mathematics

MDPI

Article

Color Image Encryption Algorithm Based on a Chaotic Model Using the Modular Discrete Derivative and Langton's Ant

Ernesto Moya-Albor [1,*,†], Andrés Romero-Arellano [1,†], Jorge Brieva [1,*,†] and Sandra L. Gomez-Coronel [2,†]

[1] Facultad de Ingeniería, Universidad Panamericana, Augusto Rodin 498, Ciudad de México 03920, Mexico; 0228652@up.edu.mx
[2] Instituto Politecnico Nacional, UPIITA, Av. IPN No. 2580, Col. La Laguna Ticoman, Ciudad de México 07340, Mexico; sgomezc@ipn.mx
* Correspondence: emoya@up.edu.mx (E.M.-A.); jbrieva@up.edu.mx (J.B.); Tel.: +52-55-5482-1600 (ext. 5210) (E.M.-A.)
† These authors contributed equally to this work.

Abstract: In this work, a color image encryption and decryption algorithm for digital images is presented. It is based on the modular discrete derivative (MDD), a novel technique to encrypt images and efficiently hide visual information. In addition, Langton's ant, which is a two-dimensional universal Turing machine with a high key space, is used. Moreover, a deterministic noise technique that adds security to the MDD is utilized. The proposed hybrid scheme exploits the advantages of MDD and Langton's ant, generating a very secure and reliable encryption algorithm. In this proposal, if the key is known, the original image is recovered without loss. The method has demonstrated high performance through various tests, including statistical analysis (histograms and correlation distributions), entropy, texture analysis, encryption quality, key space assessment, key sensitivity analysis, and robustness to differential attack. The proposed method highlights obtaining chi-square values between 233.951 and 281.687, entropy values between 7.9999225223 and 7.9999355791, PSNR values (in the original and encrypted images) between 8.134 and 9.957, the number of pixel change rate (NPCR) values between 99.60851796% and 99.61054611%, unified average changing intensity (UACI) values between 33.44672377% and 33.47430379%, and a vast range of possible keys $> 5.8459 \times 10^{72}$. On the other hand, an analysis of the sensitivity of the key shows that slight changes to the key do not generate any additional information to decrypt the image. In addition, the proposed method shows a competitive performance against recent works found in the literature.

Keywords: image encryption and decryption; modular discrete derivative; cellular automata; Langton's ant; deterministic noise; chaos theory; security

MSC: 68P25

Citation: Moya-Albor, E.; Romero-Arellano, A.; Brieva, J.; Gomez-Coronel, S.L. Color Image Encryption Algorithm Based on a Chaotic Model Using the Modular Discrete Derivative and Langton's Ant. *Mathematics* **2023**, *11*, 2396. https://doi.org/10.3390/math11102396

Academic Editor: Lingfeng Liu

Received: 4 April 2023
Revised: 6 May 2023
Accepted: 17 May 2023
Published: 22 May 2023

1. Introduction

Image encryption corresponds to a set of cryptography techniques that are used to protect confidential information contained in digital images from unauthorized access. Cryptography emerged as a sub-discipline of both mathematics and computer science, and it has been applied to different disciplines where information security is a key issue. In digital image cryptography, there are specific requirements for the methods that must be fulfilled to ensure the security of the information. For example, these methods must hide the visual information of the encrypted image, generating a high entropy value. The key space must be very large, preventing brute force techniques from being effective. Moreover, the key sensitivity analysis must show a significant security level. The differential attack test, a method used to learn about the secret key that encrypts pairs of plaintext and ciphertext images, must be passed and preserve the visual quality of both the encrypted and decrypted images [1]. In general, image encryption schemes are commonly composed

of two stages: a permutation step to secure visual information and a diffusion operation that changes the value of the pixels to obtain an avalanche effect. The use of both cellular automata (CA) [1] and chaos theory [2] generates secure and robust encryption systems. CA are mathematical systems that consist of cell grids that evolve over time according to a set of rules, while chaotic systems are mathematical models that seemingly exhibit random behavior, unpredictability, and ergodicity properties. These techniques can be combined to create complex and unpredictable behaviors, making it difficult for an attacker to deduce the original version of an image based on its encryption.

Currently, there are cryptographic image methods that use cellular automata and chaos theory-based techniques. For example, Khayyat et al. [3] presented a new blockchain-enabled method referred to as shark smell optimization (SSO), which is used in the Internet of Things (IoT) environments, along with the Hopfield chaotic neural network (HCNN), to guarantee secure encryption. The proposed SSO-HCNN cryptographic scheme employs a composite chaotic map and the SSO algorithm to determine the best possible public and secret keys of the system. HCNN is used in order to generate a self-diffusion chaotic matrix in the diffusion phase, then the keys are used in XOR operations performed by the messy image to obtain the encrypted image. The encryption of the pixel value in the image is stored on the blockchain to guarantee the security and privacy of the images. Li et al. [4] proposed an encryption algorithm that uses chaotic maps and CA to encrypt images. The 2D logistic-sine-coupling map and the logistic-sine-cosine map (LSCM) were initialized with values calculated using SHA-256 of the original image. The diffusion process was then carried out, followed by the key matrices being generated using chaotic maps during the permutation process. The index matrices obtained by sorting each row or column of the key matrices were used to scramble the diffused image. Subsequently, the scrambled image through CA generated a cipher image. The resulting cipher image is resistant to various attacks. Dong et al. [5] utilized two global rules from hybrid elementary cellular automata (ECA) to improve the chaotic behavior of the pseudo-random coupled map lattice approach based on the Chirikov standard map. This resulted in a nonlinear and irreversible model that provides resistance against chosen plaintext/ciphertext attacks. The effectiveness of the proposed scheme has been verified by testing its robustness and efficiency against differential and statistical attacks. In [6], Rupa et al. took a large image and divided it into smaller, pure image components. These components were then permuted using the cellular automata rule and subjected to a second-level transformation involving cross-pattern scanning and circular shift operations. The resulting scrambled image was then divided into smaller, encrypted images. Lv et al. [7] used reversible Life-like cellular automata with balanced rules. This algorithm adopts a classic confusion–diffusion structure at the block level by encrypting the blocks into patterns resembling random noise through the proposed CA. The resulting encryption method demonstrates satisfactory security against image processing attacks and exhibits robustness in the face of data loss and random noise. In [8], Kafetzis et al. described the use of a modified Renyi chaotic map, to define a pseudo-random bit generator (PRBG). This PRBG, in combination with a finite automaton, defines an encryption strategy for plain-text images. Overall, the proposed algorithm uses a combination of chaotic and automaton-based techniques to encrypt gray-scale images. Boudali et al. [9] proposed an algorithm that uses cellular automata and chaotic logistic mapping with an approach to facilitate the progression of configurations in ECA. This was in order to make the resulting encryption more random. The proposed technique outperforms some existing image encryption algorithms. Overall, the algorithm combines cellular automata and chaotic logistic mapping to create a secure and effective method for encrypting multimedia data. In [10], Kang et al. designed cellular automata, referred to as (n, m, k)-PC-MLCA (programmable complemented–maximum length cellular automata). This algorithm is used to encrypt color images through two stages. In the first stage of substitution, the (n, m, k)-PC-MLCA generates nonlinear sequences as encryption keys. In the shuffling step, the image is processed at the row/column level and the block unit is processed using 1D maximum length cellular automata (MLCA) to achieve faster

encryption and decryption methods. In [11], Chong et al. attempted to encrypt color images using cellular automata and deoxyribonucleic acid (DNA) sequences. They converted a color image into DNA matrices by using cellular automata to break the correlation among the various elements within the image. Then, the image was diffused using DNA operations to hide the information. Roy et al. [12] proposed an algorithm referred to as IESCA; it uses cellular automata, referred to as two-dimensional Moore cellular automata (MCA), which is used in resource-constrained IoT devices. The random chaotic sequences are generated by the system through local transformations that rely on the bit states of the cellular automaton's neighbors. It has a higher key space than other CA-based image encryption techniques, and shows better efficiency in performance, computing time, and against differential attacks. Kumar et al. [13] described an encryption algorithm using one-dimensional ECA and the Henon chaotic map. The ECA was used to extract properties that could be used in a cryptographic diffusion process, while the Henon chaotic map was used in a keyed transposition cipher to produce a shuffled image, which has been shown to be resistant to statistical attacks. In [14], Jeelani et al. used cellular automata to scramble digital images. The performance of the algorithm was evaluated in terms of the gray difference degree of the scrambled images. The algorithm's robustness was further assessed by analyzing the correlation coefficient and the rate of pixel change. These measures helped determine the algorithm's ability to withstand potential attacks and maintain data integrity. Alexan et al. [15] described an algorithm in two stages. In the initial phase, rule 30 cellular automata (RCA) was employed, followed by the utilization of a Lorenz system in the subsequent phase. The effectiveness of the algorithm was assessed using various metrics, and the findings demonstrate that it performs similarly to existing schemes in the literature while providing the additional advantage of minimal processing time. This characteristic is highly desirable for real-time applications in the image security field.

In [16], Song et al. propose an encryption algorithm that uses the integer wavelet transform to transform the original image into the frequency domain. To reorganize the pixel positions within the image blocks, they employed a one-dimensional chaotic map. This chaotic map was utilized to obtain diverse reversible cellular automata (RCA). The RCA evolution of the image blocks was executed by using varying rules and iteration durations based on the significance of the information contained within the image blocks. The image was scrambled and diffused to reduce the blocking effect. In [17], Kang et al. use a combination of the You Only Look Once (YOLO) algorithm for extracting the region of interest from the original image, the Chen system for encrypting the detected region of interest (ROI), and a hardware-friendly CA for encrypting the entire image. Gan et al. [18] proposed using a combination of compressive sensing (CS), the Game of Life (GoL), and a 5D memristive hyperchaotic system to encrypt images. The process involves permuting, compressing, and diffusing the plaintext image using the GoL-based scrambling method and CS. The key matrix used in the diffusion process is generated by using chaotic sequences from the 5D memristive hyperchaotic system, which is also used to construct the measurement matrix and generate the initial cell matrix for GoL. In [19], Ping et al. used a combination of cellular automata. During the scrambling stage, the image's rows and columns were simultaneously scrambled using a keystream generated by a 2D logistic-adjusted sine map (LASM). In the diffusion stage, the scrambled image was divided into two identical square-bit matrices, which were then encrypted using a CA. Choi et al. [20] combined a generalized 3D chaotic Arnold's cat map (ACM) with a PC-MLCA to encrypt color images. The PC-MLCA was designed for hardware implementation; with its extended duration and non-linear output, it provides an encryption key through a pseudo-random number generator (PRNG). The image undergoes a simultaneous transformation of pixel positions and color values using the generalized chaotic cat map. This approach enhances the image's resistance to noise and deletion attacks. In [21], Choi et al. proposed a combination of image shuffling and 1D MLCA to encrypt color images. Image shuffling is used to resist distortion and deletion attacks, and the 1D MLCA is used to shuffle the pixel positions of the image.

Naskar et al. [22] employed a combination of key-based block ciphering, shuffling with variable-sized blocks, and elementary cellular automata with a chaotic tent map to encrypt images. The key streams used for ciphering individual blocks varied in size, showing a dependency on the plaintext image and the previous key stream. The ciphered blocks were then shuffled to increase diffusion. The resulting encrypted image had a low level of correlation and a high rate of pixel change compared to the original image, indicating a high level of security. Zhang et al. [23] used a combination of set partitioning in hierarchical trees (SPIHT), cellular automata, and different chaotic systems to encrypt images in a lossless manner. By integrating the encryption process with the data compression process, it is possible to effectively encrypt a small portion of the data without compromising the advantageous coding properties of SPIHT. In [24], Chai et al. suggested a fusion of an ECA chaotic system with various parameters and block compressive sensing (BCS) as an image encryption/compression approach. The plaintext image was first transformed using discrete wavelet transform (DWT) to create four block matrices, representing different frequency components of the image. Subsequently, ECA was utilized to disorder the block matrices, changing the positions of their elements and increasing the confusion of the algorithm. Following this, BCS was applied over the disordered matrices to compress and encrypt them by utilizing measurement matrices. The proposed method provides good security and robustness. Gen et al. [25] proposed an image encryption method by incorporating a finite state machine (FSM) and block scrambling. The algorithm starts by decomposing the original image into four frequency bands through the discrete wavelet transform. Subsequently, a combination of zig-zag scanning curves and chaotic sequences is employed to generate a scrambling matrix. This matrix is utilized to scramble the image by using a combination of chaotic sequences, DNA coding, and automata. Finally, the image is diffused using a key stream to improve security.

In the last decade, other works have shown that methods inspired by cellular automata and chaos theory are very competitive in image encryption. In [26], Eslami and Kabirirad used cellular automata as chaos generators in a block-based image encryption approach. The proposed approach can identify subtle alterations in the encrypted image prior to the decryption process. Qi et al. [27] presented a chaos-based image encryption method using a 2D Henon–Chebyshev map (HCM). The random sequence generated by 2D HCM is used to scramble the pixel positions, which are converted into DNA planes. Then, a 2D DNA-CA is applied to update the DNA planes in each iteration of the 2D HCM stage. The authors of [28] designed an image encryption scheme that incorporates the random fractional discrete cosine transform (RFrDCT) and the Game of Life (GoL) based on chaos theory. In [29], Mondal et al. suggested a chaotic skew tent map and a CA-based encryption image method. The initial 128-bit sequence is generated from the chaotic skew tent map, which is then utilized by CA to generate pseudo-random numbers. These numbers are then utilized to shuffle the pixels of the plaintext image. Subsequently, the scrambled image is encrypted by a random number obtained by the chaotic map. An image encryption method using the logistic sine system (LSS), 2D CA, and an FSM-based DNA rule generator were proposed by Khan et al. [30]. To generate the secret key and initial values for the LSS, the researchers utilized the SHA-256 algorithm. The first stage of their encryption scheme employs the Feistel structure-based bit inversion (FSBI) to modify the pixel values. They then utilize 2D cellular automata with local rules using the structure-based Moore neighborhood. Finally, the image is transformed using a generator of rules based on a DNA finite state machine. In [31], Li et al. reported a mono-spectral image encryption method applied to multispectral color information without dividing it into three color channels. The proposed method can encrypt mono-spectral elemental images (EIs). In addition, linear CA and hyperchaotic encoding methods encode the captured EIs. Seshadhri and Chandrasaker [32] developed an image encryption approach in the hybrid domain based on a logistic map (LM), reversible integer wavelet transform (RIWT), and ECA. Thus, the LM performs a permutation of the pixel positions using CA in the spatial domain and by a random matrix generated by the LM in the transformed domain by RIWT. Ben Slimane et al. [33] defined an image

cryptosystem based on 2D LM and non-uniform CA using the SHA-2 algorithm. In [34], Rajagopalan et al. proposed an encryption approach for color images by using chaotic CA attractors. The authors encrypted the color bands using Lorenz, Lü, and CA based on rule 42. Moreover, scrambling and XOR operations were used to improve security, and the authors generated a random synthetic image to diffuse the three color channels. Ping et al. [35] reported an image encryption technique that combines Life-like CA and the theory of chaos. The method involves two stages: permutation and substitution. In the permutation stage, a 2D LASM is employed. For the substitution stage, a Life-like cellular automaton of second-order is utilized, employing a rule approach. In [36], Rajagopalan et al. proposed an encryption system for color images using a key image triggered by hardware, as well as the Lorenz, Lü, and cellular automata attractors for confusion and diffusion processes. The method uses a key image generated using a ring oscillator circuit in the cascade to facilitate pixel diffusion and secure image transfer server–client architectures. Chai et al. [37] introduced an image encryption method that combines the memristive chaotic system, CS, and ECA techniques. Initially, the plaintext image undergoes a DWT to acquire the sparse coefficient matrix. Subsequently, the sparse coefficient matrix is subjected to a zig-zag scrambling technique and the ECA algorithm. Finally, the scrambled image is compressed using a measurement matrix generated by the memristive chaotic system. Sharma and Kaur [38] reported an improved and hybrid cryptographic approach that relies on altering the mixing matrix within the independent component analysis (ICA) framework and incorporating the chaotic ACM method by using reversible cellular automata. In [39], Hanis and Amutha presented an approach that compresses and encrypts via a key generation algorithm using modified convolution and a chaotic logistic mapping method. The proposal performs a double-image encryption scheme by truncating and combining the four least significant bits. In addition, the resulting image is diffused by cellular automata to increase security. Li et al. [40] presented a color image hybrid encryption algorithm that uses cellular automata and a hyperchaotic system. The pixel values of each color component were summed, and the resulting sum was used, along with the secret keys, to generate the initial value for the logistic map used in encryption. Bhardwaj and Sharma [41] presented encryption methods for images using 2D CA; they used single and double layers to scramble the pixels. Moreover, the authors conducted a performance analysis on both single-layer and double-layer 2D cellular automata. In [42], Rajagopalan et al. reported a combination of software and hardware solutions for image encryption. The proposed method uses an optic system and a dual combination of chaotic cellular automata. On the one hand, the processes of confusion and diffusion of a grayscale image are performed by a logistic map and optocoupler. On the other hand, the encryption method is realized using a multifunctional data acquisition system (DAQ) to interface with the optocoupler for random sequence generation. Liang et al. [43] showed an image encryption algorithm based on a two-dimensional, two-state, and five-neighbor reversible CA. In [44], Chai et al. reported an image encryption approach that utilizes the memristive hyperchaotic system, CA, and DNA sequences. The SHA-256 hash algorithm was utilized to generate the confidential key and the starting values for the chaotic system. Two DNA rule matrices were used in the dynamic DNA encoding and 2D CA to encrypt the plaintext image. Yaghouti et al. [45] presented an image encryption scheme based on a non-uniform cellular automata framework. First, a chaos mapping approach performs the confusion step over the image pixels. Then, a non-uniform cellular automaton creates the key image, and random numbers from this image are selected for encryption using hyperchaotic mapping. In [46], Burak introduced an image encryption method that utilizes a parallelized implementation of the Game of Life and a chaotic system, leveraging the OpenMP standard.

Chaotic maps are used in chaos-based encryption algorithms as the main source of randomness in pseudo-random number generators. In recent years, multi-parametric maps have emerged as alternatives to chaotic maps using a single parameter, which can be vulnerable to attacks through the phase space reconstruction technique. However, these methods require research to define the areas of chaotic behavior, making it hard

to determine the areas of chaotic behavior, i.e., the possible encryption keys. Moreover, there is a degradation problem of chaotic dynamics when finite precision hardware is used due to the rounding of results from arithmetic operations. Therefore, methods utilizing chaotic maps with adaptive symmetry are proposed to develop encryption approaches based on chaos theory. These methods offer wide parameter spaces, and the bifurcation properties of the maps remain unchanged when rotating, compressing, or stretching the phase spaces of the maps [47,48]. Thus, Tutueva et al. [47] proposed an adaptive Zaslavsky web map through multi-parametric bifurcation analysis as a pseudo-random generator. In [48], the authors proved that we could overcome some disadvantages of methods that employ chaos-based cryptography using discrete maps with adaptive symmetry. Moreover, Daoui et al. [49] proposed the multiparametric 1D tent map, which is an extension of a chaotic tent map and consists of six control variables with a domain over an unlimited range and generates a secure key space.

In [50], Nepomuceno et al. proposed the application of the concept of pseudo-orbit to generate a random sequence. Instead of using chaotic systems directly, the authors used the error that appeared due to the computer's finite precision. This error was estimated as the difference between two pseudo-orbits. Furthermore, there was swift progress in the field of discrete fractional calculus, with numerous novel applications being researched. An example is the fractional-order logistic map, which has been shown to have unique bifurcation scenarios and chaotic dynamics in comparison to the whole-order system [51]. In addition, elliptic curve cryptography is a recent, popular, and effective technique for public key cryptography; it reduces the length of safe secret keys required for top-secret documents [52]. Thus, in [51], Askar et al. proposed a cryptosystem by combining the advantages of the elliptic curve techniques and the complicated dynamics of the fractional-order map, by generating an elliptic curve key exchange scheme. Al-Khedhairi and Elsonbaty [52] proposed a fractional-order two-dimensional map and a secure encryption scheme of color images. This scheme combines the associated chaotic pseudo-orbits with the advantages of elliptic curves in public key cryptography.

Other works that use finite-precision error include [53], where Nardo et al. used it as a source of randomness; they obtained the error by using two distinct interval extensions to implement a chaotic system. The resulting sequence has met the criteria for being considered a quality source of randomness by passing all NIST tests, which consist of various random number generators and a set of practical tests designed to evaluate the randomness of binary sequences. Moreover, in [54], Zhou et al. used finite precision by selecting a chaotic system and obtaining the evolution error of two different trajectories of the system to obtain a new chaotic signal that can be used for image encryption.

In this paper, we propose a color image encryption algorithm based on a chaotic model. It uses a hybrid approach based on the modular discrete derivative (MDD), cyclic permutation (CP), Langton's ant (LA), and deterministic noise (DN), in order to achieve an encrypted image with high-level security. The modular discrete derivative is a novel technique used to increase the security of the encrypted image. It is based on a variant of the discrete derivative used in many fields of science and engineering. Due to its characteristics, MDD has the advantage of producing a significant visual impact on the encrypted image. In addition, we used a variant of the deterministic noise and an improvement of Langton's ant, both previously reported in [1]. The deterministic noise and the modular discrete derivative applied to an image hide its visual information, while Langton's ant has the advantage of having a large key space. We conducted multiple tests to examine the encrypted images produced by our method and analyzed their level of security and visual encryption. These tests included statistical analyses, correlation evaluation, entropy computing, entropy quality, texture analysis, the key space universe computing, testing against differential attacks, and an analysis of the key sensitivity.

The use of Langton's ant in image encryption has been explored in only a few papers; despite its potential as a competent method, it remains an underutilized method in this context. This research aims to contribute to the existing literature by further exploring its

strengths and weaknesses in image encryption. Moreover, this research introduces the use of MDD for image encryption, which, to the best of our knowledge, has not been previously studied. By testing the effectiveness of this method, our research seeks to demonstrate its potential as a valuable addition to the existing set of image encryption techniques.

The remaining sections of this paper are structured as follows: Section 2.1 introduces the image dataset used in this work. In Section 2.3, we introduce the concept of the modular discrete derivative operation. Section 2.4 describes the algorithm of the image spatial cyclic permutation. The automaton known as Langton's ant is introduced in Section 2.5; in Section 2.6, we present a deterministic noise algorithm. Later, in Sections 2.7 and 2.8, we make use of the previous methods to propose image encryption and decryption algorithms. We show the experimental results of the proposal in Section 3. A comparison with other state-of-the-art work is given in Section 4. We discuss the results of our research in Section 5. Finally, in Section 6, we conclude our paper and propose work to be developed in subsequent research.

2. Materials and Methods

2.1. Dataset Description

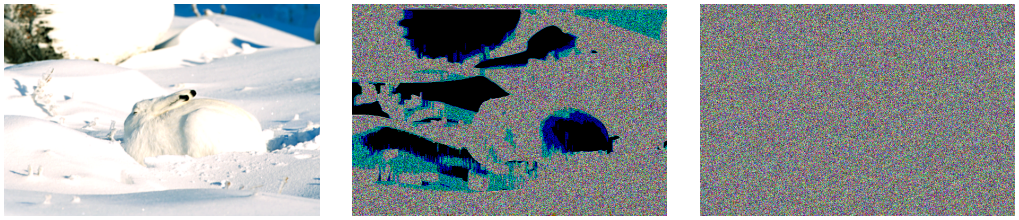
We obtained the images used in this paper from two public image datasets: the USC-SIPI dataset [55] and the University of Konstanz's dataset [56]. They are composed of four RGB images (Baboon.png, Lena.png, Peppers.png, and 4.1.03.png), and four gray-scale images (Barbara.png, Boat.png, Cameraman.png, and Zelda.png); all of them have dimensions of 512×512 pixels. The images selected are the most used as test images in cryptography methods; they are shown in Figure 1. Since our proposed algorithm uses RGB images, we had to convert our gray-scale pictures into RGB images. We accomplished this by replicating the matrix values of the original image and assigning them to each of the three color channels.

(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 1. Image dataset. (**a**) Baboon.png. (**b**) Lena.png. (**c**) Peppers.png. (**d**) 4.1.03.png. (**e**) Barbara.png. (**f**) Boat.png. (**g**) Cameraman.png. (**h**) Zelda.png.

In addition, we used the RGB image arctichare.png with dimensions of 594×400 pixels (Figure 2a) [56]. We selected this image to test the proposed cryptography method on images with large homogeneous areas.



(**a**) (**b**) (**c**)

Figure 2. (**a**) Original image. (**b**) Tenth derivative. (**c**) Tenth derivative of the image with Gaussian noise previously applied.

2.2. Overview of the Proposed Encryption System

The proposed encryption algorithm is composed of five steps, as shown in Figure 3, and as listed below:

1. First deterministic noise;
2. Cyclic permutation;
3. Second deterministic noise;
4. Modular discrete derivative;
5. Langton's ant.

We detail each method and the images used in the following sections.

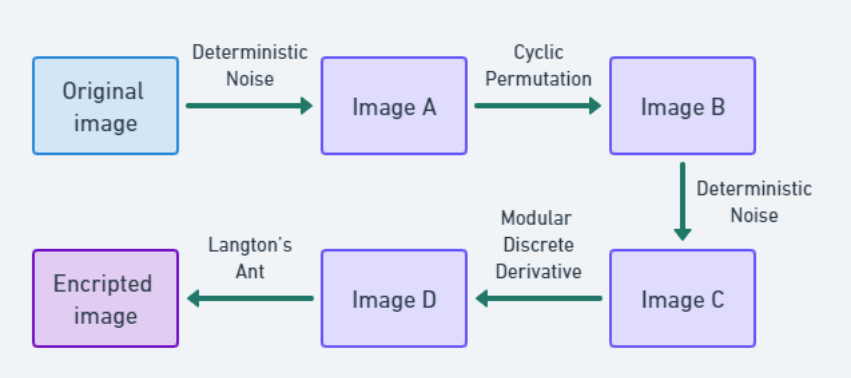


Figure 3. The five steps of the encryption algorithm.

2.3. Modular Discrete Derivative

The discrete derivative is a term used to name an analog of the derivative of a continuous function $f(x)$ with respect to the variable x, but in a discrete domain. In other words, the discrete derivative follows the same definition as the following derivative:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h},$$

except that in our discrete domain, instead of taking the limit as h approaches 0, we consider h as 1. Therefore, the discrete derivative will be given by

$$f'(x) = f(x+1) - f(x).$$

However, for the intent of our algorithm, we will need to obtain the results in a specific range; to achieve this, we will take the modulo n of the result of the discrete derivative. Then, we can define the modular discrete derivative as shown in Equation (1):

$$f'(x) = mod(f(x+1) - f(x), n). \quad (1)$$

In particular, for a range between 0 and 255, for 8-bit images, we will use the modulo $n = 256$, as shown in Equation (2):

$$f'(x) = mod(f(x+1) - f(x), 256) \tag{2}$$

to obtain numbers in the interval that we want.

The modular discrete derivative can be used to describe the following process: let K_n be the first r positive integers, and let $A_n : K_n \to \mathbb{Z}$, such that $A_n(k)$ is the kth element of a sequence. Let K_{n+1} be the first $r-1$ integers numbers, we will then use Equation (2) to define $A_{n+1} : K_{n+1} \to \mathbb{Z}$, such that $A_{n+1}(k) = A'_n(k)$. In other words, if we have a sequence of numbers determined by A_n, A_{n+1} will determine its modular discrete derivative, using modulo 256. To simplify, from now on, we will simply say that A_n is the nth derivative of A_0. It is trivial that if the cardinality of the domain of A_0 is r, A_0 will have, at most, $r-1$ degrees of derivatives. This whole procedure is illustrated in Figure 4. It should be noticed that if we remove the gray numbers in Figure 4, the dark numbers can be used to recover the lost numbers by doing the inverse procedure: $A_n(k) = mod(A_n(k-1) + A_{n+1}(k-1), 256) \mid k \geq 2$.

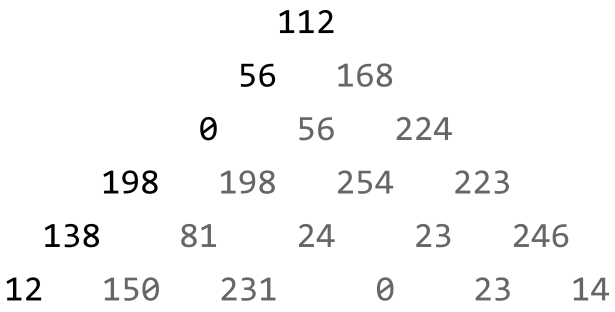


Figure 4. The lowest row is our initial sequence ($A_0(k)$ is the kth element of the sequence). The following rows are the derivatives of the rows below them.

To apply the modular discrete derivative to an image, we take each color channel of the image individually; for each one, we will change their dimensions from a $M \times N$ matrix to a $1 \times MN$ matrix, which we can then treat as our initial sequence. At first, our idea was to take all of the derivatives of the sequence to keep only the dark numbers shown in Figure 4, but given the number of elements of our sequence, this would have taken a lot of time. We noticed that it was possible to compute only a fraction of the triangle, as illustrated in Figure 5, where only the first and second derivatives were calculated.

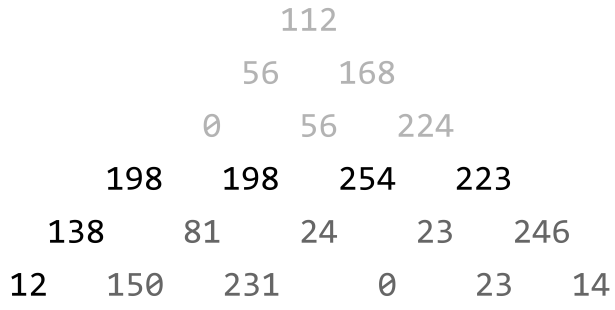


Figure 5. Instead of calculating all of the derivatives similar to before, in this case, we stop on the second derivative.

The dark numbers of Figure 5 can be used to recover the initial sequence at the bottom. Moreover, the amount of dark numbers is equal to the numbers in the initial sequence. Therefore, our next step with the image would be to take the nth derivative of our sequence, keep only the dark numbers, and transform them into an $M \times N$ matrix. After this is done

to all of the channels of the image, we can join the results together to obtain an encrypted image, which can then be decrypted by performing the inverse procedure mentioned before to recover the missing information.

Figure 6 shows the results of taking the tenth derivative of the Lena image. We can see that the MDD of the tenth order has a heavy visual impact on the encrypted image.

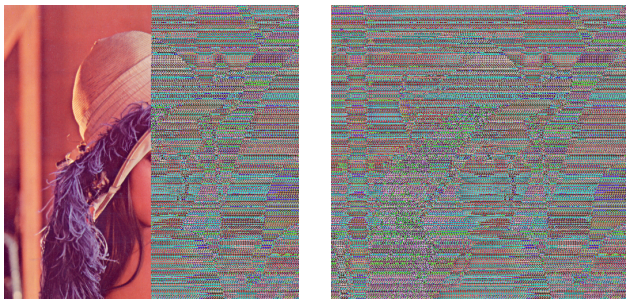
(**a**) (**b**)

Figure 6. (**a**) RGB image of Lena. (**b**) Tenth derivative.

It is evident that the resulting image appears as a random noise pattern. However, this is not always the case. If an image has any areas with the exact same colors, the derivative would be zero in them (and as a consequence, in the next derivatives as well). For example, in Figure 2, we present an image with large homogeneous areas that are then transformed into large black areas that reveal some of the shapes of the original image. To solve this, we added Gaussian noise with a mean of 0 and variance of 0.0001 to the original image, this noise is imperceptible to the human eye. When we apply the MDD, we obtain an encrypted image that again looks similar to random noise and shows none of the original shapes.

While this solution is simple, it comes at the cost of sacrificing the original image: random noise would make it impossible to obtain the exact original image, even if the resulting decrypted image looks identical to the original (to the human eye). Hence, in Section 2.6, we introduce a deterministic noise to replace the Gaussian noise Section 2.6.

An important aspect of MDD to highlight is its sensitivity to changes. If an image is encrypted using MDD, and one pixel of the resulting image is altered, when conducting the inverse algorithm, all pixels in the corresponding column below that pixel and all columns to the right will be affected. Figure 7 shows two examples of this sensitivity.



(**a**) (**b**)

Figure 7. Results of encrypting Lena using MDD, modifying a pixel of the encryption, and decrypting with the inverse MDD. (**a**) Pixel altered in column 256, row 100. (**b**) Pixel altered on column 1, row 100.

2.4. Image Spatial Cyclic Permutation

In mathematics, a permutation $P(S)$ consists of the reordering of the members of a set S into another sequence or linear order. For example, let $S = \{s_0, s_1, \ldots, s_{L-1}\}$ be a finite set composed of L elements, in Cauchy's two-line notation [57], a particular permutation P is shown as follows:

$$P(S) = \begin{pmatrix} s_0 & s_1 & s_2 & \cdots & s_{L-1} & s_L \\ s_n & s_{n-1} & s_2 & \cdots & s_1 & s_0 \end{pmatrix}, \tag{3}$$

where the list of elements of S is shown in row 1 and the corresponding image in row 2.

On the other hand, a cyclic permutation $CP(S)\{k\}$ is a permutation that shifts the elements of S by an offset, $k \in \mathbb{Z}$ and $k \neq 0$, generating a set where the elements, either the beginning or the end, are inserted in the opposite direction while retaining the relative position between them. This procedure can be expressed mathematically as $s_i \rightarrow s_{i+k \pmod L}$ [58]. Thus, when $k \geq 1$ we obtain a right-cyclic permutation, and when $k \leq -1$, we obtain a left-cyclic permutation.

Therefore, the following permutation is a left-cyclic permutation with $k = -1$:

$$CP(S)\{k = -1\} = \begin{pmatrix} s_0 & s_1 & s_2 & \cdots & s_{L-1} & s_L \\ s_1 & s_2 & s_{L-1} & \cdots & s_L & s_0 \end{pmatrix}. \tag{4}$$

In a two-dimensional case, for example, a digital image $A(x, y)$ of $M \times N$ pixels, a cyclic permutation $CP(A)\{k_x, k_y\}$ could be applied to its spatial coordinates x and y, as shown in Equation (5):

$$CP(A)\{k_x, k_y\} = A\Big(x + k_x \pmod N, y + k_y \pmod M\Big), \tag{5}$$

where k_x and k_y are the horizontal and vertical offsets, respectively.

When the cyclic permutation is applied over a multi-spectral digital image, e.g., an RGB image, the process is performed on each channel to keep the image color unchanged.

To show the cyclic permutation process, in Figure 8, we can see some results. Thus, Figure 8a shows a $512 \times 512 \times 3$ image, Figure 8b,c show the results for the horizontal left-cyclic permutation $CP(A)\{k_x = -200, k_y = 0\}$ and the cyclic permutation of elements in an upward vertical direction $CP(A)\{k_x = 0, k_y = 200\}$, respectively. Finally, Figure 8d shows the result of a permutation that cyclically shifts elements in a vertical upward direction, followed by another permutation that cyclically shifts elements in a horizontal left direction over the image $CP(A)\{k_x = -300, k_y = 300\}$.

Figure 8. A digital RGB image being modified by different cyclic permutations. (**a**) RGB image. (**b**) Horizontal left-cyclic permutation result $CP(A)\{k_x = -200, k_y = 0\}$. (**c**) Cyclic permutation of elements in an upward vertical direction result $CP(A)\{k_x = 0, k_y = 200\}$. (**d**) Vertical upward-cyclic permutation in succession of a horizontal left-cyclic permutation result $CP(A)\{k_x = -300, k_y = 300\}$.

2.5. Langton's Ant

Langton's ant is a cellular automaton that was introduced by Christopher Langton in 1986 [59]. It has garnered significant interest across multiple fields, including emergent dynamics, Lorentz lattice gas, computational complexity, and cryptography. Its study stems from the challenge of predicting the macroscopic behavior of the ant based on its initial microscopic configuration [60].

Langton's ant can be described as a two-dimensional universal Turing machine with two notable characteristics: (i) A simple set of rules and (ii) a complex emergent behavior. The ant navigates an infinite grid consisting of cells that can be in an OFF or ON state. Initially, the ant is positioned on the grid and oriented in one of four directions: up, right, down, or left. As it navigates the grid, the ant follows two rules: (i) if the current cell is OFF, it turns 90 degrees clockwise, switches the cell state to ON, and moves to the next cell in its path; (ii) conversely, if the current cell is ON, it turns 90 degrees counterclockwise, switches the cell state to OFF, and moves forward to the next cell. Figure 9 shows the first four iterations of the automaton.

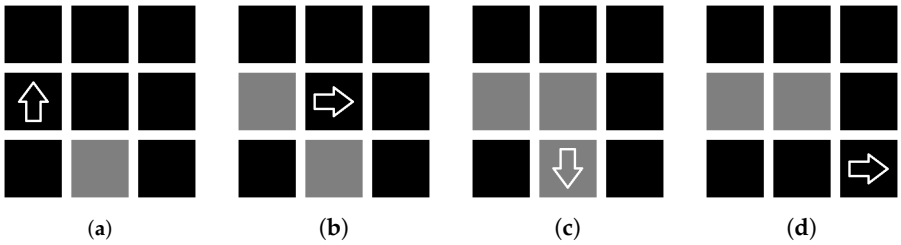


Figure 9. Langton's ant taking the first four steps on a 3×3 grid with only one cell turned ON initially. (**a**) First step. (**b**) Second step. (**c**) Third step. (**d**) Fourth step.

In the first few steps, the ant appears to have chaotic behavior; however, if placed on a grid composed entirely of turned-OFF cells, the ant eventually reaches stable behavior in the 104 steps known as "the highway", as shown in Figure 10, giving rise to the highway conjecture: "If the ant is placed on an infinite grid with either a finite amount of cells turned ON or a finite amount of cells turned OFF, the highway structure must form eventually" [60].

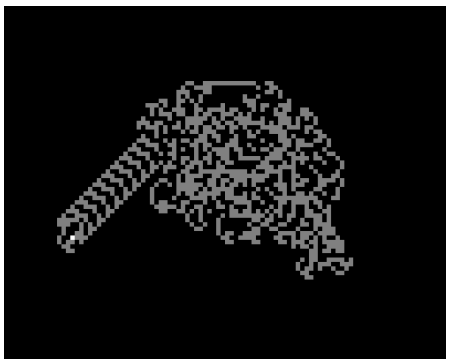


Figure 10. Langton's ant (white cell) stuck in "The Highway" after 11,538 steps.

Evidently, these rules can only be applied to a grid of infinite extensions since the behavior of the ant on an edge is not defined, as would be the case in a grid composed of a digital image.

In the research conducted by Wang and Xu [61], the original rules of Langton's ant were employed. The authors generated the values of the grid by using an intertwining logistic map and made adjustments to coordinates and rotation directions due to the finite grid of a digital image.

In this research, and as reported by Romero-Arellano et al. [1], when the ant crosses an edge of the grid, it reappears on the opposite edge, creating a topological equivalence to the ant moving on the surface of a torus. As a result, if "the highway" pattern emerges, it will eventually be disrupted, and the ant's motion will return to a state of chaos.

The reversibility of Langton's ant is trivial. The process can be reversed by rotating the ant 180 degrees and allowing it to take the same number of steps as it did initially, the ant retraces its path and returns to its original configuration.

Unlike Wang and Xu [61], where Langton's ant was applied to digital images using the original set of rules, and as presented in work by Romero-Arellano et al. [1], we focused on the gray level of the image. For an RGB image (with pixel values ranging from 0 to 255 in each color channel), we separated the color channels of the image and applied the ant to each channel individually. Subsequently, we modified the rules of the ant to adapt them to the 256 values of a gray-scale image: (i) if the ant was in an even pixel, it considered it as an OFF cell, (ii) if the ant was in a pixel with an odd value, it considered it as an ON cell. To switch the current state of the cell, the ant added an arbitrary odd integer, for example, 47, to the pixel value, changing its parity. If the result was greater than or equal to 256, we took its modulo as 256. After the ant was applied to all color channels, we combined the results to obtain an encrypted image. In order to decrypt the image, the image was first separated into its color channels. Then, the ant was placed at the final position it reached during the encryption process. The ant was rotated 180 degrees and moved the same number of steps it initially took during encryption, but instead of adding the odd integer (47) at each iteration, we subtracted 47 and took modulo 256 if necessary.

In Figure 11, we show the results of applying this algorithm on the 512×512 Lena RGB image with a variety of iterations. The results show that the main disadvantage of Langton's ant is that a very large number of steps is needed to completely hide the visual information of the image, and this amount increases with the size of the image.

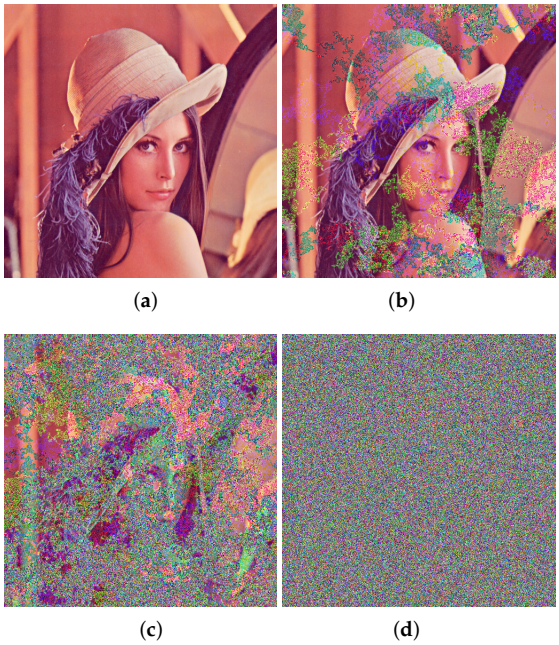


Figure 11. A 512×512 RGB image modified by Langton's ant with different amounts of iterations. (**a**) Original RGB image. (**b**) Using 100,000 iterations. (**c**) Using 300,000 iterations. (**d**) Using 1,500,000 iterations.

In our proposal, we took the color channels of an RGB image, and then used Langton's ant on each one by giving it an initial coordinate, an upward orientation, and a number of steps to walk. Then, we saved the final coordinates, final orientation, and the number of steps as our decryption key.

2.6. Deterministic Noise

In order to utilize the modular discrete derivative, we used a function that outputted pseudo-random integers. These numbers were added to the image as noise. This method is a slight modification of the deterministic noise method described by Romero-Arellano et al. [1]. The working principle of the deterministic noise is as follows: Consider an RGB image A with dimensions $M \times N \times 3$. We define three variables, s_1, s_2, and s_3, according to Equation (6):

$$s_\tau = \text{mod}\left(\sum_{i=1}^{N}\sum_{j=1}^{M} A(i,j,\tau)(jN - N + i), M \times N\right), \tag{6}$$

where i, j represent the row and column of each pixel, respectively, and $\tau = 1, 2, 3$ represents the color channel for RGB images.

Then, we define variables p_1, p_2, and p_3 as shown in Equations (7)–(9).

$$p_1 = s_2 * s_3 + s_1, \tag{7}$$

$$p_2 = s_1 * s_3 + s_2, \text{ and} \tag{8}$$

$$p_3 = s_2 * s_1 + s_3. \tag{9}$$

This ensures that applying the algorithm on two images that are almost identical, except for the value of a bit, will give different outputs for s_1, s_2, and s_3 and, consequently, different values for p_1, p_2, and p_3; therefore, a very different noise will be applied to each image.

If $p_1 = p_2$, then we increment p_2 by one unit. If $p_1 = p_3$, then we increment p_3 by one unit. Finally, if $p_2 = p_3$, then we increment p_3 by one unit. In this way, we will make sure that all of the values are different.

We now define a $M \times N$ RGB image B, which will represent the result of adding noise to A. For each row i, we multiply p_1, p_2, and p_3 by i to obtain the variables z_1, z_2, and z_3, respectively. Subsequently, we can calculate $B(i,j,1)$, $B(i,j,2)$, and $B(i,j,3)$ for the jth element of the current row of A by using Equations (10)–(12):

$$B(i,j,1) = \text{mod}\left(A(i,j,1) + \left\lfloor \frac{z_1 * i + j}{z_2 + z_3} \right\rfloor, 256\right), \tag{10}$$

$$B(i,j,2) = \text{mod}\left(A(i,j,2) + \left\lfloor \frac{z_2 * i + j}{z_1 + z_3} \right\rfloor, 256\right), \text{ and} \tag{11}$$

$$B(i,j,3) = \text{mod}\left(A(i,j,3) + \left\lfloor \frac{z_3 * i + j}{z_2 + z_1} \right\rfloor, 256\right). \tag{12}$$

Our use of modulo 256 is due to each color channel of the image being an 8-bit image; therefore, we need the output to be in the range of 0 to 255. Before moving to the next column, we will modify z_1, z_2, and z_3 by first calculating some auxiliary variables, q_1, q_2, and q_3, which are defined in Equations (13)–(15):

$$q_1 = \text{mod}\left(\left\lfloor \frac{z_1 * i + j}{z_2 + z_3 + 1} \right\rfloor, 256\right), \tag{13}$$

$$q_2 = \text{mod}\left(\left\lfloor \frac{z_2 * i + j}{z_1 + z_3 + 1} \right\rfloor, 256\right), \text{ and} \tag{14}$$

$$q_3 = \text{mod}\left(\left\lfloor \frac{z_3 * i + j}{z_2 + z_1 + 1} \right\rfloor, 256\right). \tag{15}$$

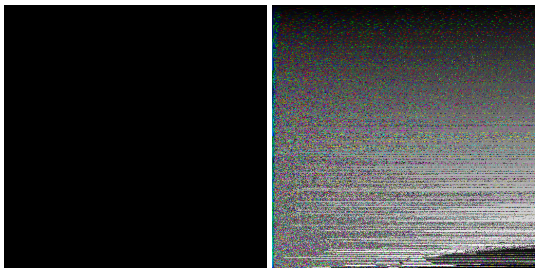
Then we use Equation (16) to recalculate z_1, z_2, and z_3:

$$z_\tau = \text{mod}(q_\tau + z'_\tau, 256) + 1, \tag{16}$$

where $\tau = 1, 2, 3$ and z'_τ corresponds to the last value of z_τ.

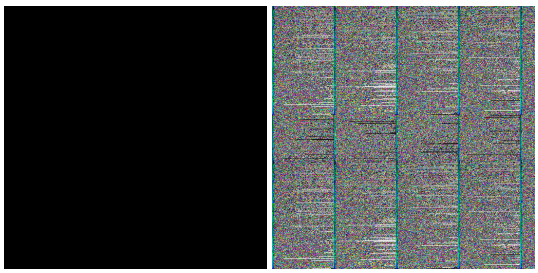
We move to the next column and use the values of z_1, z_2, and z_3 to apply Equations (10)–(12), recalculate the values using Equations (13)–(16), move on to the next column, and repeat until we finish the current row and move on to the row below. Each time we change rows, we use p_1, p_2, and p_3 again to calculate z_1, z_2, and z_3, continuing with the algorithm until we modify every pixel.

We show the results of using our deterministic noise on a completely black image of dimensions $512 \times 512 \times 3$ in Figure 12. The results show that there are areas that are almost unaffected by the noise; the best areas are those after row 300 and before column 120. To fix this issue, we altered the deterministic noise algorithm. If $i < 300$, we increase i by 300, and for every 120 columns, we recalculate the values of z_1, z_2, and z_3 as if we changed to a new row. The resulting noise generated by this modified method can be seen in Figure 13. As a consequence of this modification, a vertical pattern can be seen on the image every 120 columns.



(**a**) (**b**)

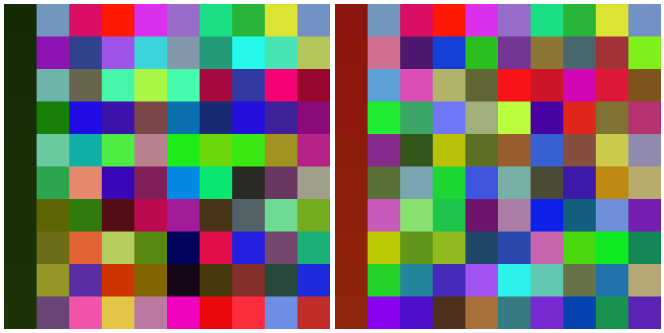
Figure 12. Applying the original deterministic noise. (**a**) Black image. (**b**) Black image with the noise applied.



(**a**) (**b**)

Figure 13. Applying the modified deterministic noise. (**a**) Black image. (**b**) Black image with the modified noise applied.

It is worth noting that when we are in any given column and recalculate the z_1, z_2, and z_3 values for the first time, we obtain three numbers in a range $[1, 256]$, which completely determine the rest of the values in the row. To be more detailed, after we obtain the first value of the row, there exists 256^3 possible sequences of values for the rest of the row. It is also worth noting that if one set of parameters p_1, p_2, and p_3 generates parameters z_1, z_2, and z_3 for row i, and a different triad of variables p_1', p_2', and p_3' gives as output for the same row i an identical set of parameters, i.e., z_1, z_2, and z_3, then the variables that will be obtained for the following row are not necessarily also equal. In simpler terms, the results of a row do not predetermine the results of the following rows. A counterexample is shown in Figure 14, where we applied the deterministic noise to a black RGB image, one time with the parameters p_1, p_2, and p_3, and another time with another set of parameters, obtaining identical values for the first row (except for the first pixel) and with different values for the following rows.



(**a**) (**b**)

Figure 14. Black 10×10 RGB image with deterministic noise applied, using two different sets of parameters. (**a**) Using $p_1 = 400$, $p_2 = 417$, and $p_3 = 20$. (**b**) Using $p_1 = 403$, $p_2 = 810$, and $p_3 = 54$.

Removing the noise is trivial. If the noise was generated using the parameters p_1, p_2, and p_3, then we used those parameters to perform the exact same algorithm but replaced Equations (10)–(12) with Equations (17)–(19), respectively:

$$A(i,j,1) = \operatorname{mod}\left(B(i,j,1) - \left\lfloor \frac{z_1 * i + j}{z_2 + z_3} \right\rfloor, 256\right), \tag{17}$$

$$A(i,j,2) = \operatorname{mod}\left(B(i,j,2) - \left\lfloor \frac{z_2 * i + j}{z_1 + z_3} \right\rfloor, 256\right), \text{ and} \tag{18}$$

$$A(i,j,3) = \operatorname{mod}\left(B(i,j,3) - \left\lfloor \frac{z_3 * i + j}{z_2 + z_1} \right\rfloor, 256\right). \tag{19}$$

2.7. Encryption Algorithm

The proposed encryption algorithm consists of five distinct steps, each contributing to the overall encryption process. The steps shown in Figure 3 outline the sequential operations performed to transform the input data into an encrypted form, ensuring the confidentiality and integrity of the information. The initial step of our proposal involves applying the deterministic noise to the image, which will mitigate the issue with homogeneous zones mentioned in Section 2.3. Due to the vertical and horizontal patterns of the deterministic noise, the second step is to shift the image 60 columns to the right and 60 rows down with the cyclic permutation, followed by applying the deterministic noise again as the third step. Now that the image is free of homogeneous zones, the next step is to apply the modular discrete derivative, which drastically changes the histogram of the image and makes it look similar to random noise. Finally, Langton's ant is applied in the fifth step, which provides security to the algorithm by providing a large key space (detailed in Section 3.6), takes advantage of the sensibility of MDD to the initial conditions in the decryption process, and does not require a large number of steps since the image has already been modified by the MDD to look similar to random noise. We position the ant on the first column, this ensures that if, during the decryption process, the ant gives one more or one less step than necessary, a pixel on the first column will have the wrong value, which will generate wrong values for all columns to the left when the inverse MDD is applied.

To summarize, LA provides security, MDD hides the image, DN prepares the image for MDD, and CP fixes any gaps left over by the DN.

Our proposal uses the following variables as parameters to achieve the image encryption: the degree of the derivative of MDD for each color channel, the row representing the initial position for each of the three ants, their orientations, and the number of steps they take.

2.8. Decryption Algorithm

In order to decrypt the resulting images, we use the inverse function of each of the methods involved in the algorithm, as represented in Figure 15. To begin, we take the

final coordinates and orientations of the ant on each color channel and let them walk the same number of steps as the ones used to encrypt. Then we use the three degrees of the derivatives used in step four. Moreover, we use the values of p_1, p_2, and p_3 for step three. We shift the image to 60 columns to the left and 60 rows up to revert to step two. Finally, we use the values of p_1, p_2, and p_3, corresponding to the inverse deterministic noise from step one.

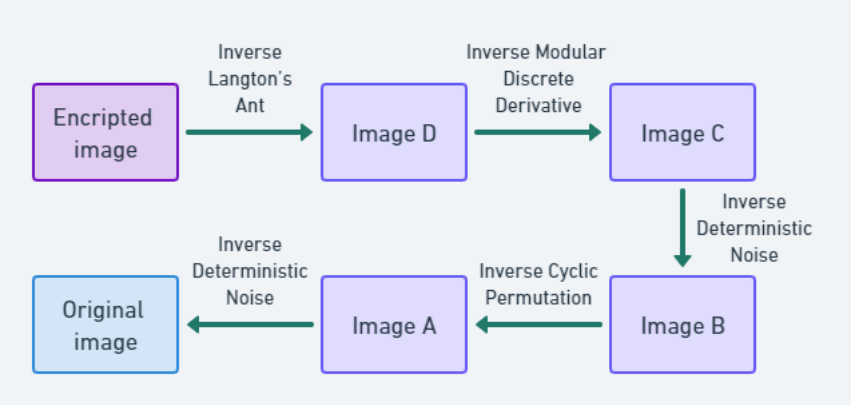


Figure 15. Decryption algorithm, divided into five steps.

3. Results and Security Analysis

The results obtained for the proposed encryption algorithm are presented in this section. The results are divided into eight parts: Section 3.1 shows all of the images obtained at each step of the encryption algorithm when encrypting Lena whilst Section 3.2 introduces a statistical analysis that compares the original image with its encrypted counterpart. Additionally, it compares the histograms and presents the correlation coefficient. Section 3.3 analyzes the entropy of the resulting encrypted image. Section 3.4 shows the metrics obtained to measure the encryption quality. Section 3.5 presents the results of the texture analysis. The key space universe of the proposed system is defined in Section 3.6. Section 3.7 studies the strength of the algorithm against differential attacks. Section 3.8 presents an analysis of the sensitivity of the key. Finally, Section 3.9 presents the computational complexity of the algorithm.

The encryption and decryption results of the proposed scheme were obtained using a PC equipped with an AMD Ryzen 5 3500U processor, with 12 GB of RAM, running at a frequency of 2.1 GHz. Both encryption and decryption algorithms were implemented in MATLAB. The encryption algorithm takes a time-consuming 4.3602 s, while the decryption algorithm takes around 4.6642 s for RGB images with dimensions of 512×512 pixels. These times are obtained when using the fifth derivative of MDD and taking 5000, 5000, and 3000 steps with Langton's ant in the bands corresponding to the colors red, green, and blue.

3.1. Encryption Results

The results of our encryption algorithm are shown in this section. We use the 512×512 images described in Section 2.1. Since the algorithm works with images with three channels, the grayscale images were transformed into RGB images. The parameters used are as follows: For the fourth step (MDD), we use the fifth derivative, for the fifth step (LA), the red channel ant is positioned at row 96 and takes 3000 steps, the green channel ant is positioned at row 120 and takes 5000 steps, and the blue channel ant is positioned at row 175 and takes 5000 steps.

Figure 16 shows the images obtained at each step of the encryption process using a 512×512 RGB Lena image as input. For the rest of the dataset, similar results were obtained. In the following sections, we will report the quantitative analyses for the encrypted images using several image processing metrics, such as statistical metrics, entropy analysis, encryption quality, and texture analysis. For RGB images, the metrics correspond to the average value between the three metrics for each color channel.

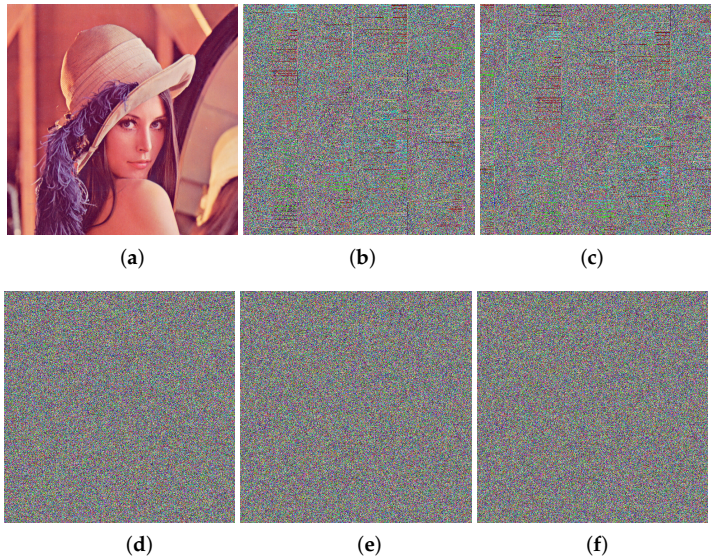


Figure 16. Images obtained at each step of the encryption scheme over the Lena image. (**a**) Original image. (**b**) Step 1: First deterministic noise. (**c**) Step 2: Cyclic permutation. (**d**) Step 3: Second deterministic noise. (**e**) Step 4: Modular discrete derivative. (**f**) Step 5: Langton's ant.

3.2. Statistical Analysis

In this section, we present a statistical analysis to evaluate the proposed scheme. Firstly, histograms of each channel are showcased for both the original and encrypted images. Secondly, the correlation of neighboring pixels is analyzed to assess the degree of local pixel dependence obtained in the results.

3.2.1. Histogram Analysis

In Figure 17, we show the histogram for each of the color channels of Lena, before and after being encrypted.

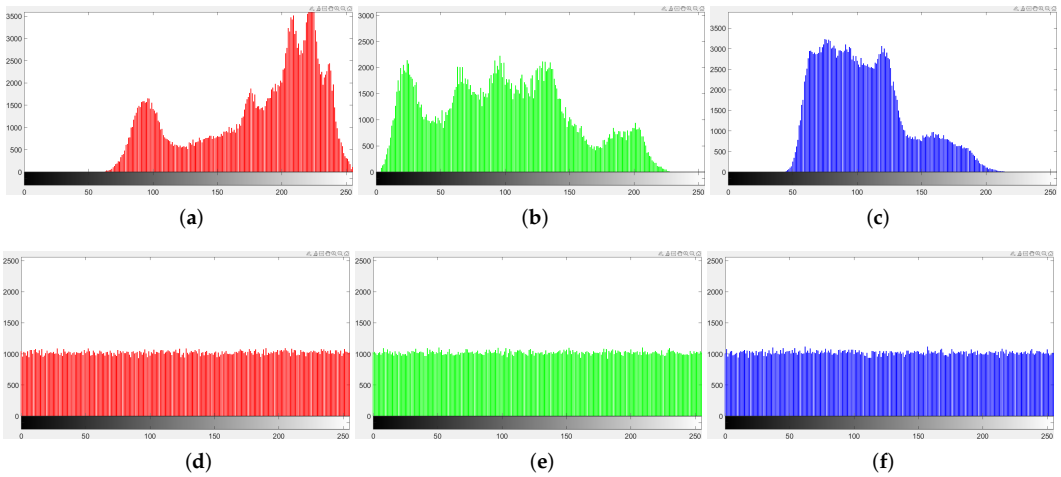


Figure 17. Histograms of Lena's channels, before and after encryption (**a**) Channel red, before encryption. (**b**) Channel green, before encryption. (**c**) Channel blue, before encryption. (**d**) Channel red, after encryption. (**e**) Channel green, after encryption. (**f**) Channel blue, after encryption.

Chi-square tests can be implemented to measure the uniformity of the resulting histograms. Pearson's chi-square (χ^2) goodness of fit statistic for categorical data of the histogram of an encrypted image histogram is computed by Equation (20):

$$\chi^2 = \sum_{k=0}^{L-1} \frac{O(k) - E(k)}{E(k)}, \tag{20}$$

where L corresponds to the number of gray levels that are possible, i.e., 256 for an 8-bit image; given a gray-level k, its observed frequency is represented by $O(k)$, while its expected frequency for the exact uniform distribution is represented with $E(k)$, which could be calculated as:

$$E(k) = \frac{M \times N}{L} \forall\, k, \tag{21}$$

with M and N representing the number of rows and columns of the gray-level image.

For gray-scale encrypted images of 8-bits, the chi-square test is passed if:

$$\chi^2 < \chi^2_\alpha(df), \tag{22}$$

where $\chi^2_\alpha(df)$ represents the critical chi-square value, α corresponds to the significance level, and df represents the degree of freedom of the chi-square distribution.

Assigning a significance level $\alpha = 0.05$, a confidence interval containing 95% of values, and fixing the degrees of freedom to $df = L - 1 = 255$, the critical chi-square value is defined as $\chi^2_{0.05}(255) = 293.247$ for intensity images of 8 bits.

Table 1 shows the chi-square scores of the ciphered images. Thus, chi-square values less than 293.247 indicate a histogram of the encrypted image that is very close to a uniform histogram, being robust to the histogram analysis.

Table 1. Chi-square values.

Image	χ^2	Remarks
Baboon	233.951	Passed
Lena	260.244	Passed
Peppers	257.839	Passed
4.1.03	281.687	Passed
Barbara	270.602	Passed
Boat	263.305	Passed
Cameraman	234.850	Passed
Zelda	261.809	Passed

3.2.2. Correlation Analysis

The 2D normalized cross-correlation (NCC), otherwise known as the correlation coefficient, measures the similarity between two images, $A(x, y)$ and $B(x, y)$, as shown in Equation (23):

$$NCC = \frac{\sum_{x,y}\left(A(x,y) - \overline{A}\right)\left(B(x,y) - \overline{B}\right)}{\sqrt{\left(\sum_{x,y}\left(A(x,y) - \overline{A}\right)^2\right)\left(\sum_{x,y}\left(B(x,y) - \overline{B}\right)^2\right)}}, \tag{23}$$

where $\overline{A}$ and $\overline{B}$ are the average intensity values of images A and B, respectively.

Thus, the 2D correlation coefficient is scaled within the range of $[-1, +1]$, where -1 indicates a maximum negative correlation, $+1$ indicates a maximum positive correlation, and a correlation of 0 represents no association between the original image and its encryption.

In Table 2, we show the correlation coefficient obtained between the plaintext image and ciphered image. Since the coefficient applies to images with one channel, we took the average of the coefficient with each color channel.

Table 2. Correlation coefficient values.

Image	*NCC*
Baboon	-5.07×10^{-4}
Lena	-1.92×10^{-5}
Peppers	$+1.79 \times 10^{-3}$
4.1.03	-7.25×10^{-4}
Barbara	$+6.26 \times 10^{-4}$
Boat	$+2.97 \times 10^{-5}$
Cameraman	$+5.45 \times 10^{-4}$
Zelda	$+1.22 \times 10^{-3}$

While Section 3.2.1 shows the global decorrelation level for each color channel of the encrypted image, this section focuses on the local correlation of neighboring pixels given a particular direction, e.g., vertical, horizontal, and diagonal.

In a plaintext image, neighboring pixels exhibit significant correlations in vertical, horizontal, and diagonal directions [62]. Conversely, during image encryption, the objective is to minimize the correlation between the pixels in the ciphered image and the original image.

We calculated the correlation between adjacent pixels by measuring the correlation (Equation (23)) of a set of 1000 randomly chosen pairs of pixels against their corresponding adjacent pixels in a specific direction (i.e., two horizontally, two vertically, and two diagonally adjacent pixels) in the plain and corresponding encrypted images.

In Figure 18, we show the correlation distributions of each of the color channels of Lena, in the vertical direction. Comparable results were achieved when examining the horizontal and diagonal orientations; this shows that the direction in which the derivative is taken does not have any significant impact on the final result.

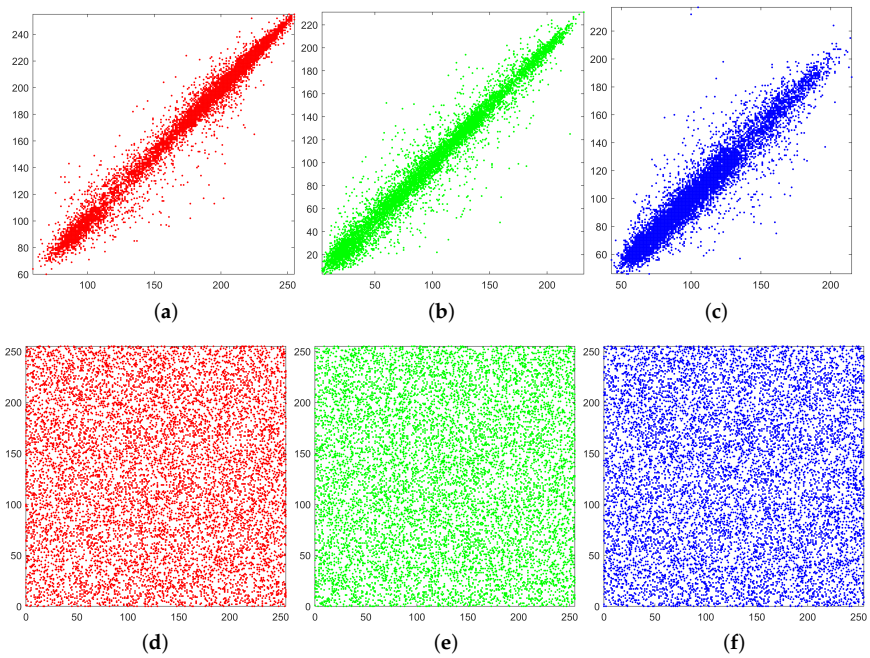


Figure 18. Correlation distributions for each color channel of Lena in the vertical direction. (**a**) Red channel, before encryption. (**b**) Green channel, before encryption. (**c**) Blue channel, before encryption. (**d**) Red channel, after encryption. (**e**) Green channel, after encryption. (**f**) Blue channel, after encryption.

3.3. Entropy Analysis

One common metric used to measure the degree of disorder in a system is its entropy, which can be used in the context of cryptography to measure the randomness of an encrypted message.

Given an information source q, we can define its entropy H_q with Equation (24):

$$H_q = \sum_{k=0}^{R-1} p(q_k) \log_2 \frac{1}{p(q_k)}, \tag{24}$$

where R stands for the total amount of symbols q_k in source q, and the probability of occurrence of each symbol q_k is given by $p(q_k)$. Given an image with 8 bits per channel, it will have a total of 256 possible symbols. If each one has the same probability of occurring, the entropy of the image will be 8.

Table 3 shows the entropy values obtained for the ciphered images of the dataset used. The 4.103 image generated the lowest value, while the Baboon image generated the highest value. On the other hand, the Lena image obtained an entropy value of 7.999283079.

Table 3. Entropy values for the encrypted images.

Image	H_q
Baboon	7.999355791
Lena	7.999283079
Peppers	7.999289444
4.1.03	7.999225223
Barbara	7.999256159
Boat	7.999275579
Cameraman	7.999352948
Zelda	7.999280102

3.4. Encryption Quality

Since a visual inspection of the encrypted image only incorporates a subjective measurement, several literature works have introduced quantitative encryption quality metrics using the deviation in the values of the pixels between the plain image and its encryption [63].

Therefore, the encryption quality is satisfactory when there is a high degree of maximum and irregular pixel deviations or alterations between the plaintext and encrypted image. Thus, in [63], four encryption quality metrics were defined:

- Maximum deviation.
- Irregular deviation.
- Deviation from the uniform histogram.
- Peak signal-to-noise ratio.

3.4.1. Maximum Deviation

Maximum deviation (MD) for 8-bit gray-scale images is obtained by applying Equation (25) [63]:

$$MD = \frac{d(0) + d(255)}{2} + \sum_{k=1}^{254} d(k), \tag{25}$$

where d represents the absolute difference in the original and encrypted histograms $d(k)$ is the difference in the amplitudes of these histograms for the intensity level k. $d(0)$ and $d(255)$ correspond to the histograms at gray-level 0 and 255, respectively.

Higher values of *MD* correspond to encrypted images that are more deviated from the original images.

3.4.2. Irregular Deviation

Irregular deviation (*ID*) is another metric used to measure the encryption quality by assuming that an effective encryption algorithm should uniformly randomize the input pixel values [63]. Its efficacy is assessed by measuring the proximity of the histogram deviation distribution to a uniform distribution.

First, the absolute difference (AD) between the plain image (A) and the encrypted image (B) is calculated:

$$AD = |A - B|. \tag{26}$$

Then, the histogram of AD, referred to h, is obtained, where $h(k)$ corresponds to the histogram value h at index k, and M_h defines the mean value of the histogram as is shown in Equation (27):

$$M_h = \frac{1}{256} \sum_{k=0}^{255} h(k). \tag{27}$$

Finally, the irregular deviation for an 8-bit image is calculated, as shown in Equation (28):

$$ID = \sum_{k=0}^{255} |h(k) - M_h|. \tag{28}$$

For smaller values of ID, we obtain better encryption qualities.

3.4.3. Deviation from the Uniform Histogram

In [63], a new encryption quality factor was proposed, which measures the deviation between the ideal and uniform histogram from the histogram of the ciphered image.

Let h_B be the histogram of the encrypted image B, where $h_B(k)$ is the frequency of occurrence at gray level k; let $E(k)$ be the frequency of gray level k in a uniform histogram, as defined in Equation (21) for $L = 256$. Thus, the deviation from the uniform histogram (DU) is obtained, as shown in Equation (29):

$$DU = \frac{1}{M \times N} \sum_{k=0}^{255} |h_B(k) - E(k)|. \tag{29}$$

A better encryption quality is obtained for lower values of DU. A lower value indicates that the histogram of the encrypted image deviates less from an ideal uniform histogram.

3.4.4. Peak Signal-to-Noise Ratio

The peak signal-to-noise ratio (*PSNR*) is another metric that could be used to evaluate the quality of an encrypted image. It measures the changes in pixel values between two images of 8 bits, the original image A and its encryption B. *PSNR* can be defined mathematically, as given by Equation (30):

$$PSNR = 10 \log_{10} \left(\frac{255^2}{\frac{1}{M \times N} \sum_{x,y} \Big(A(x,y) - CB(x,y) \Big)^2} \right). \tag{30}$$

A better encryption quality is achieved for lower values of *PSNR*.

3.4.5. Mean Square Error

In addition, the mean square error (*MSE*) is a statistical metric that is widely used to evaluate the quality of an encryption algorithm [64]. It calculates the average squared differences between the pixels of the plain image A and the encrypted image B, as shown in Equation (31):

$$MSE = \frac{1}{M \times N} \sum_{x,y} \Big(A(x,y) - B(x,y) \Big)^2. \tag{31}$$

Similar to the *PSNR*, better encryption qualities are achieved for lower values of *MSE*.

In Table 4, we show the results of the encryption quality metrics on the dataset (using the average of the results on each color channel).

Table 4. Encryption quality results of the dataset.

Image	*MD*	*ID*	*DU*	*PSNR*	*MSE*
Baboon	143,366.500	16,4376.666	0.023	8.793	8614.279
Lena	204,618.666	159,758.000	0.024	8.682	8928.916
Peppers	199,269.333	138,798.000	0.025	8.134	10,111.045
4.1.03	338,220.000	209,675.333	0.026	9.957	6567.653
Barbara	131,537.833	165,227.333	0.025	8.810	8550.455
Boat	220,773.666	186,129.333	0.025	9.295	7647.103
Cameraman	255,633.833	158,272.000	0.024	8.412	9371.920
Zelda	211,912.000	172,643.333	0.025	8.882	8410.595

3.5. Texture Analysis

The texture analysis involves a set of techniques used to assess the frequency of dissimilar combinations of gray levels within a particular spatial neighborhood or across the entire image. [62]. The gray-level co-occurrence matrix (*GLCM*) involves a commonly used texture measure of a gray-scale image $A(x,y)$. This considers the spatial relationships between the pixels of the image [65].

The *GLCM* matrix is calculated based on the frequency at which a pixel with value p occurs in correlation with another pixel having an intensity value of q, as shown in Equation (32):

$$G_M(x,y) = C_{\Delta x,\Delta y}(p,q) = \sum_{x=1}^{N} \sum_{y=1}^{M} \begin{cases} 1, & \text{if} \quad A(x,y) = p \quad \text{and} \quad A(x+\Delta x, y+\Delta y) = q \\ 0, & \text{otherwise} \end{cases}, \tag{32}$$

where $C_{\Delta x,\Delta y}(i,j)$ is the frequency with which two pixels with intensities p and q, at a specific separation $(\Delta x, \Delta y)$, occur.

3.5.1. *Homogeneity*

Homogeneity is a texture metric that evaluates the closeness of the distribution of pixels or how close the pixels are to each other [62,66]. Equation (33) shows the relation to calculate *Homogeneity*:

$$Homogeneity = \sum_{x,y} \frac{G_M(x,y)}{1 + |x-y|}, \tag{33}$$

where G_M represents the *GLCM* matrix.

The lower values of *Homogeneity* represent a higher encryption quality.

3.5.2. *Contrast*

Given a pixel and its surrounding neighborhood, the intensity contrast is calculated as shown in Equation (34) [62,66], resulting in the *Contrast* metric. It must be as high as possible:

$$Contrast = \sum_{x,y} |x-y|^2 G_M(x,y). \tag{34}$$

3.5.3. *Energy*

Energy in *GLCM* measures uniformity, the smaller value of *Energy* refers to a higher degree of disorder [62,66]. *Energy* can be calculated as shown in Equation (35):

$$Energy = \sum_{x,y} G_M(x,y)^2. \tag{35}$$

The lower values of *Energy* represent a higher encryption quality.

In Table 5, we show the results of the three metrics of *GLCM* (*Homogeneity*, *Contrast*, and *Energy*) on the dataset (taking the average of the results of each color channel).

Table 5. Texture analysis results of our dataset.

Image	*Homogeneity*	*Contrast*	*Energy*
Baboon	0.389632	10.476	0.0156291
Lena	0.389445	10.498	0.0156290
Peppers	0.389661	10.489	0.0156291
4.1.03	0.389301	10.473	0.0156289
Barbara	0.389297	10.517	0.0156285
Boat	0.389401	10.503	0.0156287
Cameraman	0.389018	10.519	0.0156297
Zelda	0.389550	10.480	0.0156290

3.6. Key Space

The key space of an encryption algorithm refers to the multitude of unique combinations that can be used to attempt the decryption of the encoded information. The higher the key space, the harder it is to decrypt with brute-force attacks.

The exact key space of the deterministic noise (K_N) is not known, but as mentioned in Section 2.6, the parameters p_1, p_2 and p_3 can generate 256 different outcomes for each column band of the first row of the image, and two different sets of parameters can generate the same outcome for the first row, but with a different outcome for the second row; therefore, the key space K_N is unknown but it can be said that:

$$K_N > 256^3. \tag{36}$$

Regarding the modular discrete derivative, it is trivial that, for a grayscale image of dimensions $M \times N$, the theoretical key space of the modular discrete derivative K_D is equal to the number of derivatives that it can have: $M \times N - 1$; therefore, for an RGB image, the key space is given by Equation (37):

$$K_D = (M * N - 1)^3. \tag{37}$$

Langton's ant needs to be decrypted by having the correct coordinates for the ant's final position, its orientation, and the number of steps S it gave, which can be almost arbitrarily large. The key space of Langton's ant K_L for an RGB image of dimensions $M \times N$ is given by Equation (38):

$$K_L = (4 * S * M * N)^3. \tag{38}$$

Therefore, the key space K for an $M \times N$ RGB image is given by Equation (39):

$$K = K_N^2 * K_D * K_L., \tag{39}$$

which gives us:

$$K > (262144 * S * (M^2 * N^2 - M * N))^3. \tag{40}$$

Assuming a 512×512 RGB image with a maximum of 100 million steps, the key space of the proposed encryption algorithm is larger than 5.8459×10^{72}.

3.7. Differential Attack

Two commonly used metrics to assess the strength of an encryption system against differential attacks are the number of pixel change rate, known as NPCR, and the unified average changing intensity, known as UACI [67].

Let $A(x, y)$ and $B(x, y)$ be two single-band images with dimensions $M \times N$; we can calculate their NPCR and UACI by applying Equations (41) and (43), respectively.

$$\text{NPCR} = \frac{\sum_{x=1}^{N} \sum_{y=1}^{M} D(x, y)}{M \times N} \times 100, \tag{41}$$

where

$$D(x, y) = \begin{cases} 0 & \text{if } A(x, y) - B(x, y) = 0 \\ 1 & \text{in any other case} \end{cases} \tag{42}$$

and

$$\text{UACI} = \frac{\sum_{x=1}^{N} \sum_{y=1}^{M} |A(x, y) - B(x, y)|}{255(M \times N)} \times 100. \tag{43}$$

If the resulting images from the encryption of two nearly identical images have an NPCR close to 100% and a UACI greater than 33%, then we will say that the encryption scheme is strong since a small change in the input of the algorithm produces a vastly different output [67].

We tested these metrics on our proposed algorithm by taking an RGB picture named A, then chose a random pixel from a random color channel, modified its value on its least significant bit, and referred to the resulting picture B. Then we encrypted both A and B by using the exact same parameters; we measure the similarity of the results by calculating the average NPCR and UACI values for their corresponding color channels.

We tested each picture in our dataset a total of 100 times by using the parameters mentioned in Section 3.1; we present the result of the averages for each picture in Table 6.

Table 6. NPCR and UACI results of the dataset.

Image	NPCR (%)	UACI (%)
Baboon	99.60844421	33.44672377
Lena	99.60851796	33.46184906
Peppers	99.61054611	33.46391676
4.1.03	99.61008962	33.45280907
Barbara	99.60963949	33.47430379
Boat	99.60962041	33.46398427
Cameraman	99.60861969	33.46164771
Zelda	99.61025492	33.45394059

3.8. Key Sensitivity

To measure the sensitivity of the key, we performed an encryption of the Lena image and subsequently decrypted it by using a slightly modified decryption key. A comparison was made between the resulting image and the original image, and the NPCR was measured (by averaging the NPCR values across the three color channels).

Starting with the first step of the algorithm, when we use the wrong key for the first deterministic noise (modifying the least significant bit of p_1), we compare the decrypted image with the original image, and obtain an NPCR of 98.13868204%. Taking an incorrect decryption key for the second deterministic noise (modifying the least significant bit of p_1), we obtain an NPCR of 98.16474914%. By performing one extra anti-derivative on each color channel to decrypt the modular discrete derivative, we obtain an NPCR of 99.60238138%, and by performing one less anti-derivative, we obtain an NPCR of 99.60810343%. Regarding Langton's ant, using one extra step on each color channel, we obtain an NPCR of 99.56118265%; positioning the ant in the wrong position (one row down) on each color channel results in an NPCR of 99.71809387%; rotating the ant 180 degrees on each color channel results in an NPCR of 99.56270853%. The images obtained using the previously mentioned wrong decryption keys are shown in Figure 19.

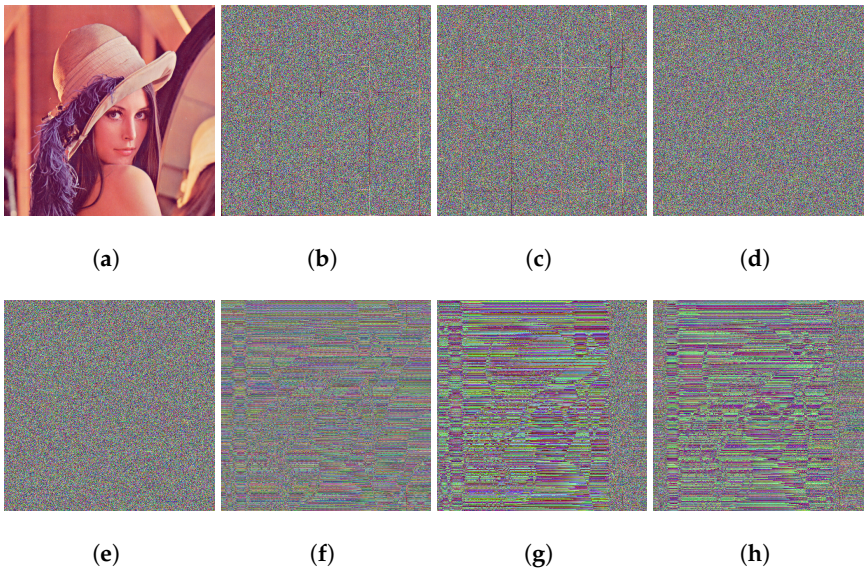


Figure 19. Resulting images from the key sensitivity test. (**a**) Correct decrypted image. (**b**) Incorrect key for the first deterministic noise. (**c**) Incorrect key for the second deterministic noise. (**d**) One extra anti-derivative for MDD. (**e**) One less anti-derivative for MDD. (**f**) Wrong amount of steps for LA. (**g**) Wrong position for LA. (**h**) Wrong orientation for LA.

3.9. Computational Complexity

For an $M \times N$ RGB image, the computational complexity of both the deterministic noise and the cyclic permutation is $\mathcal{O}(M * N)$; MDD has a complexity of $\mathcal{O}(M * N * D)$, where D stands for the degree used in MDD for the derivative; Langton's ant has a complexity of $\mathcal{O}(S)$, where S is the number of steps taken by the ant. In conclusion, the computational complexity of our proposed method is $\mathcal{O}(3 * M * N + M * N * D + S)$, which can be simplified as: $\mathcal{O}(M * N * D + S)$.

4. Comparison with Other Works

We compared our proposed encryption algorithm with three recent works that are also based on chaotic systems and cellular automata. Zhang et al. [23] use a combination of set partitioning in hierarchical trees, cellular automata, and different chaotic systems to encrypt images in a lossless manner. Roy et al. [12] proposed an algorithm referred to as IESCA, which uses cellular automata, referred to as 2D Moore cellular automata, which is

designed for use in resource-constrained IoT devices. The random chaotic sequences were generated by the system through local transformations that relied on the bit states of the cellular automaton's neighbors. In their work, they explored a version of the algorithm by using a periodic boundary for the neighborhood, as well as a null boundary. Dong et al. [5] used hybrid elementary cellular automata composed of a combination of two global rules from a hybrid ECA to improve the chaotic behavior of the Chirikov standard map-based pseudo-random coupled map lattice model.

Tables 7–14 present various comparisons between the metrics obtained in our method and the other algorithms mentioned previously. Specifically, the tables provide comparisons for the chi-square value, entropy, *MSE*, *PSNR*, NPCR, UACI, key space, and encryption time. The comparisons show that our proposed scheme is competitive with the recent works found in the literature.

Table 7. Comparison of the chi-square results with other methods. Values highlighted in bold in each row represent the best results for each image. All images have dimensions of 512 × 512 pixels.

Image	Proposal	[23]	[5]	[12]	[29]
Baboon	**233.951**	261.0619	237.3718	—	—
Lena	260.244	**248.3683**	267.1725	—	—
Peppers	257.839	**243.0886**	—	—	—
Barbara	270.602	**244.3333**	—	—	—
Boat	**263.305**	280.653	—	—	—

Table 8. Comparison of entropy results with other methods. Values highlighted in bold in each row represent the best results for each image. All images have dimensions of 512 × 512 pixels.

Image	Proposal	[23]	[5]	[12]		[29]
				Periodic	Null	
Baboon	**7.9994**	7.9992	—	**7.9994**	7.9919	7.9993
Lena	7.9993	7.9993	—	**7.9997**	7.9879	7.9993
Peppers	7.9993	7.9993	—	7.9971	7.9852	**7.9998**
Barbara	7.9993	7.9992	—	—	—	**7.9994**
Boat	**7.9993**	7.9992	—	—	—	**7.9993**

Table 9. Comparison of MSE between plaintext images and ciphered images with other methods. Values highlighted in bold in each row represent the best results for each image. All images have dimensions of 512 × 512 pixels.

Image	Proposal	[23]	[5]	[12]		[29]
				Periodic	Null	
Baboon	**8614.279**	—	—	7179	7129	—
Lena	**8928.916**	—	—	7129	6967	—
Peppers	**10,111.045**	—	—	9291	9398	—

Table 10. Comparison of PSNR between plaintext images and ciphered images with other methods. Values highlighted in bold in each row represent the best results for each image. All images have dimensions of 512 × 512 pixels.

Image	Proposal	[23]	[5]	[12]		[29]
				Periodic	Null	
Baboon	**8.793**	—	—	9.57	9.68	29.54
Lena	**8.682**	—	**8.68**	9.68	9.75	28.58
Peppers	**8.134**	—	—	8.45	8.46	28.50
Barbara	**8.810**	—	—	—	—	29.19
Boat	**9.295**	—	—	—	—	29.35

Table 11. Comparison of NPCR (%) between plaintext images and ciphered images with other methods. Values highlighted in bold in each row represent the best results for each image. All images have dimensions of 512 × 512 pixels.

Image	Proposal	[23]	[5]	[12]		[29]
				Periodic	Null	
Baboon	99.6084	—	—	**99.7384**	99.7138	99.6967
Lena	99.6085	—	—	99.6347	99.6025	**99.6881**
Peppers	99.6105	—	—	99.6284	99.4435	**99.6937**
Barbara	99.6096	—	—	—	—	**99.9799**
Boat	**99.9096**	—	—	—	—	99.7002

Table 12. Comparison of UACI (%) between plaintext images and ciphered images with other methods. Values highlighted in bold in each row represent the best results for each image. All images have dimensions of 512 × 512 pixels.

Image	Proposal	[23]	[5]	[12]		[29]
				Periodic	Null	
Baboon	**33.4467**	—	—	33.4105	32.9776	32.2756
Lena	33.4618	—	—	33.4653	33.4243	**37.5600**
Peppers	33.4639	—	—	**33.4822**	31.9396	30.8424
Barbara	33.4743	—	—	—	—	**34.6018**
Boat	**33.4639**	—	—	—	—	31.8961

Table 13. Comparison of the key space with other methods.

Proposal	[23]	[5]	[12]	[29]
$>5.8459 \times 10^{72}$	$\approx 1.635 \times 10^{296}$	1.1579×10^{77}	$>2.4179 \times 10^{24}$	$\approx 7.2057 \times 10^{16}$

Table 14. Comparison of encryption time complexity with other methods.

Method	Architecture	RAM	Platform	Size Image (px)	Time (s)
[23]	3.2 GHz Intel Core i5	8 GB	Matlab	$512 \times 512 \times 3$	not given
[5]	3.4 GHz Intel Core i7	16 GB	Matlab	512×512	0.0134
[12]	1.2 GHz ARMv8	1 GB	Python	512×512	not given
[29]	2.2 GHz Intel Pentium-B960	2 GB	not given	512×512	3.007
Proposal	2.1 GHz AMD Ryzen 5	12 GB	Matlab	$512 \times 512 \times 3$	4.360

5. Discussion

After experimentation and testing, we found that using the first and second iterations of MDD was not enough to obtain the full image encryption; therefore, it was necessary to use the third derivative or higher. Regarding Langton's ant, it can be used with any number of iterations since the visual information of the image is hidden by the deterministic noise and MDD, while LA is used to increase the security of the algorithm.

Time efficiency was not a primary consideration during the code development process, so the algorithm was not suitable for stream video encryption/decryption given its encryption time. Future research could focus on optimizing the algorithm and implementing it in a faster programming language, such as C.

Since we take the floor function in every calculation that involves a division, there is no difference between using double (64 bits) and single (32 bits) precision numbers in these calculations. However, the variables p_1, p_2, and p_3 involved in the deterministic noise of Section 2.6 can easily result in integers that are larger than the maximum sizes of single precision integers; therefore, the algorithm cannot be implemented in systems with short data types. Fortunately, the algorithm could potentially be modified to be compatible with these systems by using modular arithmetic on the values of s_τ to prevent the values of p_1, p_2, and p_3 from becoming too large.

Figure 17 illustrates the histograms of each color channel for Lena's image before and after being encrypted, allowing for a visual comparison between them, and showing that in all cases, the encrypted histograms exhibit a uniform distribution, indicating no resemblance to the original histograms.

Additionally, Table 1 shows the chi-square scores for the histograms of the encrypted images, where chi-square values less than 293.247 are indicative that the resulting image has a uniform histogram. Since the values obtained for the chi-square are less than 293.247, they indicate a robust performance to the histogram analysis.

In Table 2, the correlation coefficient (the average for all color channels) between the original and encrypted images is shown. As can be seen, the values are very small in all cases.

In addition to the histogram flatness analysis, in Figure 18, we analyze the intensity of local dependence among the surrounding pixels in a specific direction by measuring the correlation distributions in the vertical direction for each channel in the RGB image. Moreover, the results show that the original image presents a strong correlation among the adjacent pixels, and encrypted channels display a weak correlation, implying random behavior. Testing the horizontal and diagonal directions gives similar results, demonstrating that the direction of the derivative does not impact the final result.

We encrypted all of the images of the dataset and calculated the entropy value of each one, which measures their level of randomness; for example, for an image with 256 gray levels, the maximum entropy value is 8. From Table 3, we can see that the 4.103 image generated the lowest value, the Baboon image generated the highest value, and the Lena image generated an entropy value of 7.999283079.

From the results shown in Figure 16, we can see that the encrypted Lena image presents highly chaotic visual behavior. In addition to a visual inspection of the encrypted image, we calculated some encryption quality metrics, which are based on the deviation in the values of the pixels between the plain image and its encryption. Therefore, the encryption quality is acceptable if pixel changes are maximum and irregular between the plain and encrypted image. Thus, Table 4 shows four encryption quality measures: MD, ID, DU, PSNR, and MSE Higher values for MD, ID, and MSE correspond to a better encryption quality; smaller values of DU and PSNR are expected. From the results, good encryption quality values were obtained over the dataset used.

Regarding texture analysis, using the *GLCM*, we calculated three texture metrics (homogeneity, contrast, and energy) to assess the frequency of dissimilar combinations of gray levels within a particular spatial neighborhood. Thus, Table 5 shows the results of texture analysis conducted over the dataset by taking the average of each color channel. We can see that homogeneity and energy present lower values and contrast presents higher values for all images.

In Section 3.6, we calculated the key space of the proposed encryption method, which represents the number of different combinations that can be tried with brute-force attacks to decrypt an encrypted image. Thus, for a 512×512 RGB image with $S = 100 \times 10^6$ steps, the key space is larger than 5.8459×10^{72}. Therefore, an attacker would take 1.852×10^{64} Gregorian years to test each possible key, using 0.1 s for each one.

We also calculated the NPCR and UACI metrics to analyze the strength of the algorithm against differential attacks. Table 6 shows the results obtained for the dataset used; we can see that the higher NPCR value was 99.61054611% for the *Peppers* image, and the higher UACI value was 33.47430379 for the *Barbara* image, theoretically being 100% of the maximum value for the NPCR value and 33% for the UACI value, indicating that the proposed encryption approach has resistance against differential attacks. Moreover, the analysis of the key sensitivity presented over the Lena image in Section 3.8 shows that if we use a wrong key, which varies slightly from the correct one, we are not able to decrypt the image. By modifying the least significant bit of p_1 for the first deterministic noise, we obtain an NPCR of 98.13868204% and an NPCR of 98.16474914% for the second deterministic noise (by calculating the NPCR for each channel and taking the average). By applying one extra anti-derivative on each color channel to decrypt the modular discrete derivative, we obtain an NPCR of 99.60238138%, and by applying one less anti-derivative, we obtain an NPCR of 99.60810343%. For Langton's ant, by taking one extra step on each color channel, we obtain an NPCR of 99.56118265%; positioning the ant in the wrong position (one row down) on each color channel gives an NPCR of 99.71809387%; rotating the ant 180 degrees on each color channel gives an NPCR of 99.56270853%.

It is relevant to mention that due to the characteristics of the modular discrete derivative, the deterministic noise, and Langton's ant, and as reported in the sensitivity analysis, if any bit of any pixel of the encrypted image is altered, for example, due to attacks on image processing, the original image is not decrypted, showing a high-security level, but a weak performance to recover the original image against intentional or unintentional attacks.

Finally, we performed a comparison with other state-of-the-art works that used similar techniques. Table 7 compares the chi-square values obtained with existing methods. We can see that the proposed method obtained the best results on the *Baboon* and *Boat* images, whereas Zhang et al. [23] obtained the best values for the *Lena*, *Peppers*, and *Barbara* images. Regarding entropy results (Table 8), our proposed method obtained the best results for the *Baboon* and *Boat* images, Roy et al. [12] obtained the best results for the *Baboon* and *Lena* images, and Mondal et al. [29] obtained the best results for the *Peppers*, *Barbara*, and *Boat* images. Regarding the MSE results, the results shown in Table 9 highlight that the best results were obtained with our proposed encryption algorithm for all compared images (*Baboon*, *Lena*, and *Peppers*), which were images where the authors reported results. For the PSNR results (Table 10), our proposed method obtained lower values for all compared

images; Dong et al. [5] obtained a low value only for the *Lena* image. The NPCR results displayed in Table 11 indicate that we only obtained the highest value for the *Boat* image, Roy et al. [12] obtained the highest value for the *Baboon* image, and Mondal et al. [29] obtained the best values for the remaining images (*Lena*, *Peppers*, and *Barbara*). However, for the UACI results (Table 12), we obtained the best results for the *Baboon* and *Boat* images, Roy et al. [12] obtained the best result for the *Peppers* image, and Mondal et al. [29] obtained the best results for the *Lena* and *Barbara* images. Table 13 shows that the proposal by Zhang et al. [23] has the largest key space, which makes it the more robust proposal against the brute force techniques. As the last comparison metric, Table 14 shows a comparison of the encryption time. From these results, we can conclude that the proposed encryption–decryption algorithm based on Langton's ant, modular discrete derivative, and deterministic noise, is competitive with the recent works found in the literature.

6. Conclusions

In this paper, we presented an image encryption system based on the modular discrete derivative, a novel technique used to encrypt images. In addition, we continued the work presented in [1] by improving the use of Langton's ant as an image encryption method and developing a variant of the novel deterministic noise of our previous work. On the one hand, an advantage of Langton's ant is its high key space, but at the cost of its small impact on the visuals of the image. On the other hand, the modular discrete derivative and the deterministic noise have the advantage of creating a significant visual impact on the image, with the disadvantage of having a low key space. In the present work, we managed to combine these methods to take advantage of their strengths and neutralize their weaknesses.

This work contributes to the existing literature by further exploring the strengths and weaknesses of Langton's ant, a cellular automaton that has been explored in only a few works on image encryption. Moreover, this research introduces the use of a modular discrete derivative applied to image encryption, which, to our knowledge, has not been previously studied. By testing the effectiveness of this method, our research demonstrated its potential as a valuable addition to the existing image encryption techniques. Moreover, we found that it is necessary to obtain at least the third derivative of the modular discrete derivative to obtain the full image encryption. Regarding Langton's ant, it can be used with any number of iterations since the visual information of the image is hidden by the deterministic noise and MDD, while Langton's ant is used to increase the security of the algorithm.

Through several tests and experiments, we verified that the proposed algorithm is very secure and reliable if the encryption key is known, being completely reversible, resulting in decrypted images that are identical to the originals with a root mean square error (RMSE) of zero. Our proposed algorithm shows competitive results when compared to the current state of the art, as indicated by metrics such as chi-square, entropy, MSE, PSNR, NPCR, UACI, and key space. However, due to the characteristics of these methods and the results of the sensitivity analysis, if the encrypted image is altered in any way, for example, due to an attacker, the original image is not decrypted, and the original image is lost. Thus, this property represents a strength of the method, presenting a high-security level. However, it is a weak point of the proposal, showing a low performance against intentional or unintentional attacks.

In future work, we will explore new implementations of Langton's ant, modular discrete derivative, and deterministic noise, with better approximations of the key space for the deterministic noise or optimization of the implementation of any of the methods to make them more efficient. In addition, future research could focus on optimizing the algorithm and implementing it in a faster programming language, such as C, for applications such as stream video encryption/decryption. Finally, we will explore the implementation of the proposed method in systems with short data types.

Author Contributions: Conceptualization, E.M.-A., A.R.-A., J.B. and S.L.G.-C.; methodology, E.M.-A., A.R.-A., J.B. and S.L.G.-C.; software, E.M.-A. and A.R.-A.; validation, E.M.-A., A.R.-A., J.B. and S.L.G.-C.; formal analysis, E.M.-A., A.R.-A. and J.B.; investigation, E.M.-A., A.R.-A., J.B. and S.L.G.-C.; resources, E.M.-A., A.R.-A. and S.L.G.-C.; data curation, E.M.-A., A.R.-A. and S.L.G.-C.; writing—original draft preparation, E.M.-A., A.R.-A. and J.B.; writing—review and editing, E.M.-A., A.R.-A., J.B. and S.L.G.-C.; visualization, E.M.-A., A.R.-A.; supervision, E.M.-A., J.B. and A.R.-A.; project administration, E.M.-A. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was partially funded by Vicerrectoría General de Investigación at Universidad Panamericana through the Institutional Program "Fondo Open Access" and by Instituto Politécnico Nacional through the program "SIBE COFAA".

Data Availability Statement: The datasets used are publicly available from: The USC-SIPI Image Dataset [55]: http://sipi.usc.edu/database. Accessed on 13 September 2022. The University of Konstanz [56]: https://cms.uni-konstanz.de/fileadmin/archive/informatik-saupe/fileadmin/informatik/ag-saupe/Webpages/lehre/dip_w0910/demos.html. Accessed on 13 September 2022.

Acknowledgments: Ernesto Moya-Albor, Andrés Romero-Arellano, and Jorge Brieva would like to thank Facultad de Ingeniería of Universidad Panamericana for supporting this work. Andrés Romero-Arellano thanks the Facultad de Ingeniería of Universidad Panamericana for the exceptionally generous scholarship, which has enabled him to pursue his studies further. Additionally, it has provided him with the opportunity to transform and implement classroom projects into formal research endeavors.

Conflicts of Interest: The authors declare there is no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AD	absolute difference
ACM	Arnold's cat map
BCS	block compressive sensing
CA	cellular automata
CS	compressive sensing
CP	cyclic permutation
DAQ	data acquisition system
DCT	discrete cosine transform
DNA	deoxyribonucleic acid
DN	deterministic noise
DU	deviation from the uniform histogram
DWT	discrete wavelet transform
EIs	elemental images
ECA	elementary cellular automata
FSBI	Feistel structure-based bit inversion
FSM	finite state machine
GoL	Game of Life
GLCM	gray-level co-occurrence matrix
HCM	Henon–Chebyshev map
HCNN	Hopfield chaotic neural network
ICA	independent component analysis
IoT	Internet of Things
ID	irregular deviation
LA	Langton's Ant
LM	logistic map
LSS	logistic sine system
LASM	logistic-adjusted sine map
LSCM	logistic-sine-cosine map
MD	maximum deviation
MLCA	maximum length cellular automata

MSE	mean square error
MDD	modular discrete derivative
MCA	Moore cellular automata
NCC	normalized cross-correlation
NPCR	number of pixel change rate
PSNR	peak signal-to-noise ratio
PC-MLCA	programmable complemented–maximum length cellular automata
PRBG	pseudo-random bit generator
PRNG	pseudo-random number generator
RFrDCT	random fractional DCT
ROI	region of interest
RIWT	reversible integer wavelet transform
RMSE	root mean square error
RCA	rule cellular automata
SPIHT	set partitioning in hierarchical trees
SSO	shark smell optimization
UACI	unified average changing intensity
YOLO	You Only Look Once

References

1. Romero-Arellano, A.; Moya-Albor, E.; Brieva, J.; Cruz-Aceves, I.; Avina-Cervantes, J.G.; Hernandez-Gonzalez, M.A.; Lopez-Montero, L.M. Image encryption and decryption system through a hybrid approach using the jigsaw transform and langton's ant applied to retinal fundus images. *Axioms* **2021**, *10*, 215. [CrossRef]
2. Kanso, A.; Ghebleh, M. An efficient and robust image encryption scheme for medical applications. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *24*, 98–116. [CrossRef]
3. Khayyat, M.; Khayyat, M.; Abdel-Khalek, S.; Mansour, R. Blockchain enabled optimal Hopfield Chaotic Neural network based secure encryption technique for industrial internet of things environment. *Alex. Eng. J.* **2022**, *61*, 11377–11389. [CrossRef]
4. Li, L.; Luo, Y.; Qiu, S.; Ouyang, X.; Cao, L.; Tang, S. Image encryption using chaotic map and cellular automata. *Multimed. Tools Appl.* **2022**, *81*, 40755–40773. [CrossRef]
5. Dong, Y.; Zhao, G.; Ma, Y.; Pan, Z.; Wu, R. A novel image encryption scheme based on pseudo-random coupled map lattices with hybrid elementary cellular automata. *Inf. Sci.* **2022**, *593*, 121–154. [CrossRef]
6. Rupa, I.; Manideep, K.; Kamale, N.; Suhasini, S. Information Security using Chaotic Encryption and Decryption of Digital Images. In Proceedings of the 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 15–16 July 2022. [CrossRef]
7. Lv, W.; Chen, J.; Chai, X.; Fu, C. A robustness-improved image encryption scheme utilizing Life-liked cellular automaton. *Nonlinear Dyn.* **2022**, *111*, 3887–3907. [CrossRef]
8. Kafetzis, I.; Moysis, L.; Volos, C.; Nistazakis, H.; Munoz-Pacheco, J.; Stouboulos, I. Automata-Derived Chaotic Image Encryption Scheme. In Proceedings of the 2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 8–10 June 2022. [CrossRef]
9. Boudali, A.; Said, N.; Ali-Pacha, A. A new symmetrical cryptosystem based cellular automata and chaotic map function. *J. Discret. Math. Sci. Cryptogr.* **2022**, *25*, 1435–1455. [CrossRef]
10. Kang, S.; Choi, U.; Cho, S. Fast image encryption algorithm based on (n, m, k)-PCMLCA. *Multimed. Tools Appl.* **2022**, *81*, 1209–1235. [CrossRef]
11. Chong, J.; Xie, S.; Liu, D. Chaotic Block Color Image Encryption Algorithm Using Cellular Automata and DNA Sequence. In Proceedings of the AIPR 2021: 2021 4th International Conference on Artificial Intelligence and Pattern Recognition, Xiamen China, 24–26 September 2021; pp. 386–392. [CrossRef]
12. Roy, S.; Shrivastava, M.; Rawat, U.; Pandey, C.; Nayak, S. IESCA: An efficient image encryption scheme using 2D cellular automata. *J. Inf. Secur. Appl.* **2021**, *61*, 102919. [CrossRef]
13. Kumar, A.; Raghava, N. An efficient image encryption scheme using elementary cellular automata with novel permutation box. *Multimed. Tools Appl.* **2021**, *80*, 21727–21750. [CrossRef]
14. Jeelani, Z.; Qadir, F. A comparative study of cellular automata-based digital image scrambling techniques. *Evol. Syst.* **2021**, *12*, 359–375. [CrossRef]
15. Alexan, W.; Elbeltagy, M.; Aboshousha, A. Lightweight Image Encryption: Cellular Automata and the Lorenz System. In Proceedings of the 2021 International Conference on Microelectronics (ICM), New Cairo City, Egypt , 19–22 December 2021; pp. 34–39. [CrossRef]
16. Song, X.; Shi, M.; Zhou, Y.; Wang, E. An Block Image Encryption Algorithm Based on Reversible Cellular Automata. In Proceedings of the 2021 IEEE 21st International Conference on Communication Technology (ICCT), Tianjin, China, 13–16 October 2021; pp. 1167–1172. [CrossRef]

17. Kang, S.; Choi, U. ROI Image Encryption using YOLO and Chaotic Systems. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 466–474. [CrossRef]
18. Gan, Z.; Chai, X.; Zhang, J.; Zhang, Y.; Chen, Y. An effective image compression–encryption scheme based on compressive sensing (CS) and Game of Life (GOL). *Neural Comput. Appl.* **2020**, *32*, 14113–14141. [CrossRef]
19. Ping, P.; Zhang, X.; Yang, X.; Mao, Y.; Gao, Z. Parallel Image Encryption Technology Based on Cellular Automaton. In Proceedings of the 2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK, 3–6 August 2020; pp. 216–223. [CrossRef]
20. Choi, U.; Cho, S.; Kim, J.; Kang, S.; Kim, H. Color image encryption based on programmable complemented maximum length cellular automata and generalized 3-D chaotic cat map. *Multimed. Tools Appl.* **2020**, *79*, 22825–22842. [CrossRef]
21. Choi, U.; Cho, S.; Kang, S. High speed color image encryption using pixel shuffling with 1-D MLCA. In Proceedings of the 2020 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 15–18 May 2020; pp. 373–377. [CrossRef]
22. Naskar, P.; Bhattacharyya, S.; Nandy, D.; Chaudhuri, A. A robust image encryption scheme using chaotic tent map and cellular automata. *Nonlinear Dyn.* **2020**, *100*, 2877–2898. [CrossRef]
23. Zhang, H.; Wang, X.Q.; Sun, Y.J.; Wang, X.Y. A novel method for lossless image compression and encryption based on LWT, SPIHT and cellular automata. *Signal Process. Image Commun.* **2020**, *84*, 115829. [CrossRef]
24. Chai, X.; Fu, X.; Gan, Z.; Zhang, Y.; Lu, Y.; Chen, Y. An efficient chaos-based image compression and encryption scheme using block compressive sensing and elementary cellular automata. *Neural Comput. Appl.* **2020**, *32*, 4961–4988. [CrossRef]
25. Geng, S.; Wu, T.; Wang, S.; Zhang, X.; Wang, Y. Image Encryption Algorithm Based on Block Scrambling and Finite State Machine. *IEEE Access* **2020**, *8*, 225831–225844. [CrossRef]
26. Eslami, Z.; Kabirirad, S. A block-based image encryption scheme using cellular automata with authentication capability. *AIP Conf. Proc.* **2019**, *2183*, 080002. [CrossRef]
27. Qi, F.; Huang, S.; Li, T.; Yang, H.; Kang, X. 2D henon-chebyshev chaotic map for image encryption. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 10–12 August 2019; pp. 774–781. [CrossRef]
28. Wu, J.; Cao, X.; Liu, X.; Ma, L.; Xiong, J. Image encryption using the random FrDCT and the chaos-based game of life. *J. Mod. Opt.* **2019**, *66*, 764–775. [CrossRef]
29. Mondal, B.; Singh, S.; Kumar, P. A secure image encryption scheme based on cellular automata and chaotic skew tent map. *J. Inf. Secur. Appl.* **2019**, *45*, 117–130. [CrossRef]
30. Khan, S.; Han, L.; Lu, H.; Butt, K.; Bachira, G.; Khan, N.U. A New Hybrid Image Encryption Algorithm Based on 2D-CA, FSM-DNA Rule Generator, and FSBI. *IEEE Access* **2019**, *7*, 81333–81350. [CrossRef]
31. Li, X.; Wang, Y.; Wang, Q.H.; Liu, Y.; Zhou, X. Modified integral imaging reconstruction and encryption using an improved SR reconstruction algorithm. *Opt. Lasers Eng.* **2019**, *112*, 162–169. [CrossRef]
32. Seshadhri, A.; Chandrasaker, L. 2D-Noise Generation Aided by Chaotic Map, Reversible Integer Wavelet Transform and Cellular Automata. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018; pp. 292–297. [CrossRef]
33. Ben Slimane, N.; Aouf, N.; Bouallegue, K.; MacHhout, M. Hash key-based image cryptosystem using chaotic maps and cellular automata. In Proceedings of the 2018 15th International Multi-Conference on Systems, Signals & Devices (SSD), Yasmine Hammamet, Tunisia, 19–22 March 2018; pp. 190–194. [CrossRef]
34. Rajagopalan, S.; Sivaraman, R.; Upadhyay, H.; Rayappan, J.; Amirtharajan, R. ON–Chip peripherals are ON for chaos – an image fused encryption. *Microprocess. Microsyst.* **2018**, *61*, 257–278. [CrossRef]
35. Ping, P.; Wu, J.; Mao, Y.; Xu, F.; Fan, J. Design of image cipher using life-like cellular automata and chaotic map. *Signal Process.* **2018**, *150*, 233–247. [CrossRef]
36. Rajagopalan, S.; Rethinam, S.; Arumugham, S.; Upadhyay, H.; Rayappan, J.; Amirtharajan, R. Networked hardware assisted key image and chaotic attractors for secure RGB image communication. *Multimed. Tools Appl.* **2018**, *77*, 23449–23482. [CrossRef]
37. Chai, X.; Zheng, X.; Gan, Z.; Han, D.; Chen, Y. An image encryption algorithm based on chaotic system and compressive sensing. *Signal Process.* **2018**, *148*, 124–144. [CrossRef]
38. Sharma, S.; Kaur, N. Hybridization of ICA based on arnold cat map using reversible cellular automata for faster cryptographic speed. In Proceedings of the 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 21–23 December 2017; Volume 2018, pp. 563–566. [CrossRef]
39. Hanis, S.; Amutha, R. Double image compression and encryption scheme using logistic mapped convolution and cellular automata. *Multimed. Tools Appl.* **2018**, *77*, 6897–6912. [CrossRef]
40. Li, M.; Lu, D.; Wen, W.; Ren, H.; Zhang, Y. Cryptanalyzing a Color Image Encryption Scheme Based on Hybrid Hyperchaotic System and Cellular Automata. *IEEE Access* **2018**, *6*, 47102–47111. [CrossRef]
41. Bhardwaj, R.; Sharma, V. Effective image encryption technique through 2D cellular automata. *Adv. Intell. Syst. Comput.* **2018**, *518*, 39–46. [CrossRef]

42. Rajagopalan, S.; Subramani, D.; Ananthanarayanan, A.; Rethinam, S.; Upadhyay, H.; Amirtharajan, R. Optic assisted image encryption: Confluence of chaos and virtual instrumentation. In Proceedings of the 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 5–7 January 2017. [CrossRef]
43. Liang, S.L.; Chai, Z.Q.; Zhang, L.; Wu, Y.S.; Cao, C.L. Image encryption method based on partial X type cellular automaton. *Jilin Daxue Xuebao (Gongxueban)/J. Jilin Univ.* **2017**, *47*, 1653–1660. [CrossRef]
44. Chai, X.; Gan, Z.; Yang, K.; Chen, Y.; Liu, X. An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations. *Signal Process. Image Commun.* **2017**, *52*, 6–19. [CrossRef]
45. Yaghouti Niyat, A.; Moattar, M.; Niazi Torshiz, M. Color image encryption based on hybrid hyperchaotic system and cellular automata. *Opt. Lasers Eng.* **2017**, *90*, 225–237. [CrossRef]
46. Burak, D. Parallelization of image encryption algorithm based on Game of Life and chaotic system. In *Artificial Intelligence and Soft Computing. ICAISC 2017, Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 11–15 June 2017*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10246, pp. 422–431. [CrossRef]
47. Tutueva, A.V.; Nepomuceno, E.G.; Karimov, A.I.; Andreev, V.S.; Butusov, D.N. Adaptive chaotic maps and their application to pseudo-random numbers generation. *Chaos Solitons Fractals* **2020**, *133*, 109615. [CrossRef]
48. Tutueva, A.; Nepomuceno, E.G.; Moysis, L.; Volos, C.; Butusov, D. Adaptive Chaotic Maps in Cryptography Applications. *Stud. Big Data* **2022**, *102*, 193–205. [CrossRef]
49. Daoui, A.; Yamni, M.; Chelloug, S.A.; Wani, M.A.; El-Latif, A.A.A. Efficient Image Encryption Scheme Using Novel 1D Multiparametric Dynamical Tent Map and Parallel Computing. *Mathematics* **2023**, *11*, 1589. [CrossRef]
50. Nepomuceno, E.G.; Nardo, L.G.; Arias-Garcia, J.; Butusov, D.N.; Tutueva, A. Image encryption based on the pseudo-orbits from 1D chaotic map. *Chaos* **2019**, *29*, 061101. [CrossRef] [PubMed]
51. Askar, S.; Al-khedhairi, A.; Elsonbaty, A.; Elsadany, A. Chaotic discrete fractional-order food chain model and hybrid image encryption scheme application. *Symmetry* **2021**, *13*, 161. [CrossRef]
52. Al-Khedhairi, A.; Elsonbaty, A.; Elsadany, A.A.; Hagras, E.A.A. Hybrid Cryptosystem Based on Pseudo Chaos of Novel Fractional Order Map and Elliptic Curves. *IEEE Access* **2020**, *8*, 57733–57748. [CrossRef]
53. Nardo, G.; Nepomuceno, E.G.; Arias-Garcia, J.; Butusov, D.N. Image encryption using finite-precision error. *Chaos Solitons Fractals* **2019**, *123*, 69–78. [CrossRef]
54. Zhou, S.; Wang, X.; Zhang, Y. Novel image encryption scheme based on chaotic signals with finite-precision error. *Inf. Sci.* **2023**, *621*, 782–798. [CrossRef]
55. University of Southern California. The USC-SIPI Image Database. Available online: http://sipi.usc.edu/database (accessed on 27 January 2023).
56. Universität Konstanz. Test Images and Demos. 2009. Available online: https://cms.uni-konstanz.de/fileadmin/archive/informatik-saupe/fileadmin/informatik/ag-saupe/Webpages/lehre/dip_w0910/demos.html (accessed on 1 June 2022).
57. Wussing, H. *The Genesis of the Abstract Group Concept. A Contribution to the History of the Origin of Abstract Group Theory*; Dover Publications, Inc.: Mineola, NY, USA, 2007.
58. Weisstein, E.W. Cyclic Permutation. Technical Report, MathWorld—A Wolfram Web Resource, 2021. Available online: https://mathworld.wolfram.com/CyclicPermutation.html (accessed on 21 June 2021).
59. Langton, C.G. Studying artificial life with cellular automata. *Physica D* **1986**, *22*, 120–149. [CrossRef]
60. Hagiwara, T.; Tsukiji, T. Hardness of Approximation for Langton's Ant on a Twisted Torus. *Algorithms* **2020**, *13*, 344. [CrossRef]
61. Wang, X.; Xu, D. A novel image encryption scheme using chaos and Langton's Ant cellular automaton. *Nonlinear Dyn.* **2015**, *79*, 2449–2456. [CrossRef]
62. Nazir, H.; Bajwa, I.S.; Abdullah, S.; Kazmi, R.; Samiullah, M. A Color Image Encryption Scheme Combining Hyperchaos and Genetic Codes. *IEEE Access* **2022**, *10*, 14480–14495. [CrossRef]
63. Ahmad, J.; Ahmed, F. Efficiency Analysis and Security Evaluation of Image Encryption Schemes. *Int. J. Video Image Process. Netw. Secur.* **2012**, *12*, 18–31.
64. Arif, J.; Khan, M.A.; Ghaleb, B.; Ahmad, J.; Munir, A.; Rashid, U.; Al-Dubai, A.Y. A Novel Chaotic Permutation-Substitution Image Encryption Scheme Based on Logistic Map and Random Substitution. *IEEE Access* **2022**, *10*, 12966–12982. [CrossRef]
65. Haralick, R.M.; Dinstein, I.; Shanmugam, K. Textural Features for Image Classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *SMC-3*, 610–621. [CrossRef]
66. Sun, S.; Guo, Y. A New Hyperchaotic Image Encryption Algorithm Based on Stochastic Signals. *IEEE Access* **2021**, 9, 144035–144045. [CrossRef]
67. Stoyanov, B.; Kordov, K. Image Encryption Using Chebyshev Map and Rotation Equation. *Entropy* **2015**, *17*, 2117–2139. [CrossRef]