A reliable chaos-based cryptography using Galois field

Cite as: Chaos **31**, 091101 (2021); https://doi.org/10.1063/5.0061639
Submitted: 28 June 2021 • Accepted: 11 August 2021 • Published Online: 01 September 2021

Lucas G. Nardo, Erivelton G. Nepomuceno, Gustavo T. Bastos, et al.

A reliable chaos-based cryptography using Galois field

Cite as: Chaos **31**, 091101 (2021); doi: 10.1063/5.0061639
Submitted: 28 June 2021 · Accepted: 11 August 2021 ·
Published Online: 1 September 2021

Lucas G. Nardo,[1] Erivelton G. Nepomuceno,[2,a)] Gustavo T. Bastos,[3] Thiago A. Santos,[2] Denis N. Butusov,[4] and Janier Arias-Garcia[1]

AFFILIATIONS

[1]Graduate Program in Electrical Engineering, Department of Electronic Engineering, Federal University of Minas Gerais, Belo Horizonte, MG 31270-901, Brazil
[2]Control and Modelling Group (GCOM), Department of Electrical Engineering, Federal University of São João del-Rei, São João del-Rei, MG 36307-352, Brazil
[3]Department of Mathematics and Statistics, Federal University of São João del-Rei, São João del-Rei, MG 36307-352, Brazil
[4]Youth Research Institute, Saint Petersburg Electrotechnical University "LETI," 5, Professora Popova St., Saint Petersburg 197376, Russia

[a)]**Author to whom correspondence should be addressed:** nepomuceno@ieee.org

ABSTRACT

Chaos-based image encryption schemes have been extensively employed over the past few years. Many issues such as the dynamical degradation of digital chaotic systems and information security have been explored, and plenty of successful solutions have also been proposed. However, the impact of finite precision in different hardware and software setups has received little attention. In this work, we have shown that the finite precision error may produce distinct cipher-images on different devices. In order to overcome this problem, we introduce an efficient cryptosystem, in which the chaotic logistic map and the Galois field theory are applied. Our approach passes in the ENT test suite and in several cyberattacks. It also presents an astonishing key space of up to 2^{4096}. Benchmark images have been effectively encrypted and decrypted using dissimilar digital devices.

Great attention has been devoted to chaotic systems on image encryption schemes since the revolutionary Fridrich's paper. Numerous approaches have been proposed to improve chaotic-based encryption algorithms, such as diffusion and confusion methods to increase cipher security and perturbation processes to mitigate chaos degradation. Nevertheless, the impact of finite precision in different devices has not received the same attention. In this paper, we have shown that chaos-based encryption schemes may not work on distinct hardware and software setups since the finite precision error may generate dissimilar key streams. Consequently, there is apprehension about the adoption of chaotic systems in cryptosystems. To overcome this problem, we present a novel chaotic image encryption algorithm based on finite fields. We have adapted the logistic map using the Galois field as a pseudorandom number generator. In such a way, the generated cipher is the same in different devices, not compromising the key stream. Our approach has been successful in the ENT test suite, showing sufficient pseudorandom properties for our scheme. Benchmark image experiments have outperformed obtained results with well-known methods in many cyberattack tests. Furthermore, we have shown that our method can be efficiently applied in different digital devices.

I. INTRODUCTION

The considerable amount of information flowing on the Internet has required efficient and reliable encryption algorithms.[1] This explains the great attention of the scientific community to investigate such algorithms and the development of numerous techniques such as DES (Data Encryption Standard), AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), Twofish, Blowfish,[2] and chaos-based ones.[3]

Chaos-based image encryption has attracted significant interest due to its superior performance when compared to traditional

algorithms.[4,5] Indeed, since the pioneer work by Fridrich,[6] numerous chaotic image encryption schemes have been suggested. For instance, Gao and Chen[7] have proposed a cryptosystem that employs the logistic map to shuffle the image and Chen's hyperchaotic system to encrypt the respective image. Differently, Haroun and Gulliver[8] exploited the three-dimensional discrete Lorenz system as the chaotic element in the encryption algorithm. Taking advantage of the FPGA (Field-Programmable Gate Array) fast processing, Zheng *et al.*[9] have applied an n-dimensional discrete chaotic system to encrypt multi-images. In addition to the chaotic system, Rodríguez-Orozco *et al.*[10] have used voice recognition as an access key, providing more security to the cryptosystem. Furthermore, there are papers that use metaheuristic optimization methods, such as the firefly algorithm,[11] to ensure the security of ciphers.[12] Sinha and Sahu investigated an image encryption algorithm with shuffling, rotation, and swamping processes, where the Chebyshev polynomial with an adaptive firefly technique is used in order to optimize the generation of the secret key.[13]

Özkaynak[14] has reviewed chaos-based image encryption techniques and has criticized many approaches, arguing that they are weak against some cyberattacks. Moreover, Özkaynak has shown a checklist that chaos-based algorithms must comprise. At the same time, other approaches have been developed to improve chaotic-based encryption algorithms, such as diffusion and confusion methods to increase cipher security[15] and perturbation processes to mitigate chaos degradation.[16] Nevertheless, there is little attention on the effect of finite precision in different digital devices. In fact, chaotic systems have been reported to present contrasting behaviors in different software and hardware setups.[17–19]

The application of finite fields[20] in cryptography has been investigated as a possible way to tackle issues related to finite precision. Wang *et al.*[21] have proposed an encryption algorithm that ensures the security of tumor ultrasound images during the transmission based on the Galois field. Shah and Shah[22] have recognized a remarkable impact of the arithmetic properties of a finite field on the security features of symmetric and asymmetric cryptosystems. Shah *et al.*[23] have applied finite fields to an RGB image encryption application. The results, according to the authors, have outperformed previous chaos-based encryption schemes. Moreover, Broumandnia[24] has described an algorithm adopting finite fields in chaotic maps. In that paper, the focus has been on improving the security and speed of encryption using Galois fields simultaneously in the diffusion and confusion operations. However, to our knowledge, no study has yet focused on the effects of finite precision in digital devices, which may result in distinct ciphers. In other words, chaos-based encryption requires reproducibility in different digital devices, but the floating-point standard, such as the IEEE Standard for Floating-Point Arithmetic,[25] does not fulfill this requirement. In this paper, we have shown that finite precision error may generate incompatible key streams in different computers. Hence, many encryption schemes may not work properly. In order to overcome such issues, we have proposed a novel cryptosystem based on a chaotic system and the Galois field[20] to compose a pseudorandom number generator. The chaotic system chosen is the logistic map.[26] The great advantage of the proposed technique is that all the arithmetic operations occur in a field. It means that there is no need for rounding. As a consequence, the reproducibility of the operations is not influenced by differences in computer arithmetic standards across different digital devices, and the cipher to encrypt and decrypt is the same. Three benchmark images have been employed to show the effectiveness of our proposal. Moreover, we also submitted the algorithm in many statistical tests and cyberattacks such as the key space, the ENT test suite, information entropy, the correlation of adjacent pixels, the histogram analysis, the key sensitivity analysis, the differential attack analysis, and reproducibility. These benchmark image experiments have outperformed obtained results with well-known methods. We have also shown that our method can be efficiently applied in different devices.

The remainder of this paper is presented as follows: in Sec. II, a concise presentation about finite fields is described. The encryption algorithm and its performance analysis are shown in Secs. III and IV, respectively. Finally, Sec. V reports the conclusion, implications of this article as well as final remarks on future works.

II. PRELIMINARY CONCEPTS: FINITE FIELDS

Finite field is a well-developed algebraic structure extensively used in several mathematical applications on communication problems. There are plenty of textbooks about this branch of algebra. The definitions and results of this section have been obtained from Refs. 20 and 27. They were slightly altered in order to fit this context.

From now on, we use the symbols "+" and "·" to represent two distinct operations that do not necessarily represent the usual ones.

Definition II.1 *Let G be a set and "·" a binary operation defined on G. The $(G, \cdot)$ is a group if the following three properties hold:*

(i) *For each element a, b, and $c \in G$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.*
(ii) *There is an identity element e in G such that for all $a \in G$, $e \cdot a = a \cdot e = a$, for all $a \in G$.*
(iii) *For each $a \in G$, there is an inverse element $a^{-1} \in G$ such that $a \cdot a^{-1} = a^{-1} \cdot a = e$.*

What's more, if $a \cdot b = b \cdot a$, for all $a, b \in G$, then G is called an Abelian group.

Example II.2 *$(\mathbb{R}, +)$, $(\mathbb{Q} \setminus \{0\}, \cdot)$, and $(M_{n\times m}(\mathbb{R}), +)$ are examples of Abelian groups.*

Definition II.3 *A field $(F, +, \cdot)$ is a set F with two binary operations, denoted by "+" and "·," such that*

(i) *F is an Abelian group under "+" with identity element 0.*
(ii) *The nonzero elements form an Abelian group under "·"*
(iii) *The distributive laws*

$$a \cdot (b + c) = a \cdot b + a \cdot c \text{and} (b + c) \cdot a = b \cdot a + c \cdot a$$

hold to all a, b, and $c \in F$.

Example II.4 *Let $\mathbb{Z}_3$ be the set of integers modulo 3. Without loss of generality, we represent its residue classes/elements as $\mathbb{Z}_3 = \{0, 1, 2\}$. Hence, we get the following operation charts, as shown in Table I.*

In general, $\mathbb{Z}_p$ is a finite field for a p prime. From now on, we denote each finite field with q elements as $GF(q)$ (Galois field), and q is always a power prime, where $q = p^n$, for p a prime integer and $n \geq 1$.

TABLE I. $\mathbb{Z}_3$—Operation charts: we represent "+" on the left and operation "·" on the right, which are taken mod3.

+	0	1	2	·	0	1	2
0	0	1	2	0	0	0	0
1	1	2	0	1	0	1	2
2	2	0	1	2	0	2	1

Thereafter, given a positive integer $n > 1$, we present two distinct ways to construct the finite field $GF(q^n)$ from $GF(q)$. For the next definition, $GF(q)[x]$ means the set (actually, ring) of all polynomials whose coefficients are on $GF(q)$. In this set, it is possible to define addition and product of polynomials, noticing that the coefficients belong to $GF(q)$.

Definition II.5 *A polynomial $p \in GF(q)[x]$ is said to be irreducible over $GF(q)$ if p has a positive degree and $p = bc$, where b and $c \in GF(q)[x]$ imply that either b or c is a constant polynomial.*

In order to construct finite fields with more elements from $GF(q)$, we consider the following result:

Theorem II.6 *For a polynomial $f \in GF(q)[x]$, the residue class ring $GF(q)[x]/\langle f\rangle$ is a field if and only if f is irreducible over $GF(q)$.*

On $GF(q)[x]/\langle f\rangle$, it considers the classical polynomial operations "+" and "·" modulo f. Given $a, b \in GF(q)[x]/\langle f\rangle$, if the product $a \cdot b$ has degree greater than or equal to f-degree, then we replace the polynomial product $a \cdot b$ for its remainder when it is divided by f. Besides that, the polynomial addition runs as usual.

Example II.7 *Let $GF(2)[x]/\langle x^2 + x + 1\rangle$ be the residue class ring. Hence, the only elements (residue classes) in $GF(2)[x]/\langle x^2 + x + 1\rangle$ are 0, 1, x, and $x + 1$. Since $f = x^2 + x + 1$ is irreducible over $GF(2)$, then $GF(2)[x]/\langle f\rangle$ is actually a finite field, as shown in Table II.*

Therefore, $GF(2)[x]/\langle f\rangle = \{0, 1, x, x + 1\}$ is a 4-element finite field.

Additionally, it is possible to employ matrices in order to represent elements of a finite field $GF(q)$.

Definition II.8 *The companion matrix of a polynomial $f = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} + x^n \in GF(q)[x]$ is defined to be*

$$A = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & 0 & \dots & 0 & -a_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -a_{n-1} \end{pmatrix}_{n\times n}. \tag{1}$$

TABLE II. Polynomial addition and product on $GF(2)[x]/\langle f\rangle$. The coefficients are taken mod 2. In particular, the polynomial product is equivalent to its remainder when it is divided by f.

+	0	1	x	$x+1$	·	0	1	x	$x+1$
0	0	1	x	$x+1$	0	0	0	0	0
1	1	0	$x+1$	x	1	0	1	x	$x+1$
x	x	$x+1$	0	1	x	0	x	$x+1$	1
$x+1$	$x+1$	x	1	0	$x+1$	0	$x+1$	1	x

It is well-known that

$$f(A) = a_0I + a_1A + a_2A^2 + \ldots + a_{n-1}A^{n-1} + A^n = 0_{n\times n}, \tag{2}$$

where I denotes the $n \times n$ identity matrix. Thus, given $f \in GF(q)[x]$ irreducible of an n-degree over $GF(q)$ and A its corresponding $n \times n$ companion matrix, the finite field $GF(q^n)$ may also be represented as

$$GF(q)[A] = \left\{a_0I + \cdots + a_{t-1}A^{t-1} : a_i \in GF(q), 0 \le i \le t-1\right\}, \tag{3}$$

whose usual addition and product matrices are developed regarding the condition $f(A) = 0_{n\times n}$.

From the two finite field representations above, it is possible to connect them as

$$g \in GF(q)[x]/\langle f\rangle \leftrightarrow g(A) \in GF(q)[A]. \tag{4}$$

Thus, we are able to identify binary representations of real numbers as polynomials in $GF(2)[x]$ and, from exposed in (4), as matrices over $GF(2)$.

III. THE IMAGE ENCRYPTION SCHEME

The proposed scheme and the generation of the key stream can be summarized in the following steps. It has been elaborated in accordance with Kerckhoffs' principle.[28] It is worth mentioning that we have used standard Matlab algorithms to describe such steps in all digital devices, though this chaos-based image encryption algorithm can be managed in any programming language.

Step 1: Read the plain-image (P_I) with size $H \times W$, where H and W are the height and width of the image, respectively. Each pixel is represented by an 8-bit unsigned integer.

Step 2: Split the plain-image in two blocks (b_1 and b_2) with size $H \times \frac{W}{2}$.

Step 3: For each block, compute a factor ($f_{b_{1,2}}$) defined as

$$f_{b_{1,2}} = \frac{1}{256\left(H \times \frac{W}{2}\right)} \sum_{i=1}^{H} \sum_{j=1}^{W/2} P_I(i,j), \tag{5}$$

where i and j are the coordinates of each pixel in the blocks.

Step 4: Convert the two factors, previously obtained, to binary numbers as

$$c_{1,2} = \text{dec2bin}\left(\frac{f_{b_{1,2}}}{eps}\right), \tag{6}$$

where eps is the machine epsilon and dec2bin is an algorithm to convert decimal integers to binary numbers.

Step 5: For each binary number c_1 and c_2, use only the first 32 bits. Thereafter, they concatenate to each other as

$$f_{P_I} = \text{strcat}(c_1, c_2), \tag{7}$$

where strcat is an algorithm that concatenates strings horizontally. Thus, the factor f_{P_I} is dependent on the plain-image to be encrypted.

Step 6: Choose 64-bit binary numbers for the initial condition x_0 and the bifurcation parameter r of the logistic map. Both

numbers are represented using a 2Q62 fixed-point format. For friendly-user applications, a sequence of 8 ASCII encoded characters may be required and easily changed into a 64-bit string. Our method allows an initial condition of up to 4096 bits. It is important to mention that each bit of these numbers is allocated separately in an array. Thus, f_{P_I}, x_0, and r are a 1×64 vector.

Step 7: Take a 64° irreducible polynomial p. In this cryptosystem, we have chosen the irreducible polynomial $p \in GF(q)[\mathrm{x}]$ as

$$p(\mathrm{x}) = \mathrm{x}^{64} + \mathrm{x}^{63} + \mathrm{x}^{8} + \mathrm{x}^{2} + 1 \tag{8}$$

and make an $n \times n$ companion matrix P. In this case, as the polynomial $p(\mathrm{x})$ has a 64-degree, then P is a 64×64 matrix.

Step 8: In order to obtain a finite field representation of f_{P_I}, x_0, and r, iterate 64 times the following equations:

$$F_{P_I}(k) = (F_{P_I} + f_{P_I}(65-k) \times P^{k-1}) \bmod 2, \tag{9}$$

$$X_0(k) = (X_0 + x_0(65-k) \times P^{k-1}) \bmod 2, \tag{10}$$

$$R(k) = (R + r(65-k) \times P^{k-1}) \bmod 2, \tag{11}$$

where k is the respective iteration and F_{P_I}, X_0 and R are 64×64 matrices, which are $GF(2)[P]$ representations of f_{P_I}, x_0, and r, respectively.

Step 9: Merge the user initial condition X_0 and the factor F_{P_I} as

$$X_0' = (F_{P_I} + X_0) \bmod 2. \tag{12}$$

In such a manner, the initial condition X_0 of the logistic map and the plain-image factor F_{P_I} are linked, making the system resistant to differential attacks.

Step 10: From the original logistic map,[26] we redefined a recursive function using representation as matrices over $GF(2)$,

$$X_{m+1} = (RX_m(1 - X_m)) \bmod 2. \tag{13}$$

Equation (13) should be iterated $\frac{H \times W}{n \times 8}$ times. As each image pixel is represented by an 8-bit unsigned integer, the polynomial $p(\mathrm{x})$ has a 64-degree, and we work with 512×512 images, the number of iterations is $\frac{512 \times 512}{64 \times 8} = 512$. In such a situation, X_{m+1} is a 3D matrix of size $64 \times 64 \times 512$. Usually, chaos-based image encryption schemes iterate the algorithm $H \times W$ times. Thus, an important feature of our cryptosystem is that its computational complexity is reduced.

Step 11: Format the 3D binary matrix K to a 2D matrix with size $(H \times W) \times 8$, as follows:

$$K = \mathrm{reshape}(X, [(H \times W), 8]), \tag{14}$$

where reshape is an algorithm that reshapes an array.

Step 12: Thenceforth, convert the K matrix to decimal values by multiplying K with an array $2^{n_{array}}$, where $n_{array} = [0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7]$. In such a way, the key stream $K \in [0, 255]$.

Step 13: Reshape matrix K into the one that has the same height and width of the image to be encrypted as

$$K = \mathrm{reshape}(K, [H, W]). \tag{15}$$

Step 14: Encrypt the plain-image P_I using the key stream K and the bit-wise XOR operation with all the pixels, as shown in Eq. (16). The result is the cipher-image C_I,

$$C_I(i,j) = P_I(i,j) \oplus K(i,j). \tag{16}$$

The encryption algorithm is described in a flowchart diagram as seen in Fig. 1. The decryption process, as is well-known, is the reverse of the encryption. It can be done by the application of Step 14 again in the cipher-image.

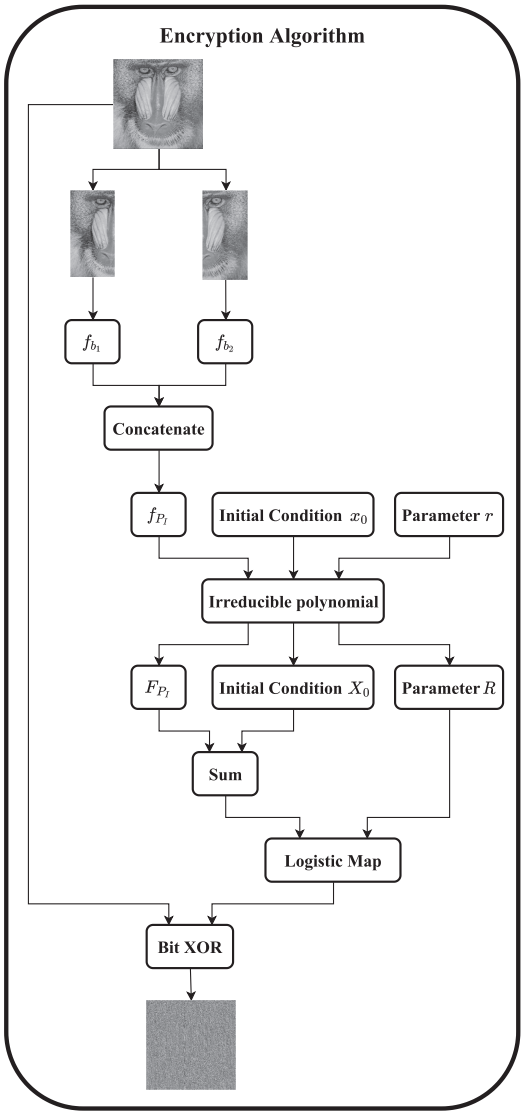

FIG. 1. Flowchart diagram of the algorithm based on the logistic map and the Galois field. The novelty presented here is the orderly encryption process, which can be reproducible in distinct machines. This scheme considers Kerckhoffs' principle,[28] which increases its robustness.

IV. PERFORMANCE ANALYSIS

A series of eight tests was conducted to analyze the security and efficiency of the proposed encryption scheme, namely, key space, ENT test suite, correlation of adjacent pixels, information entropy, histogram analysis, key sensitivity analysis, differential attack analysis, and reproducibility. Additionally, 512 × 512 Baboon, Elaine, and Pepper were used to perform such tests. The parameters as well as the factors, dependent on the plain-images, that were used to generate the key stream are shown in Table III.

A. Key space

An encryption scheme with a key space larger than 2^{100} is considered secure against brute-force attacks.[15,29] In our approach, the seed is represented in a 64 × 64 matrix, implying in an impressive key space of up to 2^{4096}. In our case, this was achieved by using the initial condition (Step 6) and an irreducible polynomial (Step 7). Alternatively, the irreducible polynomial can be fixed and public, and an initial condition of 4096-bits may be required. This can be done by a string of 256 characters using the UTF-16 encoding (MATLAB standard).

B. ENT test suite

The generated keystream K was analyzed with the help of the ENT test suite,[30] a tool that has been extensively used.[31,32] By running six different statistical tests, this series of tests presents a satisfactory indicator of quality for encryption methods, as it can identify pseudorandom features in a sequence. Starting with an input sequence with a bitstream length of 600 000 bits, Table IV shows the result for each test. Since the input sequence passed all tests, the generated ciphers in our scheme deliver acceptable pseudorandom properties.

C. Correlation of adjacent pixels, entropy, and histogram analysis

The correlation of adjacent pixels, information entropy, and histogram analysis are important measures to ensure the quality of a cryptosystem. Correlation among adjacent pixels is high, close to one in a plain-image. Its histogram is non-uniform, and as the plain-image shows information, its entropy is low, indicating low disorder. Otherwise, noise-like images have a low correlation coefficient, close to zero, its histogram is uniform and the entropy is approximately 8, implying high disorder.[1,15,29]

TABLE IV. The results of the ENT test suite[30] for our cryptosystem. This suite is useful for evaluating pseudorandom number generators for encryption and statistical sampling applications. Since the input sequence passed all tests, the generated ciphers in our cryptosystem deliver acceptable pseudorandom properties.

Statistical test	Test output	Result
Entropy	0.999 99 bits per bit	Passed
Optimum compression (OC)	OC would reduce the size of this 600 000 bit file by 0%.	Passed
Chi-square	For 600 000 samples, it is 0.52 and randomly would exceed this value 47.13% times	Passed
Arithmetic mean	0.5005 (0.5 = random)	Passed
Monte Carlo value for Pi	3.169 92 (error 0.90%)	Passed
Serial correlation coefficient	0.000 52 (totally uncorrelated =0.0)	Passed

The correlation coefficients are given by[33]

$$\rho(X, Y) = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}, \tag{17}$$

where μ is mean, E is expectation, and σ is the standard deviation for Y and X.

Information entropy is calculated by[1]

$$J(X) = \sum_{i=0}^{2^8-1} P_i log_2 \frac{1}{P_i}, \tag{18}$$

where $J(X)$ is the entropy in bits, X is an input, and P_i is the likelihood estimation of X.

Figure 2 exhibits the plain-images and the cipher-images, followed by their respective histograms. The proposed cryptosystem is capable of encrypting images efficiently. Furthermore, the histograms display a decrease of pixel variance between the plain and the noise-like images. Table V shows the entropy and correlation coefficients for each tested plain and cipher-image. The obtained results are, as expected, with a correlation coefficient roughly zero and information entropy close to eight bits for cipher-images.

TABLE III. Parameters and the plain-image factors f_{P_I} that were used to perform the encryption algorithm. Binary representations are changed into a matrix over $GF(2)$ according to Definition II.8 before being used in the adapted logistic map. The binary representation uses 2 bits for integer and 30 bits for fraction.

Parameter	Decimal representation	Binary representation
r	3.99	11.111111010111000010100011110101110000101000111101011100001010 01
x_0	0.10	00.000110011001100110011001100110011001100110011001100110011001 10
f_{Baboon}	2.041 025 400 639 708	10.000010101000000001010010000000010000011011000011100001100000000
f_{Elaine}	2.283 803 106 275 167	10.010010001010011101010010000000111111010001111110100000000000000
f_{Pepper}	3.733 667 851 371 541	11.101110111101000110101000000000111100010001110001101000000000000

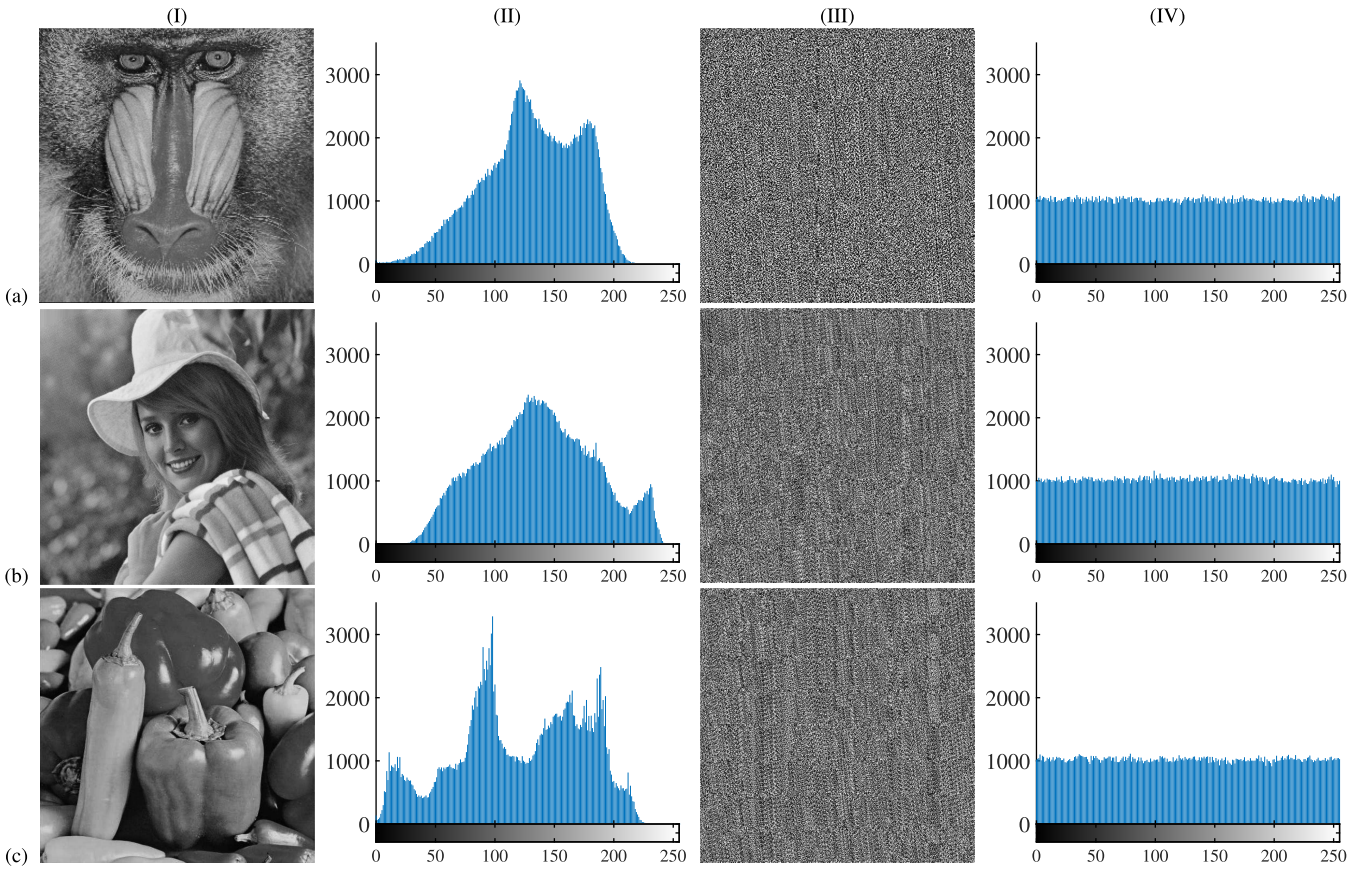

FIG. 2. Respective histograms for plain and encrypted images. The image encryption scheme is able to encrypt meaningful images to noise-like ones. The histograms are, respectively, shown in second and fourth columns for the plain (column I) and encrypted images (column III). The respective histograms show that the cipher-images have a uniform distribution of pixels (column IV), while the plain-image does not (column II), implying a decrease of pixel variance between them.

TABLE V. Correlation coefficients and information entropy for each test image. As expected, the correlation coefficient in horizontal, vertical, and diagonal directions is near to zero for encrypted images. On the other hand, meaningful images present no null values, as their adjacent pixels are strongly correlated. In addition, encrypted images present an entropy $J(X) \approx 8$, implying high disorder. Such results are expected for robust encryption schemes. Image (P) stands for plain and (C) for cipher-image.

	Correlation coefficient			
Image	Horizontal	Vertical	Diagonal	Entropy
Baboon (P)	0.8700	0.8292	0.8002	7.3050
Baboon (C)	0.0075	−0.0005	−0.0038	7.9992
Elaine (P)	0.9726	0.9696	0.9667	7.5130
Elaine (C)	−0.0037	0.0150	−0.0056	7.9991
Pepper (P)	0.9812	0.9837	0.9663	7.5952
Pepper (C)	0.0114	0.0272	−0.0038	7.9992

D. Key sensitivity and differential analysis

Key sensitivity and differential analysis measure the effect on the key stream based on a minor change in the initial conditions and the image that will be encrypted, respectively. Meanwhile, key sensitive analysis indicates that cryptosystems can have numerous

TABLE VI. Results of the key sensitivity analysis for the Baboon image. With a slight disturbance in the initial conditions, the outcome is absolutely dissimilar compared to the result obtained with an initial condition without disturbance. These values are compatible with other works in the literature.[3]

Secret key	$\text{Diff}_1(\%)$	$\text{Diff}_2(\%)$
$0.1 + 10^{-14}$	99.54	99.54
$f_{P_I} + 10^{-14}$	99.60	99.60

TABLE VII. NPCR and UACI results. The encryption scheme is secure against differential attack in accordance with the criteria established in Ref. 36. Thus, the proposed technique is secure against differential attack.

Image	NPCR (%)	UACI (%)	Decision
Baboon	99.68	33.36	Passed
Elaine	99.59	33.51	Passed
Pepper	99.57	33.47	Passed

TABLE VIII. Description of each device used to simulate our proposed encryption scheme and the other two algorithms (Refs. 3 and 39). The same version of MATLAB has been used in Devices 1 and 2.

Device	Description
1—Encryption	Intel® Core™i5-8265U CPU @1.60 GHz
2—Decryption	Intel® Xeon® CPU E5-2620 v4 @2.10 GHz

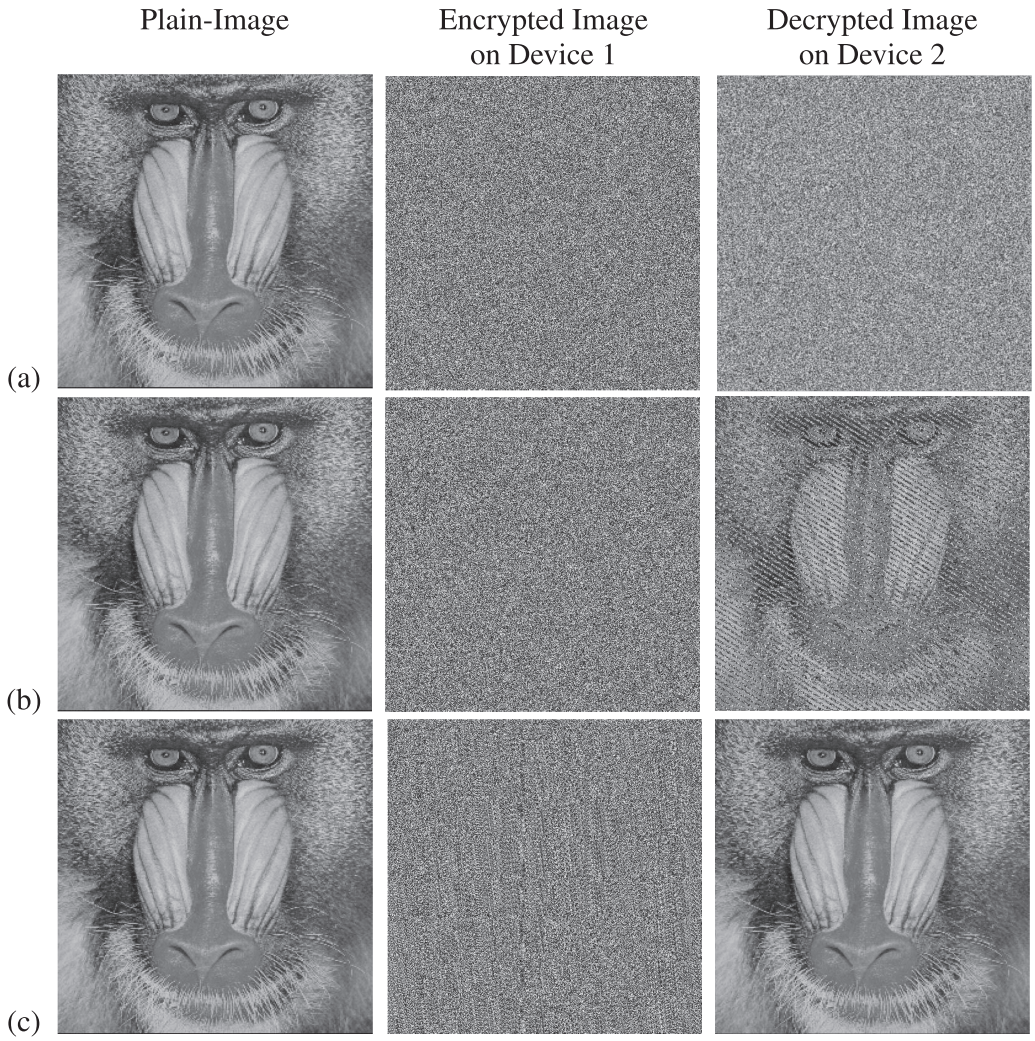

FIG. 3. Reproducibility of chaos-based encryption algorithms on different devices. Line (a) presents an encryption scheme based on finite precision error, as reported in Ref. 3. It was not possible to recover the plain-image on Device 2 after encrypting it on Device 1. Line (b) shows the encryption and decryption results using the algorithm proposed in Ref. 39. Likewise line (a), it was still unable to recover the plain-image. Line (c) exhibits the results applying our technique. Unlike the previous results, after encrypting the image on Device 1 and decrypting it on Device 2, the original image is fully recovered, without data loss. This occurs since our proposed technique performs all the arithmetic operations in a field. It means that there is no need for rounding; consequently, our approach does not have roundoff error.

possible key streams, indicating robustness to this scheme, and the differential attack analysis indicates that whether the encryption algorithm is safe against this type of attack.[34]

First of all, the key sensitivity analysis is accomplished by disturbing either the initial condition x_0 by 10^{-14} or f_{P_I} by 10^{-14}. The encryption process is performed after a disturbance to produce the cipher-image C_2. The difference between the cipher-images is quantified by[35]

$$\mathrm{Diff}_1(\%) = \frac{100}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |sign(C_1(i,j) - C_2(i,j))|, \tag{19}$$

where H and W are the height and width of the cipher-images C_1 (with no perturbed values) and C_2 (with perturbed values).

From the key stream obtained via disturbance, the decryption process is performed using the cipher-image C_1, obtaining P_2. Equation (20) computes the difference between the plain-images P_1 and P_2,[35]

$$\mathrm{Diff}_2(\%) = \frac{100}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |sign(P_1(i,j) - P_2(i,j))|. \tag{20}$$

Table VI shows the differences obtained from distinct key streams encrypting the Baboon image. Results show that the difference is nearly 100%, indicating that the outcome is absolutely dissimilar.

The Number of Changing Pixel Rate (NPCR) and the Unified Average Changed Intensity (UACI) are the two equations to quantify the system vulnerability about differential attack. These equations are described by Eqs. (21) and (22), respectively. In order to execute such an analysis, two plain-images are encrypted. A random chosen pixel of one of the two images is slightly modified by one,[34]

$$NPCR(\%) = \frac{100}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |sign(C_1(i,j) - C_2(i,j))|, \tag{21}$$

$$UACI(\%) = \frac{100}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{|C_1(i,j) - C_2(i,j)|}{255}, \tag{22}$$

where H and W are the height and width of the cipher-image C_i.

As it has been expected, our proposed encryption algorithm is secure against differential attack. Table VII presents the NPCR and UACI the indices from Baboon, Elaine, and Pepper images.

E. Reliability on different devices

The main contribution of this work is solving a recurrent problem when dealing with chaos-based encryption algorithms, where the user may be unable to decrypt the data if the device used has a different hardware and software setup than the one used for the encryption process.[37] Common chaos-based methods apply the normalized raw output of a chaotic system to perform the encryption. Such methods fail in real cases, where users need to exchange encrypted images and be able to decode them in contrasting devices. This is due to finite precision errors, present on computers.[25,38] Thus, distinct processors and software may have slightly diverse ways to calculate the same expression, which is a problem when modeling such sensitive systems.

We were capable of developing a method that is robust against hardware and software discrepancy. Representing binary numbers as elements of a finite field when iterated with the logistic map, we are able to obtain the plain-image without data loss when encrypting the image on Device 1 and decrypting it on Device 2.

Besides our proposed approach, we tested two other methods (Refs. 3 and 39) to demonstrate the problem regarding data exchange when dealing with chaos-based encryption algorithms. Table VIII shows the device configuration used to encrypt and decrypt the plain-image, respectively. Furthermore, Figs. 3(a) and 3(b) demonstrate that the methods cannot decipher the information, while our approach (c) is able to fully recover the image.

TABLE IX. We have compared the performance of our method using the Baboon image (512×512). The suggested scheme holds similar or superior performance than other algorithms outlined in Refs. 40–43.

Criteria	Ours	Ref. 40	Ref. 41	Ref. 42	Ref. 43
Key space	2^{4096}	2^{283}	2^{1024}	5.46×10^{80}	2^{256}
Entropy	7.9992	7.9993	7.9993	7.9993	NA
CC—H	0.0075	−0.0018	−0.0036	−0.0108	0.0220
CC—V	−0.0005	0.0007	0.0001	0.0087	−0.0083
CC—D	−0.0038	−0.0013	0.0023	0.0058	−0.0013
NPCR (%)	99.68	99.61	99.70	99.61	NA
UACI (%)	33.36	33.48	32.28	33.43	NA

F. Performance comparison

In order to validate the performance of our encryption scheme, we have compared the results for the Baboon image with other results found in the literature. Table IX shows such performance comparison. Note that our encryption algorithm has a similar or superior performance in all criteria.

V. CONCLUSION

In this paper, we have proposed a novel image encryption algorithm based on chaos and Galois field theory. Through the adoption of a straightforward and easy-to-implement 1D logistic map, a 64-degree irreducible polynomial on $GF(2)[x]$ and a condition based on the plain-image, the designed chaos-based scheme has proved to be secure against cyberattacks, mainly brute-force attacks, since the algorithm has an astonishing key space of up to 2^{4096}. Moreover, our approach has passed in the ENT test suite, which guarantees acceptable pseudorandom properties. Most notably, our image encryption algorithm has shown reproducibility in distinct digital devices. Additionally, a comparative study with other encryption schemes shows that our approach is just as efficient as others and reliable across distinct software and hardware setups.

ACKNOWLEDGMENTS

Erivelton G. Nepomuceno was supported by Brazilian Research Agencies: CNPq/INERGE (Grant No. 465704/2014-0), CNPq

(Grant Nos. 425509/2018-4 and 311321/2020-8), and FAPEMIG (Grant No. APQ-00870-17).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]Y. Luo, M. Du, and J. Liu, Commun. Nonlinear Sci. Numer. Simul. **20**, 447 (2015).
[2]N. P. Smart, *Choice Reviews Online*, Information Security and Cryptography Vol. 53 (Springer International Publishing, Cham, 2016), pp. 53–4409.
[3]L. G. Nardo, E. G. Nepomuceno, J. Arias-Garcia, and D. N. Butusov, Chaos, Solitons Fractals **123**, 69 (2019).
[4]H. Kwok and W. K. Tang, Chaos, Solitons Fractals **32**, 1518 (2007).
[5]X.-Y. Wang and Q. Wang, Chin. Phys. B **23**, 030503 (2014).
[6]J. Fridrich, in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation* (IEEE, 1997), Vol. 2, pp. 1105–1110.
[7]T. Gao and Z. Chen, Phys. Lett. A **372**, 394 (2008).
[8]M. F. Haroun and T. A. Gulliver, Nonlinear Dyn. **82**, 1523 (2015).
[9]H. Zheng, S. Yu, and X. Xu, Math. Probl. Eng. **2014**, 1.
[10]E. Rodríguez-Orozco, E. García-Guerrero, E. Inzunza-Gonzalez, O. López-Bonilla, A. Flores-Vergara, J. Cárdenas-Valdez, and E. Tlelo-Cuautle, Electronics **7**, 414 (2018).
[11]I. Fister, M. Perc, S. M. Kamal, and I. Fister, Appl. Math. Comput. **252**, 155 (2015).
[12]H. A. Ahmed, M. F. Zolkipli, and M. Ahmad, Neural Comput. Appl. **31**, 7201 (2019).
[13]R. K. Sinha and S. Sahu, J. Intell. Fuzzy Syst. **36**, 4437 (2019).
[14]F. Özkaynak, Nonlinear Dyn. **92**, 305 (2018).
[15]B. Norouzi and S. Mirzakuchaki, Nonlinear Dyn. **78**, 995 (2014).
[16]S. Li, G. Chen, and X. Mou, Int. J. Bifurcation Chaos **15**, 3119 (2005).
[17]E. G. Nepomuceno, S. A. Martins, B. C. Silva, G. F. Amaral, and M. Perc, Appl. Math. Comput. **329**, 408 (2018).
[18]E. G. Nepomuceno and E. M. Mendes, Chaos, Solitons Fractals **95**, 21 (2017).
[19]E. G. Nepomuceno, P. F. Guedes, A. M. Barbosa, M. Perc, and R. Repnik, Int. J. Bifurcation Chaos **29**, 1950112 (2019).
[20]R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*, 1st ed. (Kluwer Academic Publishers, 1987), p. 209.
[21]N. Wang, G. Di, X. Lv, M. Hou, D. Liu, J. Zhang, and X. Duan, IEEE Access **7**, 49945 (2019).
[22]D. Shah and T. Shah, Multimed. Tools Appl. **79**, 28023 (2020).
[23]T. Shah, A. Ali, M. Khan, G. Farooq, and A. A. de Andrade, Wirel. Pers. Commun. **113**, 1201 (2020).
[24]A. Broumandnia, J. Inf. Secur. Appl. **54**, 102553 (2020).
[25]"IEEE Standard for Floating-Point Arithmetic," in *IEEE Std 754-2019 (Revision of IEEE 754-2008)* (Institute of Electrical and Electronics Engineers (IEEE), 2019), pp. 1–84.
[26]R. M. May, Nature **261**, 459 (1976).
[27]R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*, 1st ed. (Cambridge University Press, 1986), p. 415.
[28]S. Mrdovic and B. Perunicic, in *Networks 2008—The 13th International Telecommunications Network Strategy and Planning Symposium* (IEEE, 2008), pp. 1–8.
[29]T. Hu, Y. Liu, L.-H. Gong, and C.-J. Ouyang, Nonlinear Dyn. **87**, 51 (2017).
[30]J. Walker, "ENT: A pseudorandom number sequence test program"; see https://www.fourmilab.ch/random/ (2008).
[31]B. Stoyanov and K. Kordov, Sci. World J. **2014**, 283639.
[32]X. Chen, S. Qian, F. Yu, Z. Zhang, H. Shen, Y. Huang, S. Cai, Z. Deng, Y. Li, and S. Du, Complexity **2020**, 1.
[33]A. V. Diaconu and A. C. Dascalescu, Proc. Rom. Acad. Ser. A Math. Phys. Tech. Sci. Inf. Sci. **18**, 351 (2017).
[34]Z.-H. Gan, X.-L. Chai, D.-J. Han, and Y.-R. Chen, Neural Comput. Appl. **31**, 7111 (2019).
[35]Y. Zhang, IETE Tech. Rev. **33**, 310 (2016).
[36]Y. Wu, J. P. Noonan, and S. Agaian, Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. **April**, 31 (2011).
[37]T. A. Santos, E. P. Magalhães, D. Fiorio, and E. G. Nepomuceno, in *Anais do 14° Simpósio Brasileiro de Automação Inteligente* (Galoa, 2019).
[38]M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic* (Society for Industrial and Applied Mathematics, 2001), p. 121.
[39]Z.-H. Guan, F. Huang, and W. Guan, Phys. Lett. A **346**, 153 (2005).
[40]E. G. Nepomuceno, L. G. Nardo, J. Arias-Garcia, D. N. Butusov, and A. Tutueva, Chaos **29**, 061101 (2019).
[41]B. Mondal, S. Singh, and P. Kumar, J. Inf. Secur. Appl. **45**, 117 (2019).
[42]X. Chai, Y. Chen, and L. Broyde, Opt. Lasers Eng. **88**, 197 (2017).
[43]R. Gupta, R. Pachauri, and A. K. Singh, Int. J. Inf. Secur. Priv. **13**, 53 (2019).