Cryptosystem Based on the Elliptic Curve With a High Degree of Resistance to Damage on the Encrypted Images

VÍCTOR MANUEL SILVA-GARCÍA¹, ROLANDO FLORES-CARAPIA¹, (Member, IEEE), MARLON DAVID GONZÁLEZ-RAMÍREZ², EDUARDO VEGA-ALVARADO¹, AND MIGUEL GABRIEL VILLARREAL-CERVANTES¹, (Member, IEEE)

¹Instituto Politécnico Nacional, CIDETEC, Mexico City 07738, Mexico
²Instituto Politécnico Nacional, ESIME, Mexico City 07738, Mexico

Corresponding author: Marlon David González-Ramírez (dgonzalezr@ipn.mx)

This work was supported by Instituto Politécnico Nacional of México (Comisión de Operación y Fomento de Actividades Académicas del IPN).

ABSTRACT In this work, a novel symmetric cryptosystem for image encryption is presented. The Symmetric Cryptosystem of the Elliptic Curve (SCEC) can resist damage to encrypted figures, up to 40% of the original figure. SCEC uses chaos to generate an $8 \times 8$ S-box with high nonlinearity to avoid the linear attack. A random permutation is used before starting encryption, making the cryptosystem more robust. For testing, damage according to four types of noise was applied to the encrypted images: additive, multiplicative, Gaussian, or occlusion. The median filter was applied to correct the damage in encrypted images, improving its sharpness, and a new measure, the Similarity Parameter (SP), is proposed to evaluate the difference between the original image and the decrypted image with damage. Several parameters and tests were applied to evaluate the performance of SCEC, from the encryption quality to the resistance to the differential attack. Experimental results indicate that SCEC has high-quality cryptographic properties, very much similar to the corresponding values of AES but with the addition of a high protection to noise damage on the encrypted images.

INDEX TERMS Elliptic curve, noise in encrypted images, chaos, S-box, median filter.

I. INTRODUCTION

Because of the advanced development of the communication networks and the high quantity of valuable information that can be contained in an image, the image ciphering for protecting confidential data is a field with a highly dynamic development [1]–[6].

Four points have to be considered when designing an encryption system for images, being the first one the resistance to attacks. For this work, attacks can be classified into three categories: those against the elliptic curve, those focused on the cryptosystem, and those directed to the encrypted images. Since SCEC is constructed using an elliptic curve, it is sensible to the attack of the discrete logarithm [7], [8]. The discrete logarithm problem can be compared with the problem of factoring a positive integer $n$ in the Rivest-Shamir-Adleman cryptosystem (RSA). In this sense, to solve the discrete logarithm problem when the solution set of the elliptic curve has a prime factor of $2^{256}$, is equivalent to factoring $n$ with a $2^{3072}$ size [9]. However, an elliptic curve can be generated whose number of solutions has a prime factor of $2^{512}$, following the flow chart of Fig.1. This is equivalent to factoring $n = 2^{15000}$ in the RSA scheme, which is higher than the current RSA version [10]. On the other hand, a brute force attack to SCEC when the key set has $2^{512}$ elements presents a higher complexity than a similar attack to the Advanced Encryption Standard (AES-256) [11].

The second point is related to the construction of an S-box that can resist the linear attack [12]. An $8 \times 8$ S-box build by chaos is used, with a Differential Power Analysis (DPA) $< 10$ [13], a value lower than the corresponding to the S-box of AES. SCEC is resistant to the linear attack, as it will be later
explained. Chaos is generated using Eq.(1), where \( f(x, P(x)) \) is presented in Eq.(2) and \( a, b > 1 \) [14].

\[
\frac{dP(x)}{dx} = f(x, P(x)) \quad (1)
\]

\[
f(x, P(x)) = aP(x) - bP(x)^2 \quad (2)
\]

An \( 8 \times 8 \) S-box is a permutation of elements from an array of 256 numbers, in this case, the hex values from \( 00_h \) to \( FF_h \). An algorithm to produce permutations was developed, which defines a one-to-one function [15]. The Walsh function is used to calculate the nonlinearity of the S-box [16].

The third point corresponds to the protection against damage on the ciphered images. If a communication scheme suffers an attack and the image gets damaged, the information could be unreadable depending on the damage extension and the applied cryptosystem [17]. As an example, AES working in CBC mode does not resist an additive noise attack to the encrypted image [18], [19]. SCEC can stand a certain damage degree over the encrypted image produced by additive, multiplicative, Gaussian, or occlusion noise [20], in such a way that when deciphering the information, it is possible to distinguish at a glance the original image if it has been damaged [21]. The damage degree can be up to 40% the size of the encrypted image. The parameters Number-of-Pixels Change Rate (NPCR), Unified Average Changing Intensity (UACI) and Avalanche Criteria (AC) were applied to determine the resistance of SCEC to the differential attack [22].

For evaluating the behavior of SCEC, a median filter of \( 3 \times 3 \) was applied to improve the sharpness of the decrypted images with damage [23]. The proposed symmetric cryptosystem is of the Substitution Permutation Network type with a random permutation at the start of the image encryption [24], which increases the complexity to attack the cryptosystem.

The fourth point tackles the problem of quantifying how different the damaged image is from the original. The UACI parameter is used to solve this, where \( c \) indicates a basic RGB color: red, green and blue, since the similarity percentage is evaluated for each of them. In this work, the parameter \( SP_c \) given by Eq.(3) is proposed to measure the similarity percentage. Finally, this version of SCEC ciphers images without compression in bmp format, since this research was supported by the Mexican government, and its regulation do not allow data loss during the encryption process of sensitive information [25]. However, SCEC can be configured for ciphering compressed images in different formats such as jpg or png, for commercial applications.

\[
SP_c = |100 - UACI_c(2.994)| \quad (3)
\]

In a comparison with recent developments, SCEC presents the following advantages:

- The occlusion damage is tested over the ciphered images in diverse works [17], [26]. In addition to this type of damage, SCEC considers the effect of additive, multiplicative and Gaussian noise sources.
- The number of elements in the key set is particularly important for the security of a cryptosystem. In this sense, the number of keys in SCEC is considerably higher than in other proposals [26], [27].
- Generally, the developments in cryptosystems do not consider any filter to improve the sharpness of the ciphered images that have been damaged [28], [29]. In this work, a median filter is applied to increase the sharpness, and an additional parameter is proposed to measure this feature.
- Another parameter to evaluate the quality of the encrypted images is the entropy. The entropy results generated by SCEC when applied to a series of well-known images surpassed the performance of other cryptosystems [30], [31].
- Because of its implementation using Java-language threads, the execution time of SCEC is considerably lower than in other proposals [32]–[35].

Other cryptosystems (as is the case of AES), could be used in combination with the elliptic curve. However, this is only one of the features of the proposed cryptosystem. SCEC is a novel cryptosystem with additional advantages derived from its implementation, including aspects such as the use of a variable permutation for ciphering and the application of chaos for generating the S-box. Furthermore, AES was not chosen as the base for this development because it is a well-known cryptosystem whose weakness points are public domain.

This article is organized as follows: Section 2 describes the tools used in SCEC for image encryption and for measuring the randomness. Section 3 presents the algorithm for producing the permutation, detailing how an S-box with high non-linearity is generated. In Section 4, the encryption algorithm and the test images are described. Section 5 illustrates the damages that can affect the ciphered images. In Section 6, the randomness results of the ciphered images, and the image damage values are shown. Section 7 focuses on the analysis and discussion of results, while Section 8 corresponds to the conclusions and future work.

**II. THEORETICAL TOOLS APPLIED IN SCEC FOR IMAGE ENCRYPTION**

**A. ELLIPTIC CURVE**

SCEC is a symmetric cryptosystem based on the elliptic curve depicted by Eq.(4):

\[
y^2 \equiv x^3 - kx \mod p \quad (4)
\]

The elliptic curve meets the conditions shown in Eqs.(5,6,7):

\[
4((-k)^3) \neq 0 \mod p \quad (5)
\]

\[
\#E(F_q) \neq 1 \mod p \quad (6)
\]

\[
\#E(F_q) \neq p \quad (7)
\]

where \( \#E(F_q) \) is the number of solutions for the curve in the \( F_q \) field. Eq.(5) guarantees that the curve is non-singular, which implies that it has three different real roots [36].
Eq.(6) implies that the curve is non-supersingular, a condition required to prevent the MOV attack [37]. Eq.(7) states that the curve is not trace one, that is, the Frobenius trace must be different to one since the elliptic curves of trace one present a higher vulnerability [38]. Additionally, the prime factor of the number of solutions must be large enough so the discrete logarithm attack cannot be carried out, at least with the currently available technology [7].

The sum operation (+) is defined on the set of solution points of the curve, so that this set (E, +) is an Abelian group [39]. Theorem 1 is applied to calculate the number of solutions:

**Theorem 1**: Let \( p \) be an odd prime number, \( k \neq 0 \mod p \), and \( \#E(F_p) \) the number of solutions for the elliptic curve defined by Eq.(4). Additionally, \( p \equiv 1 \mod 4 \), where \( p \) can be written as in Eq.(8),

\[
p = a^2 + b^2
\]

such that \( a, b \) are positive integers, \( b \) is an even number, and \( a + b \equiv 1 \mod 4 \). The number of solutions is given by Eq.(9) when \( k \) is not a fourth power \( p \) of some element in the field \( F_p \), but a square power \( p \).

\[
\#E(F_p) = p + 1 + 2a
\]

Consequently, \( k \) must meet the condition of being not a fourth power \( p \) of some element of \( F_p \). For this purpose, Lemma 1 is demonstrated:

**Lemma 1**: Given the elliptic curve of Eq.(4), when the condition \( p \equiv 1 \mod 4 \) is met then \( k \) is a fourth power \( p \) if \( k^{(p-1)/4} \mod p \equiv 1 \).

If \( k^{(p-1)/4} \mod p \equiv 1 \) there is a \( z \in F_p \) such that \( z^4 \equiv k \mod p \). However, \( p \equiv 1 \mod 4 \), then \( (z^4)^{(p-1)/4} \equiv k^{(p-1)/4} \mod p \) with \( (z^4)^{(p-1)/4} = (z)^{(p-1)} \).

Using the little theorem of Fermat, it follows that \( (z)^{(p-1)/4} \mod p \equiv 1 \mod 4 \) [40], implying that \( k^{(p-1)/4} \mod p \equiv 1 \). Thus, \( k^{(p-1)/4} \mod p \neq 1 \) implies that \( k \) is not a fourth power \( p \) of some element in the field \( F_p \).

The Euler criterion is used for finding out if \( k \) is a square power \( p \) of some element in the field \( F_p \) [41]. Lemma 2 is presented to develop a cyclic subgroup in the solution set.

**Lemma 2**: Let the conditions stated in Theorem 1 be met, then \( 4 \mid \#E(F_p) \).

According to the conditions mentioned in Theorem 1, it is known that \( \#E(F_p) = p + 1 + 2a \). On the other hand, since the prime \( p \neq 2 \) and \( b \) is even, it follows that \( a \) is odd. Besides, \( p + 1 + 2a = (a + 1)^2 + b^2 \), which implies that \( a + 1 \) is even.

Thus, \( 4 \mid (a + 1)^2 \) and \( 4 \mid b^2 \) imply \( 4 \mid \#E(F_p) \).

Furthermore, the Theorem 2 gives information to construct a cyclic subgroup in the solution set.

**Theorem 2**: Let \( E \) be an elliptical curve defined on \( Z_q \), where \( p \) is a prime number > 3. Then, there are two positive integers \( n_1, n_2 \) such that there is an isomorphism from \((E, +) \) to \( Z_{n_1} \times Z_{n_2} \). Also, \( n_2 \mid n_1 \) and \( n_2 \mid (p - 1) \).

For this case, \( n_2 \equiv 1 \) and \( n_1 = q \), which is a prime factor of \( \#E(F_p) \), and \( q \) is defined in Eq.(10). Sometimes, \( q \) is prime. However, it is necessary to search for another prime number \( p \) that meets the conditions of Theorem 1, generating a prime \( q \).

\[
q = (p + 1 + 2a)/4 \tag{10}
\]

In this research, the problem of finding the first solution is posed in a different way, namely: an initial point \((x, y)\) is given, and a \( k \) is required that meets Eq.(4). In fact, this \( k \) value must meet Eq.(11),

\[
k \equiv (x^3 - y^2) (x^{-1}) \mod p \tag{11}
\]

The following example illustrates the above mentioned, using particular values:

Given \( a = 197 \) and \( b = 8 \), it follows that \( p = 38873 \). It can be verified that \( p \) mod 4 \( \equiv 1 \), and \( a + b \equiv 1 \mod 4 \). A solution point \( (x, y) = (4934, 9259) \) is selected, and \( k \) is calculated according to Eq.(11) resulting in \( k = 36926 \). Then, \( k \) is verified using Eqs.(12) and (13),

\[
(36926)^{38873−1/4} \text{ mod } 38873 \not\equiv 1 \tag{12}
\]

\[
(36926)^{38873−1/2} \text{ mod } 38873 \equiv 1 \tag{13}
\]

and the resulting curve is: \( y^2 \equiv x^3 - 36926x \mod 38873 \). Thus, according to Theorem 1 the number of solutions is \( \#E(F_{38873}) = 39268 \), and the prime \( q = 392684 \mod 9817 \). On the other hand, 9817 mod 38873 \( \equiv 1 \), \( 4 \equiv (−36926)^3 \mod 38873 \not\equiv 0 \) and \( \#E(F_{9817}) \not\equiv 38873 \). From these conditions, it is concluded that the curve is non-singular, non-supersingular and not even trace one.

The next step is to verify if \( \alpha = (4934, 9259) \) is a primitive element, so \( (q - 1)\alpha = (4934, -9259) \) must be met. The following values are calculated [24]: \( 2\alpha = (30480, 20452) \), \( 4\alpha = (29490, 6305) \), \( 8\alpha = (8271, 26699) \), \( 16\alpha = (13200, 29090) \), \( 32\alpha = (19228, 3105) \), \( 64\alpha = (13363, 23103) \), \( 128\alpha = (9235, 6473) \), \( 256\alpha = (21851, 18746) \), \( 512\alpha = (6148, 34482) \), \( 1024\alpha = (13255, 38103) \), \( 2048\alpha = (23293, 22975) \), \( 4096\alpha = (16811, 7521) \), \( 8192\alpha = (30254, 19663) \). The remaining values are: \( 9216\alpha = (25275, 20077) \), \( 9728\alpha = (2515, 24269) \), \( 9729\alpha = (7738, 19229) \), \( 9808\alpha = (11676, 22000) \) and \( 9816\alpha = (4934, 29614) \). Note that \( 29614 \equiv -9259 \mod 38873 \), so \( 9816\alpha + \alpha = \alpha \).

Fig.1 explains the process to generate the elliptic curve. When the elliptic curve is calculated for real implementations of the proposed system, there is a potential problem derived from the fact that there are some values of \( a \) where the computation program requires a longer time to find the generator element \( \alpha \). Nevertheless, this problem is solved by running the algorithm of the Fig. (1) with several threads to obtain the elliptic curve, proposing for each thread a value of \( a \) chosen in a chaotic way. Additionally, the number of solutions is prime and is calculated according to Eq.(10). It follows that the probability of finding the generating element is \( \frac{1}{4} \), so the failure probability is \( \frac{3}{4} \). Using the binomial distribution, the total fail probability considering 32 threads is presented in Eq.(14). As can be seen, the probability of success is very high, that is, 0.9999, and can be increased if required.

\[
\binom{32}{32} \left( \frac{3}{4} \right)^{32} \left( \frac{1}{4} \right)^0 \approx 0.0001 \tag{14}
\]
B. ENTROPY
The entropy is calculated using Eq.(15) [42]:

\[
H(x) = - \sum_{x \in X} P(x) \log_2 P(x)
\]  

For this case, the images are coded in RGB format and each color is represented with a one-byte value, using 256 combinations. In this sense, if each basic color has a uniform distribution, that is, all points have the same probability, the entropy is 8. However, this is not a sufficient condition for considering a random bit distribution, since theoretically it is possible to construct a non-random string with entropy 8, as is the case of an image with a exact uniform distribution of colors. In practical cases, entropy values close to 8 are desired for the primary color distributions of an encrypted figure [43]. Additionally, other instruments were applied in this work to verify the randomness.

C. CORRELATION COEFFICIENT
The linear analysis between adjacent pixels of encrypted images is performed using the Correlation Coefficient, or correlation. An encrypted image complies with a good randomness degree if the correlation between its adjacent pixels is close to zero. Most image encryption works carry out this analysis in three directions: horizontal, vertical and diagonal [44]. The first step for calculating the correlation between two random variables \( y \) and \( z \) is to randomly select an encrypted image pixel. Every pixel has a value between 0 and 255 for each basic color, denoted as \( y_r \), \( y_g \) and \( y_b \). Then, an adjacent pixel is taken for each direction; horizontal, vertical or diagonal according to the case. The adjacent pixel also has a value for each color, denoted as \( z_r \), \( z_g \) and \( z_b \). Suppose that \( N \) pairs of pixels \( y, z \) are chosen at random. Thus, it is possible to calculate the correlation in the three directions for the three basic colors. The formula for calculating the correlation in the horizontal direction for the red color is given in Eq.(16), and the expressions for \( \bar{y}_r \) and \( \bar{z}_r \) are shown in Eqs.(17) and (18). The expressions for the other basic colors and the remaining directions are similar.

\[
r_{h,y_r,z_r} = \frac{\frac{1}{N} \left( \sum_{i=1}^{N} (y_{i,r} - \bar{y}_r)(z_{i,r} - \bar{z}_r) \right)}{\sqrt{\left( \frac{1}{N} \sum_{i=1}^{N} (y_{i,r} - \bar{y}_r)^2 \right) \left( \frac{1}{N} \sum_{i=1}^{N} (z_{i,r} - \bar{z}_r)^2 \right)}}
\]

\[
\bar{y}_r = \frac{1}{N} \sum_{i=1}^{N} y_{i,r}
\]

\[
\bar{z}_r = \frac{1}{N} \sum_{i=1}^{N} z_{i,r}
\]

D. DISCRETE FOURIER TRANSFORM
The Discrete Fourier Transform (DFT) is a test included in the NIST 800-22 standard, for measuring the randomness degree of a binary chain, that is, there are no repetitive zeros-and-ones patterns, one after another [45]. The parameters involved in the calculation are: \( N_0 \), an expected theoretical quantity given by \((0.95 \times n/2)\), where \( n \) is the string length; and \( N_1 \), the number of values lower than the threshold \( h \), calculated from Eq.(19),

\[
h = \sqrt{\ln \frac{1}{0.05}(n)}
\]

FIGURE 1. The Elliptic Curve flowchart.
Then, \( f_j \) is obtained using Eq.(20), with \( i = \sqrt{-1}, j = 1, 2 \ldots n \) and \( x_k = -1, 1 \). In fact, \( x_k = 2\delta_k - 1 \), where \( \delta_k \) is \( k \)-th bit of the string,

\[
f_j = \sum_{k=1}^{n} x_k e^{\frac{2\pi i(k-1)}{n}}
\] (20)

If \( n \) is odd, the last string bit is deleted, and \( f_j \) is a complex number. The module \( ||f_j|| \) is calculated and compared to \( h \). If \( ||f_j|| < h \), then \( 1 \) is added to \( N_1 \). Otherwise, \( N_1 \) remains the same.

Eq.(21) is evaluated to obtain \( d \) and then calculate Eq.(22), where \( \text{erfc} \) is determined by Eq.(23). The decision rule is: if \( P - \text{value} < 0.01 \) the hypothesis that the string is random is rejected, otherwise it is accepted.

\[
d = \frac{N_1 - N_0}{\sqrt{N_0 N_1 \times 0.05}}
\] (21)

\[
P - \text{value} = \text{erfc} \left( \frac{d}{\sqrt{2}} \right)
\] (22)

\[
\text{erfc} \left( \frac{d}{\sqrt{2}} \right) = 2(1 - \Phi (d))
\] (23)

\section*{E. PARAMETERS TO MEASURE THE STRENGTH OF SCEC AGAINST THE DIFFERENTIAL ATTACK}

As mentioned before, NPCR, UACI, and AC are used to evaluate the strength of the proposed system against the differential attack.

NPCR is defined in Eq.(24), where \( c \) indicates the color, and the function \( D(i, j)_c = 1 \) when the bytes in position \((i, j)\) of the encrypted images 1 and 2 are different. Otherwise, it is 0. The variables \( W \) and \( H \) are the width and height of the image, respectively. An appropriate percentage of this parameter to avoid the differential attack is in the range close to 99.6% [46].

\[
\text{NPCR}_c = \frac{\sum_{i,j} D(i,j)_c}{W \times H} \times 100\% 
\] (24)

The byte \( C_{1,c}(i, j) \) is defined as follows: it has a position \((i, j)\) and the color \( c \) in the first image. Similarly, the byte \( C_{2,c}(i, j) \) has a position \((i, j)\) and a color \( c \) in the second image. With this information, UACI is defined by Eq.(25).

\[
\text{UACI}_c = \frac{1}{W \times H} \left\{ \sum_i \left| C_{1,c}(i, j) - C_{2,c}(i, j) \right| \right\} \times 100\% 
\] (25)

A good percentage of UACI to endure the differential attack is close to 33.4% [47]. The calculation of AC for a particular color is carried out according to Eq.(26), where \( T \) is the total number of bits in the encrypted image, and the function \( b(i, j)_c \) is defined by Eq.(27).

\[
\text{AC}_c = \frac{\sum_i b(i,j)_c}{T} \times 100\% 
\] (26)

\[
b(i,j)_c = \begin{cases} 
0 \\
1 
\end{cases} \quad \text{otherwise, } b(i,j)_c = \begin{cases} 
1 \\
0 
\end{cases}
\] (27)

That is, if a bit in image 1 for color \( c \) is equal to the corresponding bit in image 2 for the same color, then \( b(i,j)_c = 0 \). Otherwise, \( b(i,j)_c = 1 \). An appropriated value of AC to prevent the differential attack is close to 50% [48].

\section*{F. GOODNESS-OF-FIT TEST}

This tool aims to find out if the distributions of the primary colors fit to a uniform distribution [49]. If so, the distribution of the colors is said to be random. However, the above approach leads to a statistical hypothesis test. The test requires two elements, namely: a test statistic and a rejection region. In this research, the statistic \( \chi^2 \) is utilized for each primary color. The \( \chi^2 \) variable distribution is the Chi-square with \( k-1 \) freedom degrees, and is expressed in Eq.(28), where \( o_i \) and \( \exp \) are the observed and expected values, respectively.

\[
\chi^2 = \sum_{i=1}^{k} \frac{(o_i - \exp)^2}{\exp}
\] (28)

It is important to consider the variance of the histogram of the ciphered image [50]. The \( \chi^2 \) test indicates the adjustment of the graph corresponding to the histogram, with respect to a uniform distribution. Additionally, the \( \chi^2 \) test is based on the variance of the frequencies. According to the Central Limit Theorem, the statistic \( \chi^2 \) approaches to the normal distribution with \( \mu = 255 \) and standard deviation \( \sigma \), shown in Eq.(29) [51]. Taking this into account, the threshold can be calculated using the right side of the normal distribution with a significance level of \( \alpha = 0.01 \), which is approximately 308.

\[
\sigma = (2 \times 255)^{0.5} = 22.58
\] (29)

This type of test does not appear in the NIST 800-22 test set to find out the randomness degree of the bit string. That is, the randomness of the tone distribution for the basic colors in the encrypted image is not measured in that standard.

\section*{III. BUILDING ELEMENTS}

This section presents the procedure to generate chaos, and the algorithm for producing permutations.

\section*{A. CHAOS}

The generation of an S-box with the appropriated characteristics for security, including high nonlinearity (\( \geq 100 \)), a Differential Power Analysis (DPA) < 10, and a robustness of differential cryptanalysis \( \geq 0.96 \), among others, can be a difficult issue. However, this complexity can be reduced if chaos is used, this is, choose it in a random way. In SCEC, the logistic map equation shown in Eq.(30) is applied to generate chaos:

\[
y_{n+1} = r y_n (1 - y_n)
\] (30)

The limit of \( y_n \) (if exists) can be calculated when \( n \rightarrow \infty \) in Eq.(30). This limit is expressed by Eq.(31):

\[
\lim_{n \rightarrow \infty} y_{n+1} = \lim_{n \rightarrow \infty} ry_n (1 - y_n)
\] (31)
In practical situations, the values \( y_n \) are stabilized when \( n \leq 1000 \) if the limit exists. However, when \( r = 3.88178128182845... \) 27618 311 digits after the decimal point) the number of possible \( y_n \) is so high that chaos occurs [52]. In this research, that value of \( r \) is taken, and \( x_0 \) is in the range \( 0 < x_0 < 1 \). The values \( y_n \) in Eq.(30) meets the three aspects of chaos:

1) They are deterministic.
2) They have important variations when small changes in \( r \) or \( y_0 \) values are made, i.e., are sensitive to small variations.
3) They cannot be predicted without carrying out the calculation of Eq.(30).

**B. ALGORITHM FOR THE GENERATION OF PERMUTATIONS**

Given a non-negative integer \( m \geq 2 \), then the sets \( \Pi_m = \{ \pi \mid \pi \) is a permutation from array \( 0, 1, \ldots, m-1 \} \) and \( Z_m = \{ n \in N \mid 0 \leq n \leq m! - 1 \} \) can be defined. Besides, using the Euclid division algorithm, any \( n \in Z_m \) can be written in a unique way as in Eq.(32), where the integer \( n \) is expressed on a factorial base and each \( D_i \) is a constant.

\[
n = D_0(m-1)! + D_1(m-2)! + \ldots + D_{m-2}(1)! + D_{m-1}(0)! \tag{32}
\]

The numbers \((m-1)!, (m-2)! \ldots 1!, 0!\) are fixed for a given \( m \), and for simplicity they can be expressed as \((m-i)!\), where \( i = 1, \ldots, m \) represents the factorial base. It will be seen in the algorithm description that \( D_{m-1} = 0 \), and the inequality in Eq.(33) applies:

\[
0 \leq D_i < (m-i) \text{ with } 0 \leq i \leq (m-2) \tag{33}
\]

Once the values \( D_0, D_1, \ldots, D_{m-2} \) are calculated, Algorithm 2 is executed:

**Step 0:** An array in increasing order is generated using Eq.(34):

\[
X[0] = 0, X[1] = 1, \ldots, X[m-1] = m-1 \tag{34}
\]

**Step 1:** Applying Eq.(33), it follows that \( D_0 < m \). Thus, \( X[D_0] \) is an element of the Step 0 arrangement. \( X[D_0] \) is removed from the array and its place is taken by \( X[m-1] \), that is, the last element arrangement, and two operations are performed: elimination and substitution. It means that the other array elements remain unchanged and only the new position of \( X[m-1] \) is assigned. If the value of \( D_0 \) corresponds to the last element position, then \( X[D_0] \) is removed from the array and its position is taken by \( X[m-2] \).

**Step 2:** In the same way as in the previous step, \( D_1 < (m-1) \) using Eq.(33), and \( X[D_1] \) is an element of the Step 1 arrangement. Then, \( X[D_1] \) is removed from the array and replaced with the last element. If \( X[D_1] \) is the last member of the array, proceed as in the previous step.

**Step m-1:** The result up to this step is \( X[D_{m-2}] \) and \( X[D_{m-1}] = k , \) with \( 0 \leq k \leq m - 1 \). The number \( X[D_{m-1}] \) appears automatically because it is the last one; i.e. \( D_{m-1} = 0 \). The positive integer set \( \{ X[D_0], X[D_1], \ldots, X[D_{m-2}], X[D_{m-1}] \} \) is a permutation of the \( 0, 1, \ldots, m-1 \) array. This procedure is performed in \( m-1 \) steps. The complexity of this algorithm is \( O(m) \), because in each step a removal and a substitution of an element are made, and the position of the other elements remains without change.

Algorithm 2 defines a bijective function \( Z_m \mapsto \Pi_m \), which is denoted as \( I_m \). This is stated in Theorem 3:

**Theorem 3:** Given the sets \( Z_m = \{ n \in N \mid 0 \leq n \leq m! - 1 \} \) and \( \Pi_m = \{ \pi \mid \pi \) is a permutation from \( 0, 1, \ldots, m-1 \} \), then Algorithm 2 defines a function \( I_m : Z_m \mapsto \Pi_m \) that is bijective.

**Proof:** First, it is shown that \( I_m \) is a one-to-one function. The reductio ad absurdum method is used for the proof. It is assumed that for \( n_1 \neq n_2 \in Z_m \Rightarrow I_m(n_1) = I_m(n_2) \), but according to Eq.(32), the positive integers \( n_1 \) and \( n_2 \) can be expressed as in Eqs.(35) and (36), respectively:

\[
n_1 = D_{0,1}(m-1)! + D_{1,1}(m-2)! + \ldots + D_{m-2,1}(1)! \tag{35}
\]

\[
n_2 = D_{0,2}(m-1)! + D_{1,2}(m-2)! + \ldots + D_{m-2,2}(1)! \tag{36}
\]

By hypothesis, \( I_m(n_1) = I_m(n_2) \) means that the elements of both permutations were selected in the same way, so that \( D_{0,1} = D_{0,2}, D_{1,1} = D_{1,2}, \ldots, D_{m-2,1} = D_{m-2,2} \) and if this is true then \( n_1 = n_2 \). However, this contradicts the hypothesis, so it follows that if \( n_1 \neq n_2 \in Z_m \Rightarrow I_m(n_1) \neq I_m(n_2) \). This demonstrates that \( I_m \) is a one-to-one function. The function is also onto because the number of elements in both sets \( Z_m \) and \( \Pi_m \) is the same.

**C. S-BOX GENERATION WITH HIGH NONLINEARITY**

From Eq.(30), it is possible to generate numbers with a chaotic form using a certain \( r \) value. In this research, \( r = 3.881 \ldots 7618 \) as previously stated. In spite of there are several possible values that can be assigned to \( r \) to generate chaos, \( r \) has a fix value and \( y_0 \) is variable in the range \( 0 < y_0 < 1 \).

The Java programming language was used for implementing the proposed algorithm. The procedure to generate the boxes is as follows: the \( y_0 \) value is calculated multiplying a prime \( p \) by \( \pi \). Then, 499 digits of the product are taken after the decimal point, and this is \( y_0 \). This process starts with \( p = 2, 3, 5, 7, \ldots \) and so on. The number of iterations \( n \) goes from 200 to 5,000, at most. The result is lower than one, and the digits to the right of the decimal point do not follow any pattern, that is, they appear in a pseudo-random way. Thus, 1000 digits are taken after the decimal point in every iteration.

It is possible because the BigDecimal Java class was applied for defining the number, where the precision given by the number of digits to the right of the decimal point can be arbitrary established as required. It is proposed to take blocks of one byte, one after another, from the decimal point to the right to calculate the constants that are required in expression (32), using the relation \( D_i = b_i \) mod\((256 - i)\), where \( b_i \) is the non-negative integer value associated to the
i-th byte, for \(i = 0, 1, \ldots, 254\). Once the constants \(D_i\) are calculated, Algorithm 2 is applied to obtain a permutation of an 256-element array, taking into account that \(D_{255} = 0\).

A substitution box is a permutation of a 256-position arrangement. After the box is obtained its nonlinearity is calculated by to Eq.(37), where \(NL_i\) is obtained from the Walsh function [53], expressed in Eq.(38):

\[
LN = \min NL_i \tag{37}
\]

\[
NL_i = 2^2 - \frac{1}{2} \max_{b, a \in Z_8} \mathcal{W}_f(b, a) \tag{38}
\]

It is required an S-box with a minimum nonlinearity of 100, considering that the maximum nonlinearity is 128. Additionally, the box must have a DPA < 10, since the propose box is described later, but it is pointed out that each schedule key has the image size. The substitution procedure is applied before the first round. Then, an XOR operation between the first schedule key and the string resulting from the permutation is applied as the first step in the first round. The procedures for generating the permutation and the keys are described later, but in this round each schedule key is the image size. The substitution procedure is applied after performing the XOR operation, using Table 1 according to the FIPS 197 standard, and \(P\) is applied to the result.

IV. ENCRYPTION PROCEDURE
SCEC is a cryptosystem of the Substitution-Permutation-Network type with 14 rounds, which defines a symmetric encryption process [24], using the S-box from Table 1. In Section VII, the characteristics of this box are compared with the original and a variant of the AES S-box, considering the endurance to differential and linear attacks [55], among other parameters. The following is a high-level description of the encryption algorithm:

I) A line of bits randomly selected by the algorithm is added after the last line of the image, before starting the ciphering process by itself. The resulting image is \(513 \times 512\) pixels, considering that the image size in this research is \(512 \times 512\) pixels. The objective is to improve the values of NPCR and UACI, since the key depends on the inserted line.

II) The Sha-512 function is applied to the \(513 \times 512\) image to generate a 512-bit string. It is proposed that the integer associated with this string, \(K\), be the encryption key. Because of the characteristics of the Sha-512 function, every time an image is encrypted the keys are probably different; that is, the probability of collision is very low [33], [56]. So, the number of elements in the key set is \(\approx 2^{512}\).

III) A permutation \(P\) that has the image size is applied before the first round. Then, an XOR operation between the first schedule key and the string resulting from the permutation is applied as the first step in the first round. The procedures for generating the permutation and the keys are described later, but it is pointed out that each schedule key has the image size. The substitution procedure is applied after performing the XOR operation, using Table 1 according to the FIPS 197 standard, and \(P\) is applied to the result.

IV) From the second to the thirteenth round the XOR and substitution operations are applied, in this order.

V) In the fourteenth round, the XOR and substitution operations are applied, in that order. Then, the inverse permutation \(P^{-1}\) and the XOR with the last schedule key are carried out to finish the encryption.

A. PERMUTATION
The image has \(m\) pixels numbered from 0 to \(m - 1\); then, the constants \(D_i\) are calculated to alter the order of the image pixels. It is not necessary to know \(n\) in expression (32). The first schedule key \(k_1\) is considered for calculating \(D_i\). It is divided into one-byte blocks. The first three blocks from \(k_1\) are taken; i.e. 1, 2 and 3, to form a 24-bit string, where \(a_0\) denotes the non-negative integer value of this string. Then, it is proposed to calculate the first constant as \(D_0 = a_0 \mod (m - 0)\), where \(m\) is the number of pixels in the image. It is right shifted to compute the second constant, \(D_1\). In other words, bytes 2, 3 and 4 are taken from the string \(k_1\), in another 24-bit block. Following the same process, \(D_i = a_1 \mod (m - 1)\) and the last constant \(D_{m-1} = 0\). Blocks of 24 bits are used since the intensity resolution in many images is \(2^{24}\). Once the constants \(D_i\) have been calculated, Algorithm 2 is applied to obtain the permutation \(P\).

The elliptical curve encrypts points of itself. In fact, when it is desired to encrypt symmetric-cryptosystem keys, a protocol is required for that encryption. In this sense, the proposed system encrypts only one point, and it is not necessary to elaborate an additional protocol for the distribution of the keys. Furthermore, a secure communication scheme requires two cryptosystems: a symmetric one for ciphering the image, and an asymmetric system for ciphering the key. However, SCEC is simpler since it only requires the elliptic curve. Besides, with the elliptic curve it is possible to use the Diffie-Hellman protocol.
TABLE 1. Substitution table with a nonlinearity of 100.

| Element   | Hex value |
|-----------|-----------|
| 0-15      | e0 41 15 19 4e e4 e5 f3 2b 43 08 81 62 d7 92 63 |
| 16-31     | 4a dc d0 b1 50 23 e3 9b d8 36 9e 7e 76 d1 e9 30 |
| 32-47     | 90 6b b2 88 33 0c e6 29 70 01 0a 0f 15 5f 9f 12 |
| 48-63     | c6 d5 0f 67 6b bb 00 55 4d 1d dd c4 31 ad e1 09 |
| 64-79     | ee 4e 7d 4f ed a3 35 e7 8a 54 8e 05 ff c0 47 |
| 80-95     | 2e de ae 1c cd 5b 9f 72 a6 c8 b6 04 c3 cb 52 75 |
| 96-111    | 4b f1 d5 2a 74 49 3a 87 b4 20 61 98 ec 44 a9 da |
| 112-127   | e2 c7 r2 d2 91 3e bd 9d b8 39 0b 95 7f fc cc 14 |
| 128-143   | a2 db ca 1f 2c 11 9a c5 ce 84 c9 78 57 ba b7 aa |
| 144-159   | 3b 60 af e8 93 a1 ed ea 5d 6e 18 fe 71 2f 17 b3 |
| 160-175   | ac 6c f7 ab 27 d4 f8 13 73 22 06 83 a4 be 66 a8 |
| 176-191   | 94 80 7a 3f 7b 8d f2 88 6a 4c 85 1b 0e fa 37 6f |
| 192-207   | 89 38 24 86 79 59 16 99 58 07 6d 9c b0 fd 28 64 |
| 208-223   | 56 9b 10 5e 2d 5c 34 25 82 6d 32 5a 03 21 02 be |
| 224-239   | a7 7c 68 a5 8f 1e 69 77 cf fc d9 3c fb 42 ef 46 |
| 240-255   | f0 b5 a0 1a 8e 97 45 b9 4f 0d 26 c1 51 48 7d eb |

Generator element:

\[ \alpha = (x_0, y_0) \]

The first schedule key \( k_1 \) is obtained as follows: the point \( K \alpha = P_1 = (x_1, y_1) \) is evaluated, where \( K \) is the positive integer associated to the Sha-512 chain of the image. Then, the points \( P_1 + \alpha, P_1 + 2\alpha, \ldots \) are computed, such that the concatenation of the coordinates for these points is the shortest string, but higher or equal to the image size. If it is higher, the remaining bits are removed to make the string length equal to the image size. Let\( s \) denote this chain as \( D \).

Subsequently, a one-bit circular left shift is proposed for the \( D \)-chain. The resulting string is divided into 8-bit blocks, and the substitution process is applied using the box in Table 1. Let\( s \) call \( D_0 \) the previous result. Another circular left shift is applied, and then the \( S \)-box is used in the same way as above.

The result is denoted as \( D_1 \).

If this process is repeated, the \( D_6 \) chain will be obtained which is proposed as the first schedule key, i.e. \( k_1 \). This decision is further explained in Section VII. The procedure continues to obtain the remaining 14 keys \( k_i \), with \( i = 2, \ldots, 15 \), in the same way: starts with one-bit left rotation using chain \( k_{i-1} \), and then the substitution operation is performed.

On the other hand, it can be observed that \( (x_1, y_1) \) is the base for developing the symmetric cryptosystem. The sender uses the public key of the receiver, say \( Q' \), to encrypt \( (x_1, y_1) \), and later the receiver using their private key, say \( m' \), decrypts the sent point [14]. Also, using the elliptic curve and the value of the Sha-512 function it is possible that the sender signs the information, applying the Elliptic Curve Signature Algorithm - ECSA [57].

Regarding the concatenation of the curve point coordinates, from a given point \( (x_1, y_1) \), it is important to note that each coordinate is written without zeros to the left. That is, it starts with the most significant hexadecimal number.
other than zero, so that at most there will be 3 zeros to the left. In fact, this happens when the number 1 is expressed in hexadecimal. Another relevant point to note is that the \( K \) integer is utilized to obtain the point \( K \times \alpha = (x_1, y_1) \) and only masks the information. In other words, the receiver does not need to know the value of \( K \), since the schedule keys can be calculated by knowing the point \((x_1, y_1)\).

C. IMAGES USED FOR TESTING

The images used for testing SCEC are presented in Fig. 2. These images are 512 \( \times \) 512 pixels, and have been recurrently employed in previous works for their special characteristics [58]. The Donkey and Cameraman images were selected because they are black-and-white pictures, and if a symmetric system is used for encrypting them there is a risk that the encrypted images could not pass the randomness tests proposed in this work. Two more images were employed for testing SCEC: one completely white and another completely black, with all the bits in 1 or in 0, respectively.

![Images](image-url)

**FIGURE 2. Images used for testing SCEC.**

On the other hand, the AES-CBC system is commonly used to encrypt images. However, this mode of encryption do not produce good results when the ciphered image is damaged by noise, as will be seen later. Furthermore, the CBC mode is sequential [24]. For this reason, the cryptographic characteristics of the S-box of SCEC were compared directly to AES, while the performance of SCEC for encrypting images is compared to AES-CBC.

V. NOISE EFFECT ON THE CIPHERED IMAGES

SCEC can stand a certain damage degree produced by noise over the encrypted image. In order to test this capability, different noise sources were applied to the encrypted images to simulate their effect. Four types were considered: Gaussian, additive, multiplicative, and occlusion noise. A new measure denominated Similarity Parameter is implemented in this work, to evaluate this feature. Also, a median filter was applied to the damaged images to complement the testing.

A. NOISE GENERATED BY A GAUSSIAN RANDOM VARIABLE

Mathematical models of the Gaussian noise have been developed in two domains: spatial and frequency. In this work, the standard normal distribution is applied to assign values in the frequency domain, while a uniform distribution is used for choosing points in the spatial domain.

The density function of the normal distribution is expressed in Eq.(40), where \( \mu \) is the mean and \( \sigma \) is the standard deviation. The normal distribution of the random variable \( x \) with parameters \( \mu \) and \( \sigma \) is denoted as \( x \sim N(\mu, \sigma) \).

\[
f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]

A random variable \( z \) has a standard normal distribution if \( z \sim N(0, 1) \). Eq.(41) corresponds to the density function for this particular case:

\[
f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}
\]

1) SPATIAL DOMAIN

The selection of the \((x, y)\) points is carried out as follows: the points are listed as \((0,0) \leftrightarrow 0, \cdots, (0,511) \leftrightarrow 511, (1,0) \leftrightarrow 512, \cdots, (512,511) \leftrightarrow 262655\), since the encrypted images are \(513 \times 512\) pixels size. Then, it is possible to define the set \( W \) by Eq.(42). With this information, a sample is randomly selected from \( W \), with a size up to 40% that of the encrypted image, as it was pointed out previously.

\[
W = \{w|0 \leq w \leq 262655\}
\]

2) FREQUENCY DOMAIN

For the frequency domain, a random variable \( w_c \) is defined in the discrete range \(0, 1, 2, \cdots, 255\), where \( c \) indicates the basic color. A value \( z \) is chosen with a number generator, taking into account that \( z \sim N(0, 1) \). In addition, \( z = -3 \) and \( z = 3 \) are taken if the number generator assigns values lower than \(-3\) or higher than \(3\), respectively. Then, \(-3 \leq z \leq 3\), and the random variable \( w'_c \) is defined according to Eq.(43), considering that this type of noise replaces the original intensities of the image with central values; i.e., integers around 127.5.

\[
w'_c = 127.5 + z \times 42.5
\]
Summarizing, the process for simulating noise generated by a random Gaussian variable over an encrypted figure is as follows:

- The points \((x, y)\) are selected from the map of the encrypted figure, and the number of points depends on the noise degree to simulate.
- Every point has an intensity level assigned for each basic color, denoted as \(w_c\). The procedure is carried out separately for each basic color, within an intensity range from 0 to 255.
- The \(z\)-value is generated according to the standard normal distribution, and \(w'_{c}\) is computed using Eq.(43).
- The value \(w_c\) is replaced by the discrete value of \(w'_{c}\).

### B. ADDITIVE AND MULTIPLICATIVE NOISES

For generating these types of noise, a set of pairs is randomly chosen, each pair formed by a point \((x, y)\) and an integer \(\eta(x, y)\). In the frequency domain, the intensity levels \(w_c\) are converted to \(w'_{c}\). For additive noise, Eq.(44) is used for the conversion, while Eq.(45) is applied for multiplicative noise:

\[
\begin{align*}
    w'_{c}(x, y) &= [w_c(x, y) + \eta(x, y)] \mod 256 \quad (44) \\
    w'_{c}(x, y) &= [w_c(x, y) \times \eta(x, y)] \mod 256 \quad (45)
\end{align*}
\]

### C. OCCLUSION NOISE

For this type of noise, the points \((x, y)\) are selected according to a concentric parallelogram over the encrypted image. Then, the intensity values of the points inside the parallelogram are replaced by a single color. A cherry color was selected for the simulation in this work. This process is equivalent to deleting the information in a specific central area of the encrypted image, as shown in Fig.3.

![FIGURE 3. Occlusion damage in an encrypted image.](image)

### D. MEDIAN FILTER

The median filter is a non-linear type statistical. In this case, a \(3 \times 3\) element mask is used. In general, the filter makes a manipulation in a space \((n \times m)\) of the image pixels. The median algorithm process is as follows: given any image pixel \((x, y)\) the analysis is made in the neighboring elements. For a \(3 \times 3\) mask, the adjacent pixels to \((x, y)\) are shown in Fig.4.

![FIGURE 4. 3 \times 3 mask for the median filter.](image)

The levels of the basic colors for each of the neighbor cells and the image pixel \((x, y)\) are ordered. After the cells are arranged, their median must be higher or equal than the first \(\lceil \frac{n}{2} \rceil\) - 1 elements; (i.e 50%), and lower than the remaining cells. The median for each color is denoted as \(M_{r,i}\), \(M_{g,i}\), and \(M_{b,i}\), respectively.

### E. SIMILARITY PARAMETER

As mentioned before, four types of noise were considered for testing the images encrypted with SCEC, namely: Gaussian, additive, multiplicative, and occlusion. The test began with a damage percentage of 20% of the image size, then it was increased to 30% and 40%, all this carried out for each of the noise types. In this order of ideas, it is important to measure the sharpness-loss percentage for damages in an encrypted image, in respect to the original image. Furthermore, a question arises: what is the sharpness improvement of a damaged image when it passes through a filter? For answering it, a new measure named Similarity Parameter (SP) is proposed in this research. SP is calculated according to Eq.(46):

\[
SP_c = 100\% - \text{UACI}_c(2.994) \quad (46)
\]

In Eq.(46), \(c\) indicates that the measurement is made for each color. On the other hand, since the range of values for UACI is from 0% to around 33.4%, the factor 2.994 is included to let SP cover an approximate range from 0% to 100%. When both images are equal (that is, without encryption) then \(SP = 100\%\). However, if the resulting image is well encrypted \(\text{UACI} \approx 33.4\) and \(SP \approx 0\), indicating that there is no similarity between them. As can be seen, SP gives an accurate idea of the damage impact on the sharpness, as well as the improvement when a filter is applied.

### VI. TEST PROCEDURE AND RESULTS

First, the results of images ciphered without damage are presented, followed by the results of images with damage. The instruments for measuring the randomness are divided in
two: those that present a deterministic result, such as entropy, correlation, NPCR, UACI and AC; and those that perform a hypothesis test: the Discrete Fourier Transform and the Goodness-of-Fit test. As an example of the testing, Fig. 5 presents the Lena original image, and its encryption using SCEC.

A. ENTROPY AND CORRELATION

Table 2 presents the results of entropy for the encrypted testing images. Likewise, the entropy values for the 15 schedule keys used for the Cameraman image are shown in Table 3. The first key is obtained from \((x_1, y_1)\) using \(K\), which is the integer associated to the Has Sha-512 chain of this image.

The values of the correlation are presented in Table 4. The NPCR value appears in Table 5, for each basic color.

| TABLE 2. Entropy of the testing images after encryption. |
| Entropy | Donkey | Cameraman | Jet | Lena |
|---------|--------|-----------|-----|------|
| Red     | 7.99934 | 7.99936   | 7.99937 | 7.99934 |
| Green   | 7.99926 | 7.99933   | 7.99932 | 7.99936 |
| Blue    | 7.99922 | 7.99924   | 7.99914 | 7.99939 |

| TABLE 3. Entropy of the keys used for encrypting the Cameraman image. |
| Key | Entropy |
|-----|---------|
| 1   | 7.99976 |
| 2   | 7.99977 |
| 3   | 7.99977 |
| 4   | 7.99977 |
| 5   | 7.99973 |

| TABLE 4. Correlation coefficient of the testing images after encryption. |
| Color     | Correlation | Donkey | Cameraman | Jet | Lena |
|-----------|-------------|--------|-----------|-----|------|
| Red       | Horizontal  | 0.00288 | 0.00540   | 0.00582 | 0.00342 |
|           | Vertical    | 0.00659 | 0.01335   | 0.00800 | 0.00499 |
|           | Diagonal    | 0.01242 | 0.00147   | 0.00606 | 0.01581 |
| Green     | Horizontal  | 0.01697 | 0.00477   | 0.01179 | 0.01132 |
|           | Vertical    | 0.01665 | 0.00040   | 0.00169 | 0.00668 |
|           | Diagonal    | 0.00514 | 0.00092   | 0.01326 | 0.00100 |
| Blue      | Horizontal  | 0.00093 | 0.00180   | 0.00121 | 0.00033 |
|           | Vertical    | 0.00697 | 0.00708   | 0.00039 | 0.00506 |
|           | Diagonal    | 0.00423 | 0.00200   | 0.00418 | 0.00247 |

Similarly, Table 6 and Table 7 shown the values of UACI and AC, respectively.

B. DISCRETE FOURIER TRANSFORM AND THE PROPOSAL TEST

The results of the randomness in the ciphered images measured with DFT are presented in Table 8. As previously explained, an additional test named Goodness-of-Fit using the \(\chi^2\) distribution was proposed for this measurement. Table 9 shows the results of this test for the four encrypted images.

C. TEST IMAGES COMPLETELY BLACK OR WHITE

The following experiment was carried out in this section: two images were encrypted, one completely black and another completely white, and the resulting values of NPCR, UACI and AC parameters are reported in Table 10.
TABLE 10. NPCR, UACI and AC values for the completely black and completely white images.

| Parameter | Color | Black Image | White Image |
|-----------|-------|-------------|-------------|
| NPCR      | Red   | 99.621      | 99.607      |
|           | Green | 99.617      | 99.613      |
|           | Blue  | 99.604      | 99.609      |
| UACI      | Red   | 33.476      | 33.493      |
|           | Green | 33.394      | 33.477      |
|           | Blue  | 33.453      | 33.472      |
| AC        | Red   | 50.035      | 50.023      |
|           | Green | 50.031      | 49.958      |
|           | Blue  | 49.946      | 50.049      |

D. RESULTS OF THE ENCRYPTED IMAGES WITH DAMAGE

This section presents the results when noise was applied to the encrypted images to simulate damage. It starts presenting the results of images encrypted with the AES-CBC mode and a percentage of noise. Two encryption cases are considered, one with additive noise and another with occlusion. Fig. 6a presents the original Lena image. Subsequently, this image is encrypted with AES-CBC mode with an additive noise equivalent to a 35% of the image size, applied to the encrypted image. Then, it is deciphered and the result appears in Fig. 6b.

Fig. 7 illustrates the second case; i.e., the image is encrypted according with AES-CBC mode and occlusion noise of 40% of the image size. This same process was carried out encrypting the image with the proposed algorithm. Fig. 8 shows the image of Lena encrypted and decrypted with 40% damage applying occlusion noise, using SCEC.

In Section V, it was pointed out that a $3 \times 3$ median filter would be used to improve the sharpness in the encrypted images with damage. In this vein, Fig. 9a presents the Jet image decrypted with SCEC, after additive noise of 40% was inserted to the encrypted image. Subsequently, the filter was applied to the decrypted image with damage, and the result is shown in Fig. 9b.

Table 11 presents the value of SP with different percentages of additive noise damage, for the encrypted test images. Table 12 shows the SP values after applying the filter, inserting a 40% damage for the four noise types. The results in both tables were generated by SCEC.

E. SENSITIVITY

In this research, the sensitivity is measured by using the correlation, and the objective is to prove that there is no
TABLE 12. SP when the 3 × 3 median filter was applied to encrypted images with 40% of damage from different noises.

| Color | Noise Type | Donkey | Cameraman | Jet | Lena |
|-------|------------|--------|-----------|-----|------|
| Red   | Occlusion  | 77.84  | 91.83     | 90.18 | 89.80 |
|       | Additive   | 77.55  | 91.14     | 89.95 | 89.70 |
|       | Multiplicative | 78.27  | 91.36     | 90.43 | 90.03 |
|       | Gaussian   | 77.54  | 91.20     | 90.05 | 89.74 |
| Green | Occlusion  | 77.45  | 91.44     | 88.92 | 89.67 |
|       | Additive   | 77.12  | 91.16     | 88.93 | 89.55 |
|       | Multiplicative | 78.12  | 91.48     | 89.17 | 89.79 |
|       | Gaussian   | 77.10  | 91.22     | 88.71 | 89.64 |
| Blue  | Occlusion  | 77.90  | 91.34     | 90.35 | 91.16 |
|       | Additive   | 77.60  | 91.15     | 90.18 | 91.13 |
|       | Multiplicative | 78.23  | 91.41     | 90.47 | 91.25 |
|       | Gaussian   | 77.58  | 91.24     | 90.08 | 91.19 |

TABLE 13. Correlation between two cipher operations applied to the same plain image using $K$ and $K + 1$.

| Color | Lena | Cameraman | Black image | White image |
|-------|------|-----------|-------------|-------------|
| Red   | 0.00872 | 0.01069   | 0.00533     | 0.00789     |
| Green | 0.00587 | 0.00008   | 0.00579     | 0.01027     |
| Blue  | 0.0046  | 0.00116   | 0.00528     | 0.00013     |

relation between two images that were ciphered with very similar (close) keys [61]. So, an attack making small changes on the key can be implemented in the same way a differential attack is performed by small changes in the plain image. The sensitivity is evaluated as the correlation between two cipher operations applied to the same plain image for different S-boxes. The correlation and entropy correlation to the linear and differential attacks is very much similar to AES.

On the other hand, the differential attack over SCEC cannot be performed because NPCR ≈ 99.96%, as well as UACI ≈ 33.4%, and AC ≈ 50%, indicating the system robustness [48]. The same situation occurred when encrypting completely white or black images.

As for the noise attack to encrypted images, it can be affirmed that SCEC resists damage up to 40% of the encrypted image size, from four types of noise: occlusion, additive, multiplicative, and Gaussian. In addition, a 3 × 3 median filter was applied to the decrypted images with damage to improve their sharpness, such that the similarity parameter SP has at least a value of 78%, and in some cases it reached almost a 90% as indicated in Table 12.

The encryption quality of the images is analyzed in two directions: the first considers the entropy, correlation, DFT, and the proposed Goodness-of-fit test, and the second one is referenced to the NPCR, UACI and AC parameters. In both cases, the results show that the encryption is adequate from a security point of view. The correlation and entropy corresponding to some of the ciphered images also were calculated in [59], [60]. The results generated by SCEC are similar to the reported results in those works for the correlation, while in the case of the entropy the values from SCEC surpass them. However, these parameters in SCEC are calculated for each basic color, producing a higher precision to evaluate the performance. As an example, Table 15 presents the corresponding results for the cameronaman image, considering both the horizontal correlation and the entropy.

Finally, the entropy of the schedule keys is analyzed. In Section IV-B it was indicated that the process of shifting and substitution is applied six times before generating the first key. It is because the first strings have a low entropy, that is, 7.9... but from the seventh step and ahead the chains reach an entropy of 7.999... This results are shown in Table 3.

The proposed algorithm was developed using the Java language, taking advantage of the thread programming approach. An equivalent to an internal multitasking can be implemented with this approach, but avoiding the overhead derived from security and data integrity aspects corresponding to the process scheduling. This programming scheme highly reduces the execution time of the tasks that can be parallelized [62], without requiring multiprocessor platforms. Table 16 shows the reported time for different encryption systems, when ciphering a 512 × 512 image.
V. M. Silva-García et al.: Cryptosystem Based on the Elliptic Curve With a High Degree of Resistance to Damage

TABLE 14. Comparison of the cryptographic properties of the proposed S-box.

| Properties                        | AES S-box | Ref. [65] S-box | SCEC S-box | Range of Good Cryptographic Properties |
|-----------------------------------|-----------|-----------------|------------|----------------------------------------|
| $S$-box                           | Balanced  | Balanced        | Balanced   | $n = 0$                                | No Exploitable Bias |
| Non-linearity                     | 112       | 112             | 100        | $120 \geq n \geq 100$                  | High                |
| Correlation immunity              | 0         | 0               | 0          | $n \leq 0$                             | Low                 |
| Balanced                          | 0         | 0               | 0          | $n = 0$                                | Balanced            |
| Algebraic immunity                | 4         | 4               | 4          | $n \leq 4$                             | Low                 |
| Transparency order                | 7.860     | 7.859           | 7.815      | $n < 7.8$                             | Low                 |
| Propagation characteristic        | 0         | 0               | 0          | $n \leq 0$                             | Low                 |
| Number of fixed points            | 0         | 1               | 1          | $n \leq 4$                             | low                 |
| Number of opposite fixed points   | 0         | 0               | 1          | $n \leq 4$                             | low                 |
| Robustness to differential cryptanalysis | 0.984 | 0.981            | 0.961      | $n < 0.98$                            | low                 |
| Signal to Noise Ratio             |           |                 |            |                                        |                     |
| SNR(DPA)/(F)                      | 9.6       | 9.8             | 9.3        | $n < 10$                              | High                |
| Confusion coefficient variance    | 0.111     | 0.101           | 0.124896   | $n \leq 0$                             | Low                 |

TABLE 15. Correlation and entropy values reported for the Cameraman image.

| Parameter            | SCEC       | Ref. [60]  | Ref. [61]  |
|----------------------|------------|------------|------------|
| Correlation          | 0.0047988  | 0.0025     | 0.0062     |
| Entropy              | 7.99931    | 7.9976     | 7.9961     |

TABLE 16. Average ciphering time for a $512 \times 512$ image.

|                      | SCEC [32] | Ref. [33] | Ref. [34] | Ref. [35] |
|----------------------|-----------|-----------|-----------|-----------|
| Time (s)             | 0.2       | 1.708     | 4.750     | 4.828     | 3.253     |

the time required by SCEC for this operation is considerably lower.

VIII. CONCLUSION

In this work, a novel symmetric cryptosystem (SCEC) is presented, for encrypting color images using the elliptic curve with a prime number $p = 2^{565}$. SCEC is highly secure because it can resist the following attacks: linear, discrete logarithm, differential, brute force, and noise affecting the encrypted images. The quality of the encryption was measured according to the following parameters: entropy, correlation, DFT, and Goodness-of-fit test. In addition, the NPCR, UACI, and AC, among other parameters, were also used for evaluating the performance of SCEC from the security point of view. The results are satisfactory, and very much similar to the corresponding values of AES. However, the addition in SCEC of a high protection to different types of noise damage on the encrypted images produces a higher performance than the version of AES for this task, AES-CBC. Also, the encryption of completely black or white images was carried out, obtaining good results in both cases.

It is pointed out that the permutation applied at the beginning of the encryption and its inverse in the last round is variable; that is, it changes in every encryption process because it depends on the first schedule key. For this reason, the complexity to attack the SCEC cryptosystem is highly increased.

An additional advantage of SCEC is its flexibility to be converted from a symmetrical system to a hybrid one, since even its schedule keys can be ciphered without requiring to send the key to the receiver.

As future work it is considered to implement versions of SCEC for ciphering compressed images in different formats such as jpg or png, for commercial applications. Also, it is intended to evaluate the effect of different filter types for improving the sharpness of encrypted images with damages from diverse sources, in order to include one of these filters as a part of the proposed cryptosystem. Finally, the use of tools as the Matlab function GLCM2 is considered for comparison purposes.

ACKNOWLEDGMENT

The authors would like to thank the Instituto Politécnico Nacional de México (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas COFAA, SIP, and CIDETEC), and the CONACYT (SNI) for their support to the development of this work.

REFERENCES

[1] U. Hayat and N. A. Ahmed, “A novel image encryption scheme based on an elliptic curve,” *Signal Process.*, vol. 155, pp. 391–402, Feb. 2019.
[2] Ü. Çavuşoğlu, S. Kaçar, I. Pehliván, and A. Zengin, “Secure image encryption algorithm design using a novel chaos based S-box,” *Chaos, Solitons Fractals*, vol. 95, pp. 92–101, Feb. 2017.
[3] R. I. Abdelfatah, “Secure image transmission using chaotic-enhanced elliptic curve cryptography,” *IEEE Access*, vol. 8, pp. 3875–3890, 2020.
[4] Y. Luo, X. Ouyang, J. Liu, and L. Cao, “An image encryption method based on elliptic curve Elgamal encryption and chaotic systems,” *IEEE Access*, vol. 7, pp. 38507–38522, 2019.
[5] J. Wu, X. Liao, and B. Yang, “Color image encryption based on chaotic systems and elliptic curve ElGamal scheme,” *Signal Process.*, vol. 141, pp. 109–124, Dec. 2017.
K. Shankar, M. Elhoseny, E. Perumal, M. Hayaraja, k. S. Kumar, "An efficient image encryption scheme based on signcryption technique with adaptive elephant herding optimization," in Cybersecurity and Secure Information Systems. Springer, 2019, pp. 31–42.

N. N. Hla and T. M. Aung, "Attack experiments on elliptic curves of prime and binary fields," in Emerging Technologies in Data Mining and Information Security. Springer, 2019, pp. 667–683.

M. Zia and R. Ali, “Crypanalysis and improvement of blind signcryption scheme based on elliptic curve,” Electron. Lett., vol. 55, no. 8, pp. 457–459, Apr. 2019.

C. M. Hemanth and E. Kannan, “Hybrid elliptic curve cryptography using ant colony based authentication system for cloud computing,” J. Eng. Appl. Sci., vol. 10, no. 16, pp. 7273–7279, 2015.

Y. Yarom, D. Genkin, and N. Heninger, “CacheBleed: A timing attack on OpenSSL constant-time RSA,” J. Cryptograph. Eng., vol. 7, no. 2, pp. 99–112, Jun. 2017.

S. Gueron, Advanced Encryption Standard (AES) New Instructions Set, Intel Corp., Santa Clara, CA, USA, 2010.

Z. Liu, S. Han, Q. Wang, W. Li, Y. Liu, and D. Gu, “New insights on linear crypanalysis,” Sci. China Inf. Sci., vol. 63, no. 1, Jan. 2020, Art. no. 121014.

B. B. Pedersen, “Differential power analysis resistant encryption and decryption functions,” Google Patents, 2019.

H. Liu, B. Zhao, and L. Huang, “A remote-sensing image encryption scheme using DNA bases probability and two-dimensional logistic map,” IEEE Access, vol. 7, pp. 65450–65459, 2019.

P. Kerpentop, K. Podlaski, C. Moraga, and R. Stankovic, “New results on reversible Boolean functions having component functions with specified properties,” in Advanced Boolean Techniques. Springer, 2020, pp. 217–236.

S. Kavut and S. Tutdere, “Highly nonlinear (vectorial) Boolean functions that are symmetric under some permutations,” Adv. Math. Commun., vol. 14, no. 1, pp. 127–136, 2020.

Z. Hua, Y. Zhou, and H. Huang, “Cosine-transform-based chaotic system for image encryption,” Inf. Sci., vol. 480, pp. 403–419, Apr. 2019.

V. Shadangi, S. Choudhary, K. Patro, and B. Acharya, “Novel Arnold scrambling based CBC-AES image encryption,” Int. J. Control Theory Appl., vol. 10, no. 15, pp. 93–105, 2017.

W. Barker, “Recommendation for the triple data encryption algorithm (TDEA) block cipher,” NIST Special Publication 800-67, 2008.

J. M. Vilard, M. S. Millán, and E. Pérez- Cabrér, “Oclusion and noise tests on the encrypted image produced by a security system based on a joint transform correlator and the fresnel transform,” J. Phys., Conf. Ser., vol. 1221, Jun. 2019, Art. no. 012046.

X. Wang, H. Zhao, and M. Wang, “A new image encryption algorithm with nonlinear-diffusion based on multiple coupled map lattices,” Opt. Laser Technol., vol. 115, pp. 42–57, Jul. 2019.

M. Alawida, A. Samsudin, J. S. Teh, and R. S. Alkhawaldeh, “A new hybrid digital chaotic system with applications in image encryption,” Signal Process., vol. 160, pp. 45–58, Jul. 2019.

S. Jian, R.-G. H. W. Zhou, and Y. Li, “Improved quantum image median filtering in the spatial domain,” Int. J. Theor. Phys., vol. 58, pp. 2115–2133, Apr. 2019.

D. Sinson and M. Patterson, Cryptography: Theory and Practice, 4th ed. Boca Raton, FL, USA: CRC Press, 2019.

J. G. G. Loza, “NOM-151-SCFI-2002: Uso de la criptografia para la conservación de la información,” Ph.D. dissertation, 2012.

Z. Hua, B. Xu, F. Jin, and H. Huang, “Image encryption using Josephus problem and filtering diffusion,” IEEE Access, vol. 7, pp. 8660–8674, 2019, doi: 10.1109/ACCESS.2018.2890116.

F. J. Farsana and K. Gopakumar, “Speech encryption algorithm based on nonorthonormal quantum state with hyperchaotic keystreams,” Adv. Math. Phys., vol. 2020, pp. 1–12, Jan. 2020.

S. N. Prayatvalsintha and L. Basavaraj, “Performance analysis of transformation and Bogdoven chaotic substitution based image cryptosystem,” Int. J. Electr. Comput. Eng., vol. 10, no. 1, p. 188, 2020.

P. W. Khan and Y. Byun, “A blockchain-based secure image encryption scheme for the industrial Internet of Things,” Entropy, vol. 22, no. 2, p. 175, Feb. 2020.

C. L. Chowdary, P. V. Patel, K. J. Kathrotia, M. Attique, K. Perumal, and M. F. Ijaz, “Analytical study of hybrid techniques for image encryption and decryption,” Sensors, vol. 20, no. 18, p. 5162, Sep. 2020.

F. Masood, J. Ahmad, S. A. Shah, S. S. Jamal, and I. Hussain, “A novel hybrid secure image encryption based on Julia set of fractals and 3D Lorenz chaotic map,” Entropy, vol. 22, no. 3, p. 274, Feb. 2020.

S. Zhu, G. Wang, and C. Zhu, “A secure and fast image encryption scheme based on double chaotic S-boxes,” Entropy, vol. 21, no. 8, p. 790, 2019.

M. Ahmed, E. A. Solami, X. Y. Wang, M. Doja, M. Beg, and A. Alizad, “Crypanalysis of an image encryption algorithm based on combined chaos for a BAN system, and improved scheme using SHA-512 and hyperchaos,” Symmetry, vol. 10, no. 7, p. 266, Jul. 2018.

X.-P. Zhang, R. Gao, H.-W. Chen, Z.-M. Zhao, and J.-Y. Wang, “Efficient image encryption scheme with synchronous substitution and diffusion based on double S-boxes,” Chin. Phys. B, vol. 27, no. 8, Aug. 2018, Art. no. 080701.

X. Wang, U. Čavuşoğlu, S. Kacar, A. Akgul, V.-T. Pham, S. Jafari, F. Alsaidi, and X. Nguyen, “S-box based image encryption application using a chaotic system without equilibrium,” Appl. Sci., vol. 9, no. 4, p. 1781, 2019.

L. C. Washington, Elliptic Curves Number Theory and Cryptography. London, U.K.: Chapman & Hall/CRC, (2003).
[56] A. Maetouq, S. Daud, N. Ahmad, N. Maarop, N. Sjarif, and H. Abas, “Comparison of hash function algorithms against attacks: A review,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 8, pp. 98–103, 2018.

[57] K. A. Nagaty, “A public key cryptosystem and signature scheme based on numerical series,” *Social Netw. Appl. Sci.*, vol. 2, no. 2, Feb. 2020.

[58] S. M. Ismail, L. A. Said, A. G. Radwan, A. H. Madian, and M. F. Abu-ElYazeed, “A novel image encryption system merging fractional-order edge detection and generalized chaotic maps,” *Signal Process.*, vol. 167, Feb. 2020, Art. no. 107280.

[59] S. Zhu and C. Zhu, “Plaintext-related image encryption algorithm based on block structure and five-dimensional chaotic map,” *IEEE Access*, vol. 7, pp. 147106–147118, 2019, doi: 10.1109/ACCESS.2019.2946208.

[60] S. Zhu, C. Zhu, Y. Fu, W. Zhang, and X. Wu, “A secure image encryption scheme with compression-confusion-diffusion structure,” *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 31957–31980, Nov. 2020, doi:10.1007/s11042-020-09699-4.

[61] C. J. Singh and R. A. Singh, “Novel hybrid image encryption (64-bit) based on Rubik cube and block cipher,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 14, no. 7, pp. 184–192, Jul. 2016.

[62] X. Wang, L. Feng, and H. Zhao, “Fast image encryption algorithm based on parallel computing system,” *Inf. Sci.*, vol. 486, pp. 340–358, Jun. 2019.

[63] S. Picek, L. Batina, D. Jakobović, B. Ege, and M. Golub, “S-box, SET, match: A toolbox for S-box analysis,” in *Proc. IFIP Int. Workshop Inf. Secur. Theory Pract.* Berlin, Germany: Springer, 2014, pp. 140–149.

**VÍCTOR MANUEL SILVA-GARCÍA** received the Ph.D. degree in computer science. He is currently a Research Professor with the Center for Innovation and Technological Development in Computing, Instituto Politécnico Nacional, where he is also a Coordinator of the Security and Network Laboratory. He is a member of the Mexican Mathematical Society.

**ROLANDO FLORES-CARAPIA** (Member, IEEE) received the Ph.D. degree from the Instituto Politécnico Nacional in 2011. He is currently a Professor with CIDETEC-IPN and the Researchers National System. His research interests include image processing and cryptography.

**MARLON DAVID GONZÁLEZ-RAMÍREZ** received the bachelor’s degree in computer systems engineering from the Universidad Tecnológica de México, Mexico, in 2004, the master’s degree in computing technology from the Instituto Politécnico Nacional, Mexico, in 2007, where he is currently pursuing the Ph.D. degree in systems engineering. He is currently a full-time Professor with CIDETEC-IPN. His areas of interest are computer security, cryptography, and its applications.

**EDUARDO VEGA-ALVARADO** was born in Mexico City, Mexico, in 1965. He received the B.Sc. degree in communications and electronics engineering from the Escuela Superior de Ingeniería Mecánica y Eléctrica, Mexico, in 1992, the M.Sc. degree in digital systems from the Centro de Investigación Tecnológica en Computación, in Mexico, in 1996, and the Ph.D. degree in computational and electronic systems from the Universidad Autónoma de Tlaxcala, Mexico, in 2017. He is currently a full-time Researcher and a Professor with the Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, Mexico City. His areas of interest include programming, optimization with metaheuristics, the development of hybrid algorithms, and its application on engineering problems.

**MIGUEL GABRIEL VILLARREAL-CERVANTES** (Member, IEEE) received the B.S. degree in electronics engineering from the Instituto Tecnológico de Veracruz, Veracruz, Mexico, in 2003, and the M.Sc. and Ph.D. degrees in electrical engineering from the Center for Research and Advanced Studies, CINVESTAV-IPN, Mexico City, Mexico, in 2005 and 2010, respectively. He is currently a Full Professor of the Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, Mexico City. He is also a member of the National System of Researchers and the Expert Network on Robotics and Mechatronics, IPN. His current research interests include mechatronic design based on optimization (mono- and multi-objective optimization), bio-inspired metaheuristics in the optimal mechatronic design, optimal tuning for the mechatronic system control based on metaheuristics algorithms, including off-line and online strategies, and robotics.