damaged. In this vein, the discussion is started with the encrypted image randomness for a particular key proposal, which passed all the tests suggested in this paper. However, the results of 100,000 keys that approved the proposed test were observed. Picture (c) of Fig. 6 is used for this purpose.

Subsequently, for these keys, the average entropy for each basic color was calculated and the furthest value of 8 for each primary color is also reported, i.e., the furthest amount from the perfect randomness. The averages for the basic colors were presented in Table 6 and, as can be seen, these quantities are very close to 8. Likewise, the furthest values from 8 for the primary colors are very close to 8. This means that the

Table 7 Average and the furthest value from 0 for the correlation coefficient using image (c) of Fig. 6.

Color	Correlation coefficient	Average	The furthest value of 0
Red	Horizontal	0.0154	0.0832
	Vertical	0.0154	0.0830
	Diagonal	0.0153	0.0876
Green	Horizontal	0.0153	0.0894
	Vertical	0.0153	0.0833
	Diagonal	0.0153	0.0838
Blue	Horizontal	0.0153	0.0875
	Vertical	0.0153	0.0948
	Diagonal	0.0153	0.0931

Table 8 τ ratio for images in Figs. 3 and 6 with 40% damage.

		Figure 6			
Color	Figure 3	(a)	(b)	(c)	(d)
Red	0.35	0.35	0.35	0.34	0.35
Green	0.35	0.35	0.34	0.35	0.35
Blue	0.35	0.36	0.35	0.35	0.35

encrypted figures have a random distribution in their bits for each of the basic colors.

In regard to the analysis of the correlation coefficient between adjacent pixels in the horizontal, vertical, or diagonal directions, it is expected that in a "good encrypted" image, adjacent pixels have a correlation coefficient close to zero.[4] For the 100,000 images encrypted in Fig. 6 clause (c), the average values of the correlation coefficient in the three directions and for each of the primary colors were reported in Table 7. The amounts found were close to zero.

The biggest amounts looked for the correlation coefficients, whose absolute values were the largest, taking into account the 100,000 observations reported in Table 7 for the three directions and the three basic colors. These amounts are the furthest from zero, which means that even in the worst cases, these images have correlation coefficients near to zero.

With regard to figures decrypted with damage, the $\tau = \chi^2_{x\%,c}/\chi^2_{0\%,c}$ parameter was used, which gives us an idea of the percentage of information that remains of the original image. Taking into account that a figure deciphered with damage has more noise than the original, this makes the distribution of the bits of each of the primary colors more random than those in the original image. Therefore, $\chi^2_{x\%,c} < \chi^2_{0\%,c}$. Then, if τ is close to zero, this means that all information of the original image is lost, but if it is close to 1 it would mean the opposite. The size of the damage is 40% in this investigation, but it may be higher or lower depending on the "sharpness" desired in the figure decrypted with failure. The τ value is around 35% when the damage is 40%.

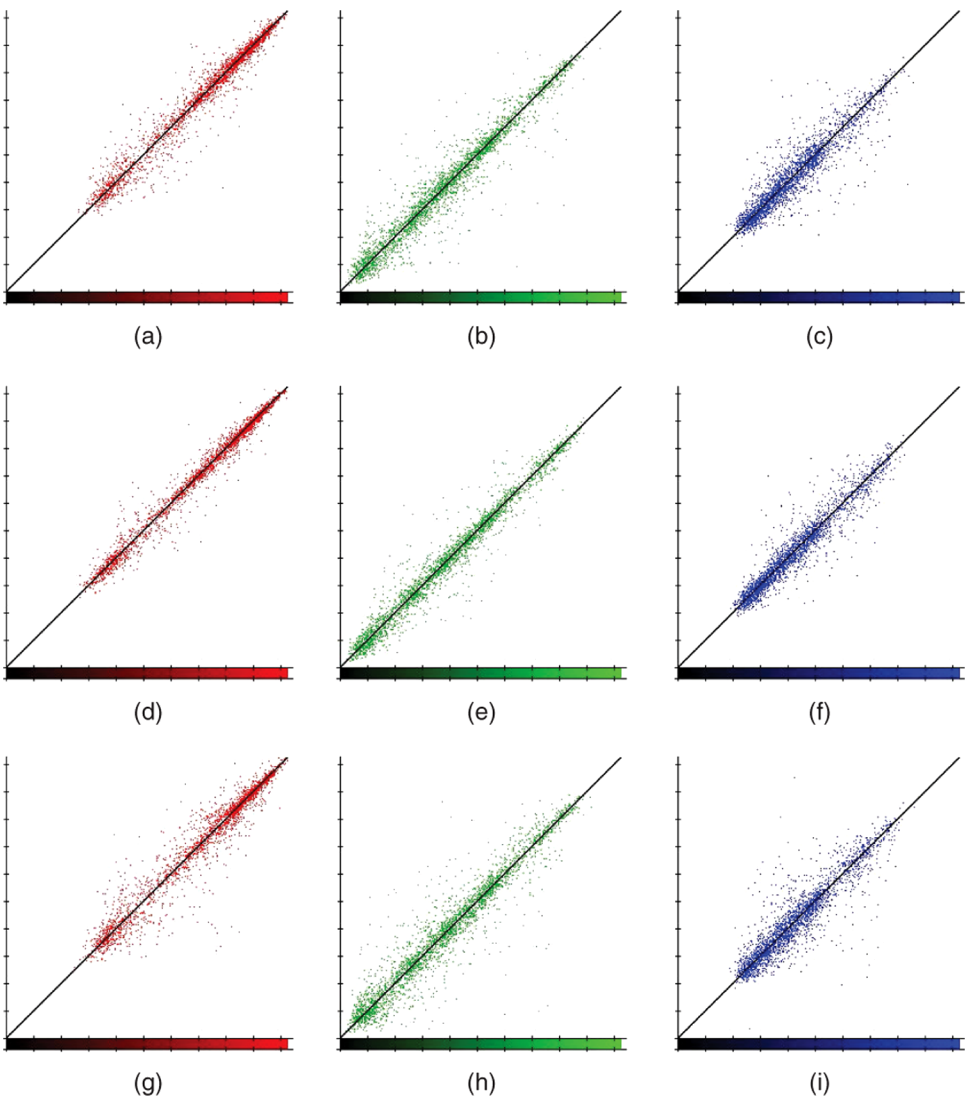

Fig. 9 The correlations: horizontal (a), (b), (c); vertical (d), (e), (f); and diagonal (g), (h), (i) for original image of Fig. 6 clause (c). The columns correspond to red, green, and blue colors.

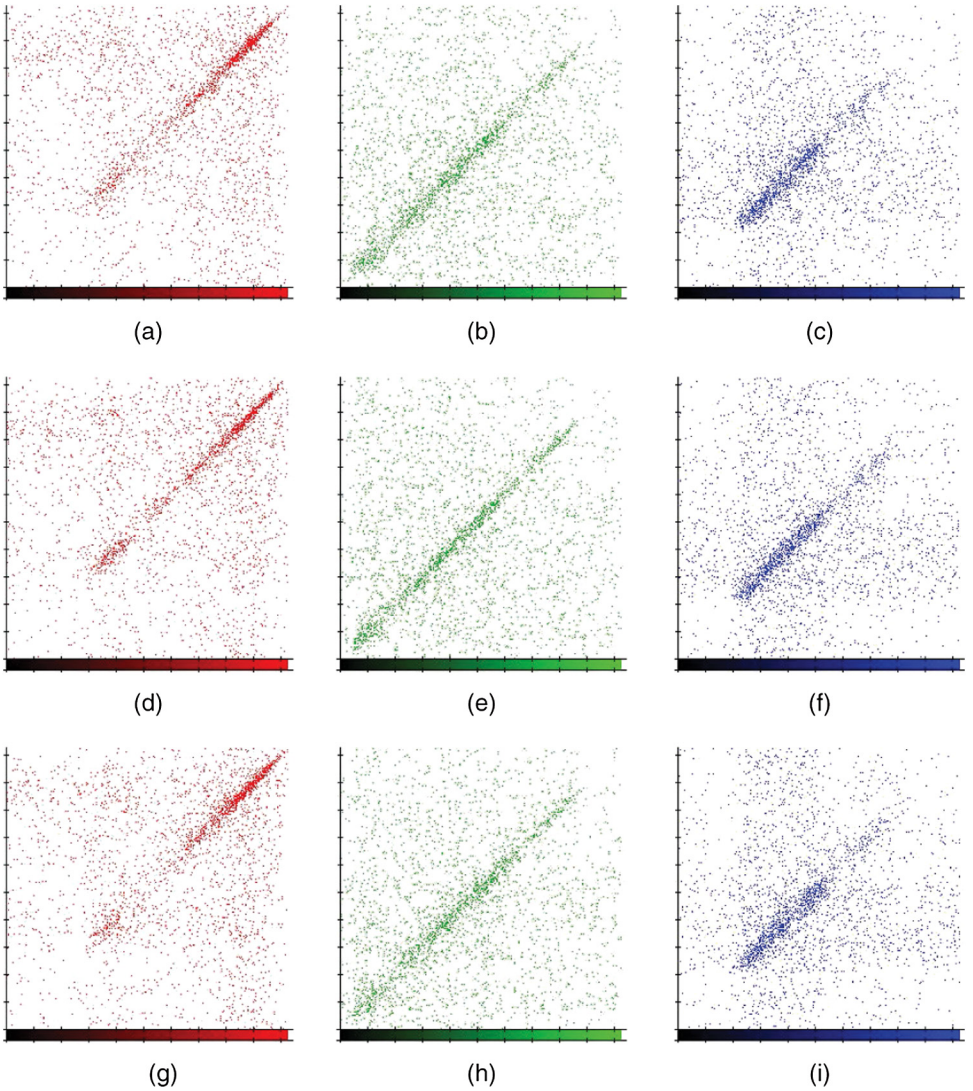

Fig. 10 The correlations: horizontal (a), (b), (c); vertical (d), (e), (f); and diagonal (g), (h), (i) for decrypted image of Fig. 6 clause (c) with 40% of damage. The columns correspond to red, green, and blue colors.

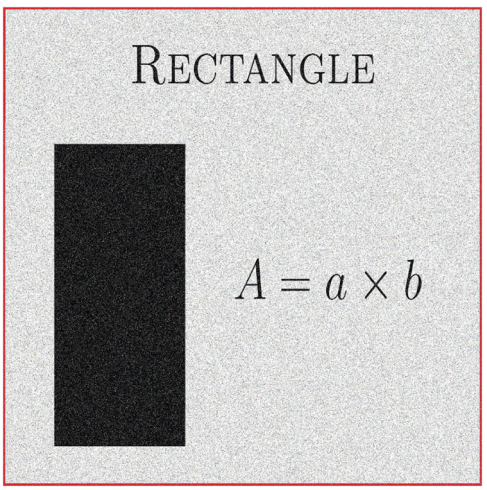

Fig. 11 Cipher image with initial permutation and decipher with damage.

9 Conclusions

This paper has built an algorithm which defines a bijective function between the nonnegative integer and permutation sets. Using the transcendental number pi and this algorithm, it is possible to construct a pseudorandom permutation over 250,000 array positions or more in less than 10 ms. The permutation on the whole original image before the encryption stage is intended to disperse the pixels, thus, when the encrypted file is damaged and later is decrypted, the result does not present the failure in a focalized way as shown in Fig. 7. This is important, because sometimes decisions have to be made quickly, that is, there is no time to wait for an answer later or what is called in real time.[39]

The encrypted images pass the entire randomness test applied for a particular key proposal and reported in Sec. 7.1. These randomness tests are: DFT, entropy, correlation coefficient in: horizontal, vertical, and diagonal directions and the proposal test. Indeed, for entropy, the results are better than other studies.[4]

One hundred thousand keys were chosen and applied to Fig. 6 clause (c), whose only requirement was to pass the

proposed test. It showed the average entropy for each basic color for these keys, and also the correlation coefficients in three directions and for the three primary colors. The results confirmed the randomness of the encrypted images.

Section 7.2 carried out the analysis of encrypted images with damage, noting that the size of the encrypted images failure is 40%, see Fig. 8. This analysis uses the ratio τ for each of the primary colors. In fact, the τ value measures the amount of information lost with respect to the original image for each basic color.

Finally, it is reported that the images' encryption times of Fig. 6 are about 85 ms. The software was developed in C++ language and an intel core i7 was used.

Acknowledgments

The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, CIDETEC, ESCOM and ESFM), the CONACyT, and SNI for their economical support to develop this work.

References

1. J. Li and L. Gan, "Study on chaotic cryptosystem for digital image encryption," in *Third Int. Conf. Measuring Technology and Mechatronics Automation*, pp. 426–430, IEEE, Shangshai, China (2011).
2. L. Xuemei, X. Tong, and L. Dai, "A novel scheme reality preserving image encryption," in *Third Int. Conf. Measuring Technology and Mechatronics Automation*, pp. 218–221, IEEE, Shangshai, China (2011).
3. C. Fu et al., "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Opt. Express* **20** (3), 2363–2378 (2012).
4. H. Zhu, C. Zhao, and X. Zhang, "A novel image encryption-compression scheme using hyper-chaos and Chinese remainder theorem," *Signal Process. Image Commun.* **28**, 670–680 (2013).
5. FIPS PUB 197, *Federal Information Processing Standards Publications*, NIST (2001).
6. D. R. Stinson, *Cryptography: Theory and Practice*, Chapman & Hall/CRC Press, New York (2005).
7. E. Biham and A. Shamir, "Differential cryptanalysis of the full 16-round DES," *Lect. Notes Comput. Sci.* **740**, 494–502 (1993).
8. M. Matsui, "Linear cryptanalysis for DES cipher," *Lect. Notes Comput. Sci.* **765**, 386–397 (1994).
9. J. Daemen and V. Rijmen, "AES proposal: Rijndael, AES algorithm submission," in *FIPS 197*, NIST (1999).
10. C. Carlet, "On highly nonlinear S-boxes and their inability to thwart DPA attacks," in *6th Int. Conf. on Cryptology of the Springer-Verlag*, pp. 49–62, Springer-Verlag (2005).
11. FIPS PUB 46-3, *Federal Information Processing Standards Publication*, NIST (1999).
12. S. Keshari and S. Gopal-Modani, "Color image encryption scheme based on 4-weighted fractional Fourier transform," *J. Electron. Imaging* **21**, 033018 (2012).
13. L. Zhengjun et al., "Image encryption by using gyrator transform and Arnold transform," *J. Electron. Imaging* **20**, 013020 (2011).
14. C. Wen, C. Xudong, and C. J. R. Sheppard, "Optical color-image encryption and synthesis using coherent diffractive imaging in the Fresnel domain," *Opt. Express* **20**(4), 3853–3865 (2012).
15. P. Refregier and B. Javidi, "Optical image encryption based on input plane and Fourier plane random encoding," *Opt. Lett.* **20**, 767–769 (1995).
16. C. Wen and C. Xudong, "Double random phase encoding using phase reservation and compression," *J. Opt.* **16**, 025402–025409 (2014).
17. C. Wen and C. Xudong, "Ghost imaging for three-dimensional optical security," *Appl. Phys. Lett.* **103**, 221106 (2013).
18. C. Wen, J. Bahram, and C. Xudong, "Advances in optical security systems," *Adv. Opt. Photonics* **6**, 120–155 (2014).
19. W. Yue et al., "Image encryption using the two-dimensional logistic chaotic map," *J. Electron. Imaging* **21**, 013014 (2012).
20. D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: the case of AES," *Ext. Version* **3860**, 11–20 (2005).
21. A. Rukhin et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," in *NIST 800-22*, NIST (2010).
22. R. Wolpe and R. Myers, *Probability and Statistics for Engineers and Scientists*, Prentice Hall, México (2007).
23. Nom-151, *Norma Oficial Mexicana NOM-151-SCFI-2002, Prácticas comerciales—Requisitos que deben observarse para la conservación de mensajes de datos*, SCFI (2002).
24. T. M., *Apostol, Análisis Matemático*, Editorial Revert, Barcelona (1994).
25. S. Michael, *Calculus: Cálculo Infinitesimal*, Reverte, Barcelona Espanã (1993).
26. FIPS PUB, *Federal Information Processing Standards Publication 180-3*, NIST (2008).
27. Alexander J. Yee and Shigeru Kondo, "5 Trillion Digits of Pi," 17 October 2011, http://www.numberworld.org/misc_runs/pi-5t/details.html (18 December 2014).
28. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.* **27**, 377–423, 623–656 (1948).
29. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions, Applied Mathematics Series*, Vol. 55, National Bureau of Standards, New York (1964).
30. J. Gallian, *Contemporary Abstract Algebra*, 7th ed., Brooks/Cole, California (2011).
31. A. Gómez, *Enciclopedia de la Seguridad Informática*, Alfaomega, México (2007).
32. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory* **31**, 469–472 (1985).
33. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Commun. ACM* **21**, 120–126 (1978).
34. R. Flores-Carapia, V. M. Silva-García, and C. Rentería-Márquez, "Monte Carlo scheme: cryptography application," *Appl. Math. Sci.* **6** (136), 6761–6768 (2012).
35. C. Grinstead and L. Snell, *Introduction to Probability*, American Mathematical Society (1997).
36. J. Devore, *Probabilidad y Estadística: para ingeniería y ciencias*, 6th ed., International Thompson, México (2005).
37. R. Azarderakhsh and A. Reyhani-Masoleh, "Parallel and high-speed computations of elliptic curve crytography using hybrid-double multipliers," *IEEE Trans. Parallel Distrib. Syst.* **PP**(99), 1–11 (2014).
38. H. Jost, *El detalle en la tipografía*, Comgrafic, Barcelona (2008).
39. S. Baruah et al., "Scheduling real-time mixed-criticality jobs," *IEEE Trans. Comput.* **61**(8), 1140–1152 (2012).

Victor Manuel Silva-García is a research fellow at the Innovation Center in Computer and Technologic Development (CIDETEC). He has a PhD degree in computer science. He belongs to the Researchers National System and is a member of the computer network at the National Polytechnic Institute. He was a director of CIDETEC from 2005 to 2012.

Rolando Flores-Carapia received his ScD degree from the National Polytechnic Institute in 2011. He is a professor in the CIDETEC-IPN. His research interests include image processing and cryptography.

Carlos Rentería-Márquez is a research fellow at the Faculty of Physics and Mathematics (ESFM) of the National Polytechnic Institute. He has a PhD degree in mathematics, belongs to the Researchers National System level III, and is a member of the Mathematical Society México. The topics of investigation and teaching are modern algebra, commutative algebra, and coding theory.

Benjamín Luna-Benoso received his PhD degree in computer science in 2011 from the Computing Research Center, México. He is a professor in the School of Computing (ESCOM). He is a member of the computer network at the National Polytechnic Institute. His research interests include image processing, pattern recognition, and cellular automata.

Cesar Antonio Jiménez-Vázquez received his ScM degree in computer technology from CIDETEC-IPN. He is a project management and technical leader in security systems from Indra company. His research interests include efficient arithmetic for cryptographic algorithms, side-channel security, image encryption using ECC, and PKI applications.

Marlon David González-Ramírez received his ScM degree in computer technology from the Innovation Center in Computer and Technologic Development and is a computer network specialist. He is a project management and Java developer for development and management associated with institutional projects. He is a consultant, teacher, and researcher, and a software tester. He has been a head of the Computer Network Research since 2010.