*mathematics*

MDPI

*Article*

# Drone-Captured Wildlife Data Encryption: A Hybrid 1D–2D Memory Cellular Automata Scheme with Chaotic Mapping and SHA-256

**Akram Belazi [1,2] and Héctor Migallón [2,*]**

[1] Laboratory RISC-ENIT (LR-16-ES07), Tunis El Manar University, Tunis 1002, Tunisia; akram.belazi@enit.utm.tn or abelazi@umh.es
[2] Department of Computer Engineering, Miguel Hernández University, 03202 Elche, Spain
* Correspondence: hmigallon@umh.es; Tel.: +34-966-65-8390

**Abstract:** In contemporary wildlife conservation, drones have become essential for the non-invasive monitoring of animal populations and habitats. However, the sensitive data captured by drones, including images and videos, require robust encryption to prevent unauthorized access and exploitation. This paper presents a novel encryption algorithm designed specifically for safeguarding wildlife data. The proposed approach integrates one-dimensional and two-dimensional memory cellular automata (1D MCA and 2D MCA) with a bitwise XOR operation as an intermediate confusion layer. The 2D MCA, guided by chaotic rules from the sine-exponential (SE) map, utilizes varying neighbor configurations to enhance both diffusion and confusion, making the encryption more resilient to attacks. A final layer of 1D MCA, controlled by pseudo-random number generators, ensures comprehensive diffusion and confusion across the image. The SHA-256 hash of the input image is used to derive encryption parameters, providing resistance against plaintext attacks. Extensive performance evaluations demonstrate the effectiveness of the proposed scheme, which balances security and complexity while outperforming existing algorithms.

**Keywords:** wildlife conservation; drone data encryption; chaotic systems; memory cellular automata; diffusion; confusion; image security

**MSC:** 37M99



**Citation:** Belazi, A.; Migallón, H. Drone-Captured Wildlife Data Encryption: A Hybrid 1D–2D Memory Cellular Automata Scheme with Chaotic Mapping and SHA-256. *Mathematics* **2024**, *12*, 3602. https://doi.org/10.3390/math12223602

Academic Editor: Lingfeng Liu

Received: 17 October 2024
Revised: 9 November 2024
Accepted: 11 November 2024
Published: 18 November 2024



## 1. Introduction

Drones now stand at the forefront of wildlife conservation, enabling the precise, non-intrusive monitoring of animal populations and habitats [1–6]. This sensitive data, however, must be secured; if intercepted, it could be exploited for poaching or habitat disruption, endangering ecosystems [7]. Encryption, therefore, is not merely technical but an ethical safeguard, ensuring data access is limited to authorized users and aligning drone technology with conservation principles.

In this context, encryption methods based on chaotic systems and cellular automata have gained significant attention due to their inherent non-linear properties, which make them particularly suitable for securing sensitive data. Chaotic systems, characterized by their deterministic yet unpredictable behavior, offer a robust framework for encryption [8–10]. These systems are highly sensitive to initial conditions, making it extremely difficult for unauthorized parties to decipher encrypted data without precise knowledge of the encryption parameters [11,12]. Moreover, cellular automata, which consist of discrete grids of cells that evolve over time according to specific rules, also exhibit complex, non-linear behavior that can be harnessed for encryption [13–15]. These systems can generate pseudo-random sequences that are highly effective for cryptographic applications, providing an additional layer of security for drone-captured images [16,17].

Recent literature has extensively explored the application of chaos theory and cellular automata in encryption schemes. For instance, Xu et al. discussed the use of chaotic maps in securing image data, highlighting their effectiveness in producing high levels of entropy, which is crucial for robust encryption [18]. Similarly, Singh et al. examined cellular automata-based encryption techniques, noting their flexibility and efficiency in handling large datasets, such as those generated by drone surveillance [19]. Furthermore, Zhang et al. provided a comprehensive review of hybrid encryption systems that combine chaos theory with other cryptographic methods to enhance security in real-time data transmission [20].

The potential of these systems is further demonstrated by recent advancements in their implementation. Liu et al. developed an encryption algorithm based on a chaotic system that was shown to outperform traditional methods in terms of speed and security [21]. Meanwhile, Bhat et al. explored the integration of cellular automata with machine learning techniques to create adaptive encryption models that can dynamically adjust to different security needs [22]. These studies underscore the growing importance of chaos and cellular automata in the development of next-generation encryption technologies, especially in fields like wildlife conservation where data security is paramount.

In this work, we introduce a novel encryption algorithm designed to safeguard wildlife data captured via drones. This approach leverages the integration of one-dimensional and two-dimensional memory cellular automata (1D MCA and 2D MCA), combined with an intermediate confusion layer via a bitwise XOR operation. Initially, the original image undergoes transformation using a 2D MCA, governed by chaotic rules derived from the sine-exponential (SE) map, which exhibits chaotic behavior akin to HD maps with the complexity of a 1D map [23]. The 2D MCA sequentially employs three neighbor configurations—periodic-boundary neighbors, hexagonal-boundary neighbors, and Moore neighbors—to enhance both diffusion and confusion. This process ensures that minor alterations in the image propagate widely (diffusion) while introducing non-linear transformations that obscure pixel relationships (confusion). The varied neighbor configurations further diversify and complicate the encryption patterns, making the encrypted image more resistant to reverse-engineering. Next, a bitwise XOR operation is performed using a chaotic matrix derived from the SE map. This step enhances confusion by altering each pixel's value in a non-linear manner, making the relationship between the original and encrypted pixels more obscure. Finally, the 1D MCA, controlled by random rules generated from a pseudo-random number generator (PRNG), is applied. This step primarily contributes to diffusion by ensuring that small changes in the initial state or key affect the entire image. Additionally, it supports confusion by generating complex, non-linear transformations, further complicating the relationship between the original and encrypted images. The SHA-256 hash of the input image is used to derive the SE map parameters and the PRNG's seed, ensuring unique keystreams for each image. This process provides strong resistance against known or chosen plaintext attacks. An extensive performance evaluation demonstrates the proposed scheme's balance between security and complexity, and a comparative analysis with state-of-the-art algorithms underscores its efficiency.

The primary contributions of this paper are threefold: (i) The integration of 2D and 1D MCA significantly enhances both diffusion and confusion, making the encryption scheme more robust. The 2D MCA manages spatial relationships and broad diffusion, while the 1D MCA introduces sequence complexity and strong non-linear transformations, creating a multi-dimensional encryption approach that resists cryptanalysis. (ii) Utilizing different neighbor configurations in 2D MCA further strengthens both diffusion and confusion by introducing varied pixel interactions, complicating patterns, and making the encryption more resilient to attacks exploiting predictable correlations. (iii) To the best of our knowledge, this is the first attempt to sequentially employ varying neighbor configurations within 2D MCA.

The proposed algorithm is designed with remarkable versatility, capable of encrypting a wide range of images, including standard, medical, and wildlife images. Our focus on wildlife preservation stems from two primary motivations: (i) The algorithm's low com-

plexity makes it ideally suited for real-time applications in computationally constrained environments, such as drones. This efficiency is vital in scenarios requiring swift processing or in systems where hardware limitations are significant. (ii) In light of the accelerating impact of climate change on ecological balance and the alarming rise in unregulated hunting, we seek to underscore the urgency of wildlife protection. Through encryption applied to drones used for monitoring wildlife, our approach safeguards sensitive visual data, reinforcing the importance of securing images in applications where data confidentiality is paramount. By doing so, we aim to bolster conservation efforts, helping to ensure the safety and stability of vulnerable animal populations.

The remainder of the paper is structured as follows: Section 2 reviews state-of-the-art image encryption techniques utilizing 1D and 2D MCA. Section 3 discusses the foundational concepts of cellular automata and the sine-exponential map. Section 4 offers a brief overview of wildlife data protection. Section 5 describes the proposed encryption algorithm, with the analysis presented in Section 6. Section 7 presents the security and performance validation of the proposed MCA architecture, while Section 8 articulates the concluding remarks.

## 2. Related Works

Recent developments in image encryption have prominently featured the use of cellular automata (CA), with both one-dimensional (1D) and two-dimensional (2D) memory cellular automata (MCA) gaining attention for their cryptographic potential. Souyah and Faraoun utilized 1D MCA combined with chaotic maps to introduce strong diffusion and confusion properties, ensuring that even slight changes in the input lead to substantial differences in the output, thereby enhancing security [24]. Jeyaram and Raghavan also leveraged 1D CA for pixel permutation, emphasizing the rule-based approach that ensures sensitivity to initial conditions, a key feature in preventing unauthorized decryption [25].

On the 2D MCA front, Roy et al. applied multiple neighborhood templates to improve spatial diffusion, which helps in spreading minor pixel alterations across the entire image, making it harder for attackers to predict the encryption pattern [26]. Wang and Luan further strengthened 2D MCA's utility by integrating it with chaotic maps, achieving a robust encryption scheme that effectively counters differential attacks [27]. Additionally, Habibipour et al. proposed a dynamic rule selection mechanism within a hybrid 1D chaotic map and 2D MCA framework, optimizing both security and computational efficiency [28].

Beyond these foundational works, Habibipour et al. introduced a hybrid encryption scheme that marries the strengths of 1D chaotic map and 2D MCA, capitalizing on the sequence complexity of 1D chaotic map and the spatial diffusion of 2D MCA. This combination proved effective in balancing the computational load while maintaining high security [28]. Similarly, Haque et al. demonstrated that by integrating 1D chaotic map and 2D MCA, the encryption scheme could incorporate various neighbor configurations and rule sets, thereby significantly enhancing resistance to cryptanalysis [29].

Moreover, recent advancements by Ismail et al. in adaptive MCA-based encryption have shown promise, where 1D MCA dynamically adjusts based on the image content [30]. This adaptive approach could be further enhanced by combining it with 2D MCA, allowing for even more complex and unpredictable encryption schemes. Furthermore, other studies, such as those by Pokkuluri et al., explored the integration of MCA with deep learning techniques, providing a new dimension to CA-based encryption by enabling rule optimization through learning algorithms [31]. This integration paves the way for more intelligent and responsive encryption systems that can adapt to varying security needs.

## 3. Preliminaries

### *3.1. Cellular Automata*

Cellular automata (CA) are a class of discrete dynamical systems introduced by John von Neumann [32], governed by local rules. CA effectively model the emergence and

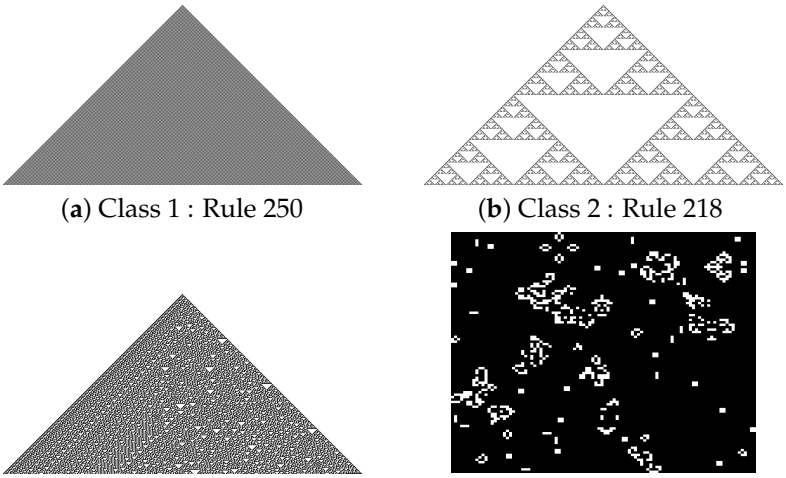
complex behavior observed in various natural systems [33]. The structure and function of a cellular automaton are defined by four key elements:

1. A grid of identical entities called cells, arranged in a circular register. Each cell can occupy one of a finite number of possible states.
2. The specific states that the cells can assume.
3. The neighborhood of a cell, which includes adjacent cells that influence its state.
4. Transition rules that dictate how the state of a cell evolves over time based on the states of its neighbors.

The grid of cells is typically arranged as either a 1D or 2D array. The output at each discrete time step is a new grid with the same dimensions as the initial configuration. These grids, referred to as CA configurations, form patterns when visualized over successive iterations. These patterns can be classified into four distinct categories based on the behavior of the resulting configurations [34,35]:

- Class 1: The system evolves towards a simple, homogeneous final state, regardless of the initial configuration.
- Class 2: The system may reach different final states, but the resulting pattern consists of stable or periodically repeating structures.
- Class 3: The system exhibits random behavior, with unpredictable structures appearing throughout the pattern.
- Class 4: The system displays a mix of order and randomness, producing complex behavior.

Examples of patterns from each of these four classes for a 1D CA are illustrated in Figure 1. In cryptography, the randomness inherent to Class 3 can be leveraged to introduce confusion and diffusion within ciphertext, enhancing security.



(**a**) Class 1 : Rule 250 (**b**) Class 2 : Rule 218

(**c**) Class 3 : Rule 30 (**d**) Class 4 : Conway's Game of Life

**Figure 1.** Examples of patterns corresponding to the four classes of a 1D cellular automaton.

#### 3.1.1. Elementary CA

An elementary cellular automaton (CA) is a fundamental class of CA with a one-dimensional grid where each cell has two possible states, 0 or 1. The neighborhood includes the cell itself and its $r$ left and right neighbors, totaling $2r + 1$ cells, with $r$ representing the radius. The transition rule is typically defined as follows:

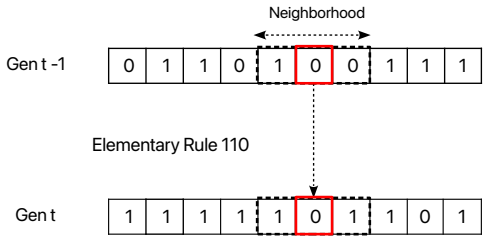
$$S_i^t = F\{S_{i-r}^{t-1}, S_{i-r+1}^{t-1}, \ldots, S_i^{t-1}, \ldots, S_{i+r-1}^{t-1}, S_{i+r}^{t-1}\}, \quad (1)$$

where $S_i^t$ denotes the state of the $i$-th cell at discrete time $t$, and $F$ is a function that depends on the states of the neighborhood surrounding cell $i$. A configuration, $C_i$, with a total of $N$ cells is defined as the set of all cell states at a given time, $t$:

$$C_i = \{S_0^t, S_1^t, \ldots, S_{N-1}^t\}. \quad (2)$$

The discrete time at which a configuration is defined is referred to as a generation. The state of each cell at generation $t$ is determined by the states of its neighborhood at the preceding generation, $t-1$.

The operation of the simplest possible 1D CA, with cell states of 0 or 1 and a radius of $r = 1$, is illustrated in Figure 2. In this case, the neighborhood size is 3, allowing for $2^3 = 8$ possible configurations, as shown in Figure 2. For each neighborhood pattern in the current generation, a binary output is assigned to the corresponding cell in the next generation. Consequently, a set of eight binary digits, known as the ruleset, is required to define the transition rules. Since there are $2^8 = 256$ possible ways to configure these rulesets, there are 256 distinct rules that can be used to generate the next configuration. The ruleset depicted in Figure 2 is known as Rule 110, as the decimal equivalent of the binary digits is 110.




Rule of CA definition:

r =1, rule $01101110_2 = 110_{10}$

| Number | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Neighborhood pattern | 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 |
| Transition Rule | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

**Figure 2.** CA evolution and corresponding ruleset.

### 3.1.2. Memory Cellular Automata

The standard cellular automata (CA) discussed previously are memoryless, as the configuration at generation $t$ depends only on the state at $t-1$. In contrast, memory cellular automata (MCA) consider states from multiple prior generations, not just the immediate past. This class of CA is reversible, enabling the original configuration to be recovered using the inverse MCA.

A $k$-th-order MCA implies that the configuration at generation $t$, $C_t$, depends on the configurations from the previous $k$ generations, $C_{t-1}, C_{t-2}, \ldots, C_{t-k}$. Additionally, it can be shown that, in order to ensure the reversibility of a $k$-th-order MCA, $k-1$ transition rules are sufficient to retrieve the original configuration. Therefore, a $k$-th-order MCA is mathematically defined as follows:

$$C_t = F_1(C_{t-1}) \oplus F_2(C_{t-2}) \oplus \cdots \oplus F_{k-1}(C_{t-k+1}) \oplus (C_{t-k}), \quad (3)$$

where $F_1, F_2, \ldots, F_{k-1}$ are the transition rules, and $\oplus$ denotes the bitwise XOR operation. Equation (3) is reversible for any set of transition rules, and the corresponding inverse MCA is defined by the following:

$$P_t = F_{k-1}(P_{t-1}) \oplus F_{k-2}(P_{t-2}) \ \oplus \cdots \oplus F_1(P_{t-k+1}) \oplus (P_{t-k}), \quad (4)$$

where $P_{t-i}$, with $i \in \{0, 1, \ldots, k\}$, represents the possible configurations of the inverse MCA.

The reversibility of MCA can be demonstrated as follows. Suppose the configuration $C_t$ is obtained by applying the $k$-th-order MCA to the $k$ previous configurations $C_{t-1}, C_{t-2}, \ldots, C_{t-k}$ using Equation (3). We aim to prove that the configuration $C_{t-k}$ can be recovered from the latest $k$ configurations, $C_t, C_{t-1}, \ldots, C_{t-k+1}$ using the inverse MCA defined by Equation (4). Note that the configurations are arranged in reverse order when applying the inverse MCA, such that

$$P_{t-1} = C_{t-k+1}, P_{t-2} = C_{t-k+2}, \ldots$$

$$P_{t-k+1} = C_{t-1}, P_t = C_t.$$

Thus, Equation (4) becomes

$$\begin{aligned} P_t &= F_{k-1}(P_{t-1}) \oplus F_{k-2}(P_{t-2}) \oplus \cdots \oplus F_1(P_{t-k+1}) \oplus (P_{t-k}) \\ &= F_{k-1}(C_{t-k+1}) \oplus F_{k-2}(C_{t-k+2}) \oplus \cdots \oplus F_1(C_{t-1}) \oplus (C_t) \\ &= F_1(C_{t-1}) \oplus F_2(C_{t-2}) \oplus \cdots \oplus F_{k-1}(C_{t-k+1}) \oplus (C_t), \end{aligned} \tag{5}$$

Substituting the value of $C_t$ from Equation (3) into Equation (5), we obtain

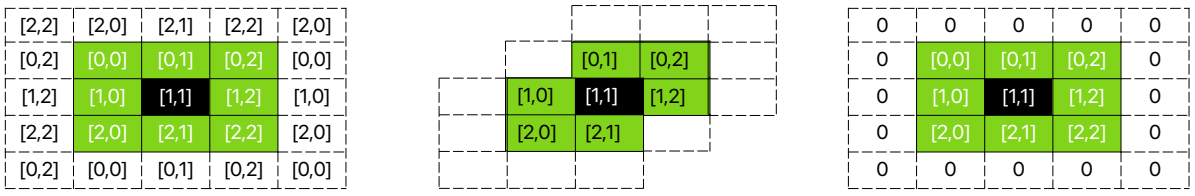
$$\begin{aligned} P_t &= F_1(C_{t-1}) \oplus F_2(C_{t-2}) \oplus \cdots \oplus F_{k-1}(C_{t-k+1}) \oplus F_1(C_{t-1}) \oplus \\ &\quad F_2(C_{t-2}) \oplus \cdots \oplus F_{k-1}(C_{t-k+1}) \oplus (C_{t-k}) \\ &= [F_1(C_{t-1}) \oplus F_1(C_{t-1})] \oplus [F_2(C_{t-2}) \oplus F_2(C_{t-2})] \ldots \\ &\quad \oplus [F_{k-1}(C_{t-k+1}) \oplus F_{k-1}(C_{t-k+1})] \oplus (C_{t-k}) \\ &= 0 \oplus 0 \cdots \oplus 0 \oplus (C_{t-k}) \\ &= C_{t-k}. \end{aligned} \tag{6}$$

Hence, the configuration $C_{t-k}$ can always be recovered using the MCA defined by Equation (4) and the configurations $C_t, C_{t-1}, \ldots, C_{t-k+1}$. Therefore, Equation (4) represents the inverse MCA corresponding to Equation (3).

#### 3.1.3. 2D Memory Cellular Automata

In this study, we examine three neighbor configurations in 2D cellular automata (CA)—periodic-boundary, hexagonal-boundary, and Moore neighbors—to enhance cryptographic robustness. By combining these configurations, we leverage the periodic boundary's uniformity, the hexagonal boundary's complexity, and the Moore neighbors' comprehensive diffusion to strengthen encryption resilience and maintain efficient computation.

Figure 3 illustrates periodic-boundary and Moore neighbors, where each cell's state, represented by black cells at position $(i, j)$, depends on its surrounding cells. Each cell in the grid can be in one of two states, '0' or '1', with its future state influenced by neighbors, ensuring robust diffusion of information across the grid, which is vital for secure encryption.



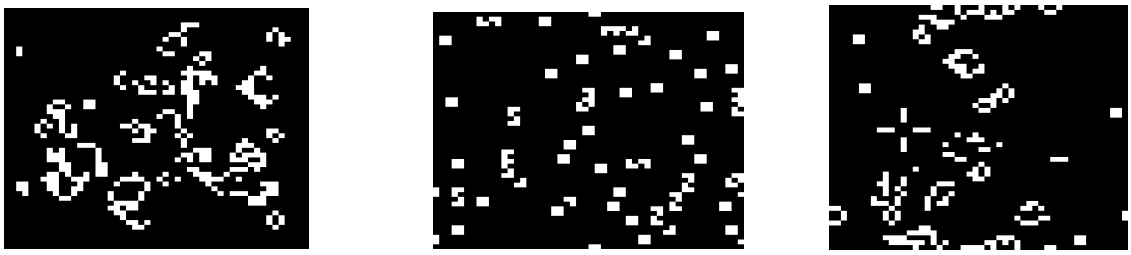

(**a**) Periodic-boundary neighbors (**b**) Hexagonal-boundary neighbors (**c**) Moore neighbors

**Figure 3.** Illustration of the three neighbor configurations in a 2D cellular automaton.

Moore neighbors, encompassing eight surrounding cells, improve security by increasing complexity and ensuring that minor input changes propagate widely. This neighborhood model, ideal for image encryption, prevents discernible patterns by efficiently mixing pixel values, enhancing resistance against differential cryptanalysis. Periodic-

boundary conditions further ensure the uniform treatment of the entire image by avoiding edge effects, while hexagonal-boundary configurations, with irregular neighborhood structures, add complexity that reduces linear patterns, strengthening the resistance to pattern-based attacks.

Together, periodic-boundary, hexagonal-boundary, and Moore neighbors create a robust cryptographic structure within the 2D CA. Periodic-boundary neighbors eliminate edge effects, hexagonal-boundary neighbors introduce complexity, and Moore neighbors ensure broad diffusion. By balancing these attributes—complexity, randomness, and diffusion—the combined approach delivers highly secure encryption, making decryption without a key exceedingly difficult. Illustrative patterns generated via each configuration, shown in Figure 4, demonstrate their unique dynamics and cryptographic potential, particularly in image encryption applications.



(**a**) Periodic-boundary neighbors (**b**) Hexagonal-boundary neighbors (**c**) Moore neighbors

**Figure 4.** Illustrative patterns generated via the three neighbor configurations in a 2D cellular automaton.

A 2D CA is represented as an arrangement of $m \times n$ cells in a matrix form, where each cell can be in one of two states: '0' or '1'. A configuration is defined as the allocation of states to each cell in the 2D CA. The next configuration is determined by a local CA transformation rule, which specifies how the state of a cell at time $t+1$ depends on the state of the cell and its neighbors at time $t$. Mathematically, the state of the cell at position $(i, j)$ at time $t+1$ can be defined as

$$\begin{aligned} S_{i,j}^{t+1} &= w_1 \times S_{i,j}^t + w_2 \times S_{i+1,j}^t + w_4 \times S_{i+1,j-1}^t + w_8 \times S_{i,j-1}^t + w_{16} \times S_{i-1,j-1}^t \\ &\quad + w_{32} \times S_{i-1,j}^t + w_{64} \times S_{i-1,j+1}^t + w_{128} \times S_{i,j+1}^t + w_{256} \times S_{i+1,j+1}^t, \\ &= F\{S_{i,j}^t, S_{i+1,j}^t, S_{i+1,j-1}^t, S_{i,j-1}^t, S_{i-1,j-1}^t, S_{i-1,j}^t, S_{i-1,j+1}^t, S_{i,j+1}^t, S_{i+1,j+1}^t\} \end{aligned} \tag{7}$$

where $w_i \in \{0, 1\}$. This equation implies that, for a two-state 2D CA, any value other than '0' or '1' will not be considered. Here, $S_{i,j}^{t+1}$ represents the state of the cell at position $(i, j)$ at time $t+1$, and $S_{i,j}^t$ represents the state of the cell at time $t$.

A $k$-th-order 2D-MCA indicates that the configuration at generation $t$, denoted as $C_t$, is determined by the configurations from the previous $k$ generations, specifically $C_{t-1}, C_{t-2}, \ldots, C_{t-k}$. Analogous to a $k$-th-order 1D-MCA, the $k$-th-order 2D-MCA is formally expressed in (3). Consequently, the reversible $k$-th-order 2D-MCA is characterized by (4). The primary distinction between 2D-MCA and 1D-MCA lies in the transition rules $F_i$: in 2D-MCA, it is essential to account for the specific neighbor configuration when applying these rules.

### *3.2. The Sine-Exponential Map*

The sine-exponential (SE) map is a one-dimensional, nonlinear amalgamation of two well-established seed maps: the sine map and the exponential map [23]. Mathematically, it is expressed as follows:

$$x_{n+1} = \begin{cases} -\sin(\alpha(e^{-\pi x_n} - 1)), & -1 \leqslant x_n \leqslant 0, \\ \sin(\alpha(e^{\pi x_n} - 1)), & 0 < x_n \leqslant 1, \end{cases} \tag{8}$$

where $\alpha \in (0, 8]$ serves as the control parameter, and $x_0 \in (0, 1)$ is the initial condition (seed).

Figure 5a,b depict the bifurcation diagram and the Lyapunov exponent of the SE map, respectively. An analysis provided in [23] highlights the superior chaotic properties of (8), demonstrating that the SE map surpasses both of its seed maps (i.e., the sine and exponential maps) in performance.

Firstly, for $\alpha > 3.17$, the output sequences of (8) exhibit full coverage across the entire range of the seed $x_0 \in (0, 1)$. Specifically, within the interval $3.17 \leq \alpha \leq 7.82$, the SE map achieves a uniform distribution over $(0, 1)$, as shown in Figure 5a.

Secondly, the Lyapunov exponent of (8) remains consistently positive for $3.17 \leq \alpha \leq 7.82$, as depicted in Figure 5b, whereas the Lyapunov exponents of the seed maps are positive only within restricted parameter ranges.

Lastly, the Lyapunov exponent of the SE map is consistently larger than those of its corresponding seed maps.

In conclusion, considering these characteristics, the SE map not only outperforms its seed maps but also exhibits pure chaotic behavior throughout the entire range of $x_0$ when $3.17 \leq \alpha \leq 7.82$.

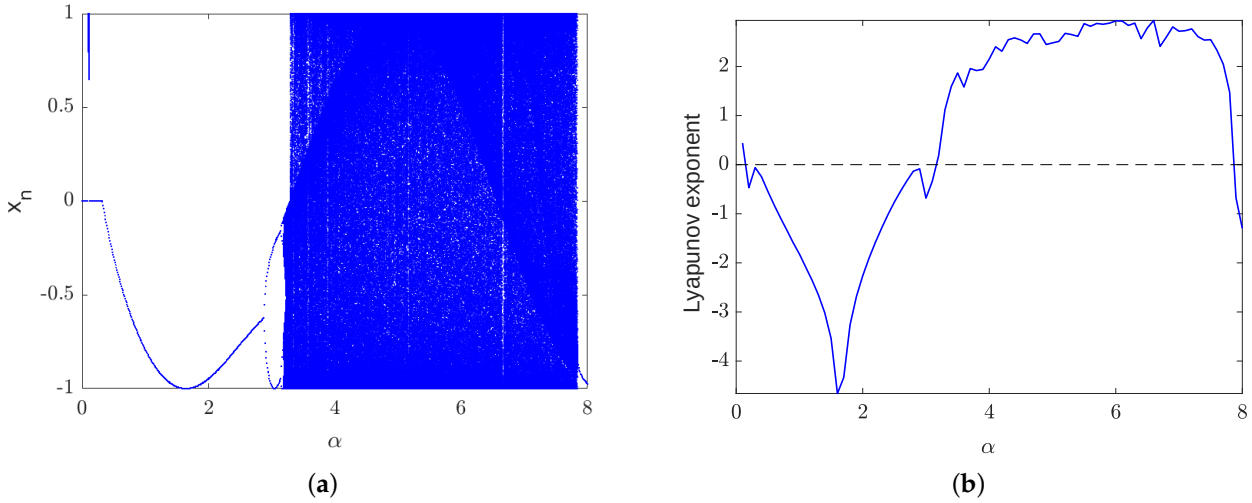



**Figure 5.** Bifurcation diagram (**a**) and Lyapunov exponent (**b**) of the SE map.

## 4. Protecting Wildlife Data: The Necessity of Encryption for Drone-Captured Images

In the realm of contemporary wildlife conservation, drones have become essential tools for the non-invasive monitoring of animal populations, behaviors, and habitats. These sophisticated unmanned aerial vehicles (UAVs) are equipped with high-resolution imaging capabilities, enabling conservationists and biologists to conduct detailed analyses of ecosystems with minimal disruption to wildlife [1–3,7,36,37]. Deployed in various conservation efforts, drones track endangered species, monitor illegal activities, and assess the impacts of climate change on biodiversity [4,7]. The data gathered through these technologies are indispensable for informed decision-making in species protection and habitat preservation [5].

However, the transmission of such sensitive data, particularly when conducted over unsecured channels, presents significant risks. Unencrypted images and metadata, which often include precise geolocation information, can be intercepted by unauthorized individuals or groups. Among the most concerning scenarios is the potential exploitation of these data by professional hunters or poachers. These actors can use the intercepted information to locate and target endangered species or vulnerable animal populations, leading to illegal hunting activities that could severely destabilize fragile ecosystems [38]. Such breaches not only threaten individual species but also jeopardize broader conservation efforts by undermining the safety and secrecy of protected wildlife areas [39].

The encryption of drone-captured images is, therefore, not merely a technical safeguard but also an essential element of wildlife protection. By encrypting data, conservationists ensure that only authorized personnel can access the information, thus preventing it from being misused for poaching or other harmful activities [40]. This layer of security is particularly crucial in remote or under-patrolled regions, where the interception of unencrypted data could lead to devastating consequences, including the loss of rare species and the collapse of local biodiversity.

In this context, encryption serves as both a defensive measure and a proactive measure, fortifying the integrity of conservation initiatives. It transforms drones from mere surveillance tools into secure, ethical instruments of environmental stewardship. By mitigating the risks associated with data interception, encryption upholds the core values of conservation, safeguarding not just the data themselves but also the wildlife and ecosystems they represent. Thus, encryption is a critical enabler of the broader goal of sustainable environmental management, ensuring that technological advancements in conservation are aligned with the ethical imperatives of wildlife protection.

Despite these advancements, an important consideration in using drones for wildlife monitoring is the impact of drone noise on animal behavior, particularly for species that rely heavily on sound for communication or navigation. Drone noise can be especially disruptive to aquatic wildlife, potentially causing stress, altering natural movement patterns, or even prompting animals to avoid the area altogether. These behavioral changes vary according to species sensitivity and the intensity of drone noise. To mitigate such effects, the deployment of quieter drones and careful flight strategies is recommended to reduce disturbances and ensure that monitoring remains as unobtrusive as possible.

## 5. Proposed Image Encryption Algorithm

In this section, we present a detailed exposition of the proposed encryption and decryption framework, which leverages a sophisticated array of cryptographic techniques, including one-dimensional and two-dimensional memory cellular automata (1D MCA and 2D MCA), hash functions, chaotic systems, and pseudorandom number generators (PRNGs). The hash function is strategically employed to produce a unique signature of the original image, serving as a critical component for ensuring system integrity and security. The resulting hash value is instrumental in deriving the system parameters and cryptographic keys required throughout the various stages of the encryption and decryption processes.

### *5.1. Key Generation*

Key generation is the process through which cryptographic keys are produced, fundamental to securing data. In this study, the key generation mechanism is intricately tied to the plain text, ensuring that each unique input yields a distinct key.

Our algorithm employs SHA-256, a specially designed cryptographic hash function that produces a 256-bit hash value.

Initially, the algorithm generates a hash value string, from which the first 212 bits are partitioned into four substrings of unequal lengths, as defined in (9)–(12). These substrings are subsequently converted into their corresponding decimal values, denoted as $h_{1-52}, h_{53-106}, h_{107-158}$, and $h_{159-212}$. These decimal values are then utilized to derive system parameters and seeds for (8), as outlined below:

$$x'_0 = \frac{1}{2}(x_0 + h_{1-52}), \tag{9}$$

$$\alpha'_0 = \frac{1}{2}(\alpha_0 + h_{53-106}), \tag{10}$$

$$x'_1 = \frac{1}{2}(x_1 + h_{107-158}), \tag{11}$$

$$\alpha_1' = \frac{1}{2}(\alpha_1 + h_{159-212}), \tag{12}$$

where $x_0, \alpha_0, x_1$, and $\alpha_1$ are the initial secret keys, and $h_{i-j}$ represents the decimal conversion of the substring spanning from the $i$-th to the $j$-th bit of the hash value.

A reversible 2D-MCA of $k$-th order necessitates $k-1$ transition rules and $k$ previous configurations to compute the subsequent configuration. In our implementation, utilizing a fourth-order 2D-MCA requires three transition rules. These rules are defined using a set of sub-keys generated via (8), with system parameters derived from (11) and (12). The detailed procedure is as follows:

1. Iterate (8) $l$ times ($l \geq 500$) to eliminate transient effects, using the control parameter $\alpha_1'$ and the seed $x_1'$. Continue iterating (8) $n/4 \times 3$ times to produce the sequence $\mathbf{e}$, where $n$ denotes the number of columns in the input image.
2. Map $\mathbf{e}$ to the interval $[0, 1]$ using (13), resulting in the sequence $\mathbf{k}$.

$$\mathbf{k} = \left(\mathbf{e} \times 10^{15}\right) \bmod 2 \tag{13}$$

3. Partition $\mathbf{k}$ into three sub-sequences, each comprising $n/4$ bits, to form the sub-key set $\mathbf{k} = [\mathbf{k}_1\ \mathbf{k}_2\ \mathbf{k}_3]$.

Similarly, a reversible 1D-MCA of the $k^{th}$ order requires $k-1$ transition rules and $k$ previous configurations to determine the next state. In our scheme, a fourth-order 1D-MCA is employed, necessitating three transition rules. These rules, denoted as $\mathbf{r}_1, \mathbf{r}_2$, and $\mathbf{r}_3$, are generated via a PRNG, with each rule consisting of 128 bits.

### *5.2. Encryption Process*

In this section, we meticulously detail the stepwise procedure of the proposed encryption scheme, as illustrated in Figure 6. The overall process consists of three distinct phases executed within a single round.

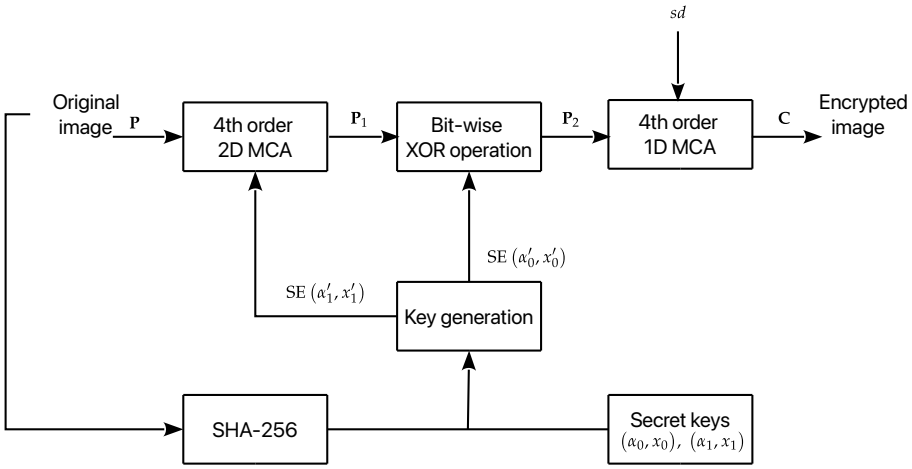



**Figure 6.** Block diagram of the encryption process of the proposed scheme.

First, a fourth-order reversible 2D-MCA is applied to the original image. Second, a chaotic matrix of dimensions $M \times N$ is generated using (8), where $M$ and $N$ correspond to the number of rows and columns of the original image, respectively. This chaotic matrix is then mapped to a suitable range and bitwise XORed with the output of the previous phase. Finally, a fourth-order reversible 1D-MCA is applied to the XORed image, yielding the encrypted image.

5.2.1. Fourth-Order 2D-MCA

The original image is first divided into four $\frac{M}{2} \times \frac{N}{2}$ blocks, denoted as $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$, and $\mathbf{B}_4$. Each block is then transformed into its binary form and subsequently converted into a $\frac{M}{2} \times (\frac{N}{2} \times 8)$ matrix. This results in binary matrices $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$, and $\mathbf{p}_4$, which serve as the initial configurations for the fourth-order 2D-MCA defined by (3). The transition rules of the 2D-MCA are derived from sub-keys $\mathbf{k}_1, \mathbf{k}_2$, and $\mathbf{k}_3$. The sequential execution of neighbor configurations follows this order: Moore, hexagonal, periodic, and then hexagonal again. After $r$ iterations of the 2D-MCA evolution mechanism, the final configurations are obtained, converted to their decimal forms, rearranged into $\frac{M}{2} \times \frac{N}{2}$ matrices, and merged to produce the image $\mathbf{P}_1$. The fourth-order 2D-MCA mechanism is expressed as

$$\mathbf{P}_1 = \mathcal{F}_{\alpha_1', x_1'}(\mathbf{P}). \tag{14}$$

5.2.2. Bitwise Chaotic Diffusion

The resulting image from the 2D-MCA, $\mathbf{P}_1$, is then subjected to a bitwise XOR operation with a chaotic matrix as follows:

- Iterate (8) $M \times N + 500$ times with the control parameter $\alpha_0'$ and seed $x_0'$. By discarding the first 500 values and rearranging the remainder, a $M \times N$ chaotic matrix $\mathbf{S}$ is obtained.
- Map $\mathbf{S}$ from the interval $[0, 1]$ to the range $\{0, 1, 2, \ldots, 255\}$ as follows:

$$\mathbf{H} = \left(\mathbf{S} \times 10^{15}\right) \bmod 256. \tag{15}$$

- Perform a bitwise XOR operation between $\mathbf{H}$ and $\mathbf{P}_1$, as given by

$$\mathbf{P}_2 = \mathbf{H} \oplus \mathbf{P}_1. \tag{16}$$

The bitwise chaotic diffusion process can be succinctly represented as

$$\mathbf{P}_2 = \mathcal{G}_{\alpha_0', x_0'}(\mathbf{P}_1). \tag{17}$$

5.2.3. Fourth-Order 1D-MCA

The image $\mathbf{P}_2$ is divided into four $\frac{M}{2} \times \frac{N}{2}$ blocks, namely $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$, and $\mathbf{B}_4$. Each block is transformed into its binary form and then converted into a one-dimensional binary sequence of length $\frac{M}{2} \times \frac{N}{2} \times 8 = 2MN$. This results in binary sequences $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$, and $\mathbf{c}_4$, which are considered the initial configurations for the fourth-order 1D-MCA defined by (3). The transition rules for the 1D-MCA are derived using sub-keys $\mathbf{r}_1, \mathbf{r}_2$, and $\mathbf{r}_3$. After $r$ iterations of the 1D-MCA evolution mechanism, the final configurations are converted into their decimal forms, rearranged into $\frac{M}{2} \times \frac{N}{2}$ matrices, and merged to form the encrypted image $\mathbf{C}$. The fourth-order 1D-MCA mechanism is described as

$$\mathbf{C} = \mathcal{H}_{sd}(\mathbf{P}_2). \tag{18}$$

*5.3. Decryption Process*

As is customary, the decryption process is meticulously designed to reverse the encryption procedure, thereby restoring the original, unaltered content. For our proposed scheme, the decryption methodology is depicted in Figure 7. Given that the proposed algorithm is symmetric, ensuring the accurate recovery of the original image necessitates that the decryption system utilizes the same secret keys employed during encryption. To facilitate proper decryption, a total of six critical parameters must be transmitted to the decryption side. These parameters encompass the seeds and system parameters of the SE map (namely, $x_0, \alpha_0, x_1$, and $\alpha_1$), the PRNG seed (denoted as $sd$), and the 256-bit hash value of the original image.

Upon receiving these parameters, the receiver can regenerate the values $x'_0, \alpha'_0, x'_1$, and $\alpha'_1$ using Equations (9)–(12). The decryption process, as illustrated in Figure 7, is executed in the reverse sequence of the encryption steps. The original image is recovered through (19).

$$\mathbf{D} = \mathcal{F}^{-1}_{\alpha'_1, x'_1}\left(\overbrace{\mathcal{G}^{-1}_{\alpha'_0, x'_0}\left(\underbrace{\mathcal{H}^{-1}_{sd}(\mathbf{C})}_{\mathbf{C}_1}\right)}^{\mathbf{C}_2}\right), \tag{19}$$

where $\mathbf{D}$ is the decrypted image, and $\mathcal{F}^{-1}_{\alpha'_1, x'_1}, \mathcal{G}^{-1}_{\alpha'_0, x'_0}, \mathcal{H}^{-1}_{sd}$ represent the inverse functions corresponding to $\mathcal{F}_{\alpha'_1, x'_1}$, $\mathcal{G}_{\alpha'_0, x'_0}$, and $\mathcal{H}_{sd}$, respectively.

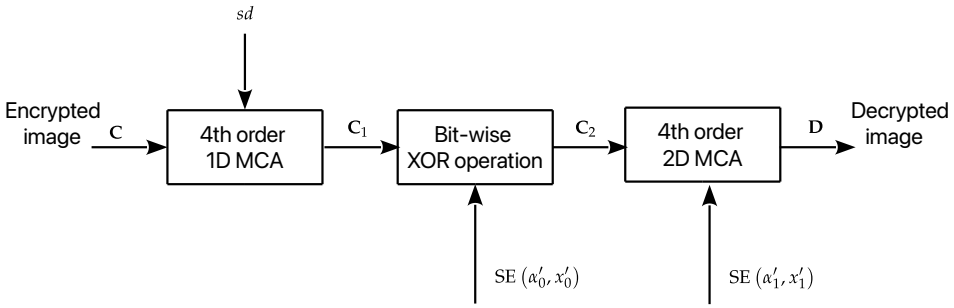



**Figure 7.** Block diagram of the decryption process of the proposed scheme.

## 6. Performance and Security Analysis

In this section, we present and discuss a comprehensive performance and security analysis to validate the efficacy of the proposed scheme. This includes keyspace analysis, statistical evaluation, sensitivity assessment, robustness testing, and complexity analysis. To benchmark the performance, we utilize a set of one hundred 8-bit grayscale images, each of size $512 \times 512$, sourced from the WAID dataset [41]. The use of this extensive dataset helps minimize experimental errors and ensures that the results obtained are reasonably accurate. A selection of results for standard images from the CVG-UGR dataset [42], including Lena, Cameraman, and Peppers, is presented to showcase the versatility of the proposed algorithm, highlighting its applicability across a broad range of images beyond wildlife imagery captured via drones. The experimental parameters are preset as follows: the seeds and control parameters $(x_0, \alpha_0)$ and $(x_1, \alpha_1)$ in Equation (8) are set to $(0.3, 7.7)$ and $(0.7, 3.9)$, respectively. Additionally, the seed for the PRNG is chosen as $sd = 100$. Figure 8a–f depict the original images of Camelus, Sheep, and Zebra, alongside their respective encrypted versions. Similarly, Figure 8a–f present the original images of Lena, Cameraman, and Peppers, together with their corresponding encrypted counterparts. Visually, the encrypted images resemble white noise and are entirely distinct from their original counterparts. The proposed decryption scheme successfully restores the images in Figures 8d–f and 9d–f, as illustrated in Figure 8g–i and Figure 9g–i, respectively. To further substantiate the superiority of the proposed algorithm, we compare its performance with state-of-the-art algorithms, as reported in [43–46]. Additionally, it is worth noting that the proposed cryptosystem can be seamlessly extended to color images. By splitting a color image into three channels—R, G, and B—the proposed algorithm can be applied in a loop until all channels are encrypted. The encrypted channels are then merged to produce the final encrypted color image.

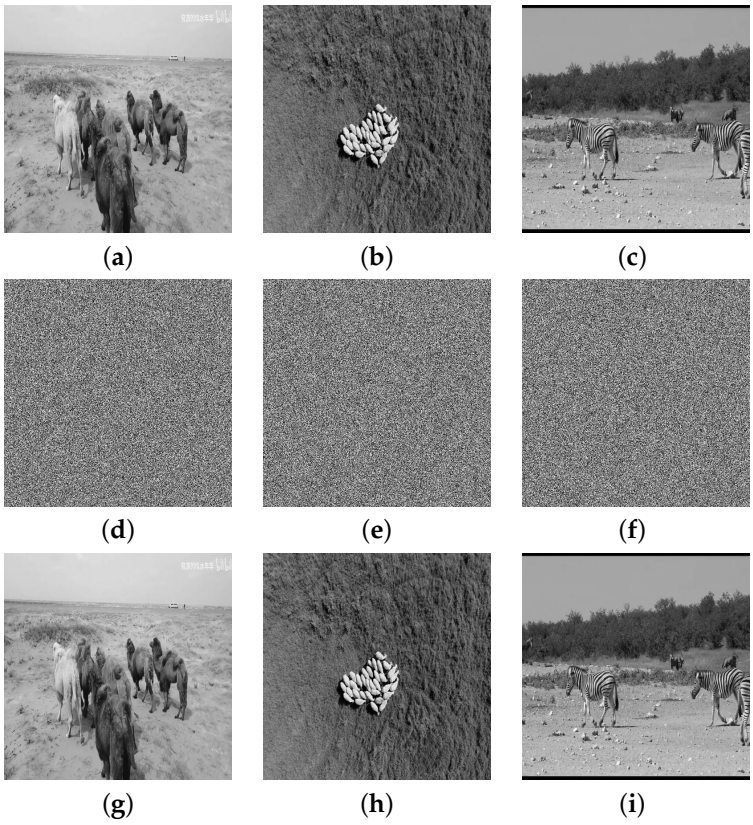


**Figure 8.** Original images of (**a**) Camelus, (**b**) Sheep, and (**c**) Zebra; their corresponding encrypted images in (**d**–**f**); and the corresponding decrypted images in (**g**–**i**) matching the encrypted versions in (**d**–**f**).

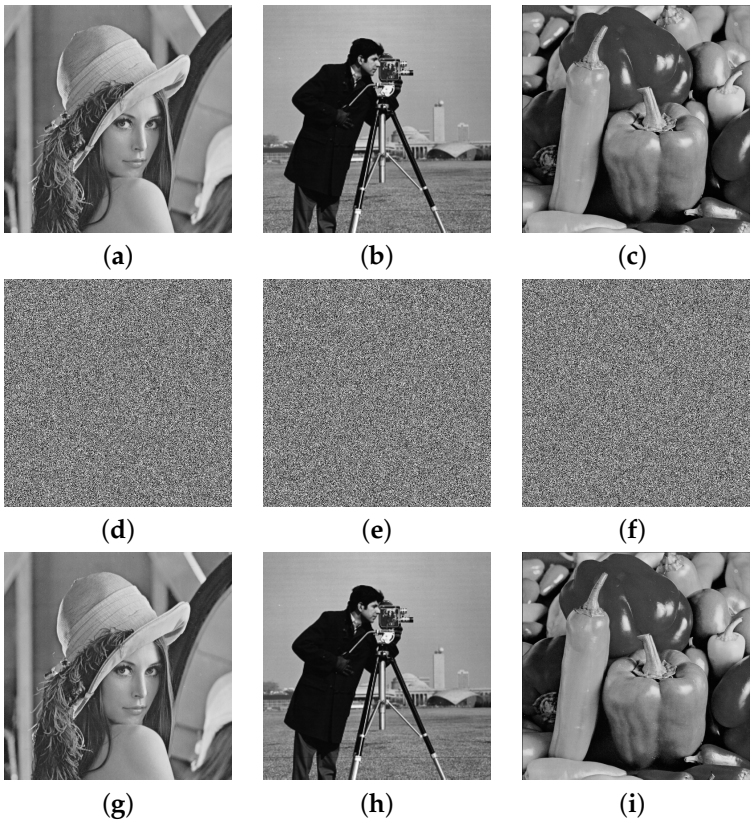


**Figure 9.** Original images of (**a**) Lena, (**b**) Cameraman, and (**c**) Peppers; their corresponding encrypted images in (**d**–**f**); and the corresponding decrypted images in (**g**–**i**) matching the encrypted versions in (**d**–**f**).

### *6.1. Implementation and Technical Requirements*

The algorithm was implemented in MATLAB 2019b on a Linux system featuring a 10th-generation Intel Core i7 processor and 16 GB of RAM. Its design prioritizes low computational complexity, enabling efficient operation within modest resource constraints. This efficiency renders the algorithm particularly adaptable to a wide array of platforms, including resource-limited environments such as drones, where computational capacity is often restricted. Consequently, the algorithm remains viable on systems with similar or even lower specifications, extending its applicability across diverse, constrained computational settings.

### *6.2. Keyspace Analysis*

It is well established that a robust cryptosystem must possess a sufficiently large keyspace to resist brute-force attacks [47,48]. The keyspace for the proposed scheme is constructed based on the following elements:

- The seeds $x_0, x_1$ and the control parameters $\alpha_0, \alpha_1$ (cf. Equation (8)).
- The seed $sd$ of the PRNG.
- A 256-bit long hash value.

Given that the computational precision of 64-bit double-precision numbers is $2^{-49}$, the possible values for $x_0$, as well as for $x_1$, $\alpha_0$, $\alpha_1$, and $sd$, exceed $2^{49}$. Furthermore, the complexity of the most effective known practical collision attack on the SHA-256 hash function is $2^{128}$. Consequently, the overall keyspace exceeds 373 bits, making it infeasible for brute-force attacks to succeed.

### *6.3. Statistical Analysis*

#### 6.3.1. Uniformity of the Bit Distribution Within Each Bit-Plane

Bit-planes are derived by converting an image into its binary representation, which visualizes each bit position within the pixels' binary forms. In a grayscale image, each pixel is represented by 8 bits, resulting in eight corresponding bit-planes. It has been observed that the distribution of 1 s and 0 s within the bit-planes of a pristine (i.e., unaltered) image is typically non-uniform due to the inherent shape and texture of the image [49]. Moreover, as noted in [50], in such images, the higher bit-planes are correlated, allowing an attacker to recover significant information about the original content. However, for an encrypted image, this distribution should be highly uniform, approaching 50%, due to the bit-level randomness introduced via the confusion and diffusion mechanisms [49]. Therefore, a secure encryption scheme should aim to achieve maximum uniformity in the bit distribution across all bit-planes.

Table 1 presents the mean and variance values of the percentage of 1 s in some of the original images used in our experiments, as well as in their encrypted counterparts. As shown in the table, the percentage of 1 s in each bit-plane of an encrypted image consistently approaches the optimal value of 50%, indicating a high level of uniformity across all bit-planes. Consequently, it can be concluded that no meaningful information can be extracted by analyzing these bit-planes.

**Table 1.** Percentage of '1 s (mean and variance) in original (O) and encrypted (E) images processed via the proposed algorithm.

| | | **8th Bit** | **7th Bit** | **6th Bit** | **5th Bit** | **4th Bit** | **3rd Bit** | **2nd Bit** | **1st Bit** |
|---|---|---|---|---|---|---|---|---|---|
| Mean | O | 47.3471 | 52.6529 | 53.6961 | 46.3039 | 50.7238 | 49.2762 | 49.9940 | 50.0060 |
| | E | 50.0024 | 49.9967 | 49.9832 | 49.9939 | 49.9872 | 50.0024 | 50.0088 | 50.0099 |
| Variance | O | 971.6803 | 971.6803 | 654.6106 | 654.6106 | 311.8808 | 311.8808 | 134.7241 | 134.7241 |
| | E | 0.0120 | 0.0088 | 0.0133 | 0.0076 | 0.0086 | 0.0095 | 0.0093 | 0.0106 |

6.3.2. Correlation of Adjacent Pixels

It is well established that pixels in pristine (or original) images exhibit a strong correlation with their neighboring pixels [48]. In contrast, this correlation should be significantly reduced (ideally approaching zero) in encrypted images, making the encrypted data unpredictable and distinctly different from the original image. Some statistical attacks exploit the correlation among adjacent pixels in encrypted images to retrieve information from the original version. Therefore, reducing the correlation between adjacent pixels is crucial for thwarting such attacks.

Mathematically, the correlation $C_{\alpha\beta}$ between adjacent pixels can be computed using the following equation:

$$C_{\alpha\beta} = \frac{\sum_{i=1}^{N}(\alpha_i - \mathbb{E}\{\{\alpha\})(\beta_i - \mathbb{E}\{\beta\})}{\sqrt{\sum_{i=1}^{N}(\alpha_i - \mathbb{E}\{\alpha\})^2}\sqrt{\sum_{i=1}^{N}(\beta_i - \mathbb{E}\{\beta\})^2}}, \tag{20}$$

where $\alpha_i$ and $\beta_i$ represent the values of two selected neighboring pixels, $N$ is the total number of pixel pairs, and $\mathbb{E}$. denotes the expectation operator.

Table 2 presents a comparison of the absolute values of correlation coefficients (in horizontal, vertical, and diagonal orientations) for 3000 randomly selected pairs of adjacent pixels from the original images and their corresponding encrypted versions using the proposed algorithm. The results show that the correlation coefficients for the original images are close to 1 in all directions, indicating a strong correlation, while the coefficients for the encrypted images are close to zero, indicating that the adjacent pixels in the encrypted images are uncorrelated.

Furthermore, Figure 10a–c and Figure 10d–f illustrate these results for the horizontal, vertical, and diagonal correlation coefficients of the Camelus image and its encrypted version, respectively.

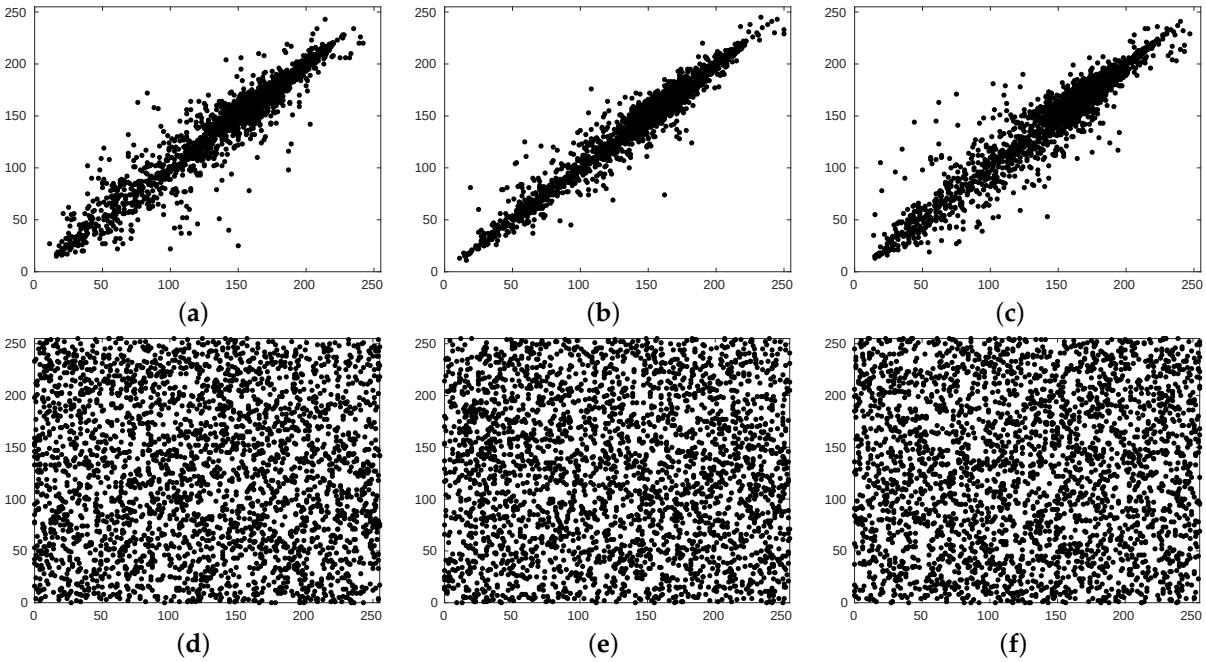



**Figure 10.** Distribution of adjacent pixel pairs in the original and encrypted images of Camelus. The distributions of two adjacent pixels in the original image are shown for the horizontal (**a**), vertical (**b**), and diagonal (**c**) directions. Similarly, the distributions of two adjacent pixels in the encrypted image are depicted for the horizontal (**d**), vertical (**e**), and diagonal (**f**) directions.

**Table 2.** Mean of the absolute correlation values between pixel pairs in the original and encrypted images.

| | Scanning Direction | Original Images | Encrypted Images ($\times 10^{-3}$) |
|---|---|---|---|
| Mean | Horizontal | 0.9480 | 0.0021 |
| | Vertical | 0.9513 | 0.0022 |
| | Diagonal | 0.9150 | 0.0022 |

6.3.3. Histogram and Chi-Square Test

The histogram of an image provides insights into the distribution of pixel intensities within that image. It visually represents the image's contents by plotting pixel intensities against the number of pixels with those intensities. Consequently, an original image typically exhibits a non-uniform histogram due to its unique shape and texture variations [48]. In contrast, the histogram of an encrypted image should be highly uniform to prevent the extraction of any meaningful information.

Figure 11 displays the Camelus and Zebra images along with their respective histograms, while Figure 12 presents the encrypted versions of these images alongside their corresponding histograms. Similarly, Figure 13 shows the Lena and Cameraman images and their histograms, whereas Figure 14 displays the encrypted versions of these images with their associated histograms. As observed, the histograms of the encrypted images are uniformly distributed.

The Chi-square test is employed to assess the conformity of these histograms (i.e., for the encrypted images) to a uniform distribution. The Chi-square statistic is formally defined as

$$\chi^2 = \sum_{i=0}^{L_p-1} \frac{(o_i - e_i)^2}{e_i}, \tag{21}$$

where $L_p$ represents the number of pixel levels in the image (e.g., 256 for grayscale images), $o_i$ denotes the observed frequency of the $i$th pixel level in the histogram of the encrypted image, and $e_i$ represents the expected frequency for a uniform distribution, calculated as $e_i = \frac{M \times N}{L_p}$.

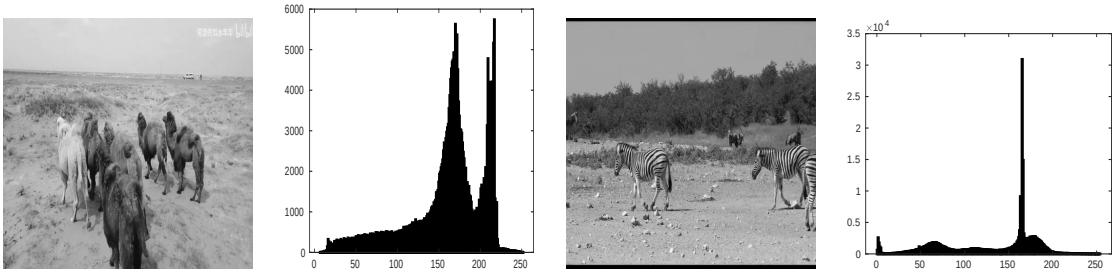



**Figure 11.** Original images of Camelus (**a**) and Zebra (**c**), along with their respective histograms (**b**,**d**).

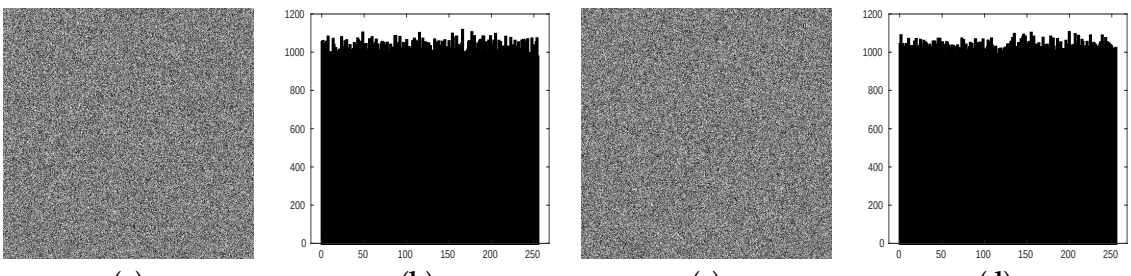



**Figure 12.** Encrypted images of Camelus (**a**) and Zebra (**c**), along with their corresponding histograms (**b**,**d**).

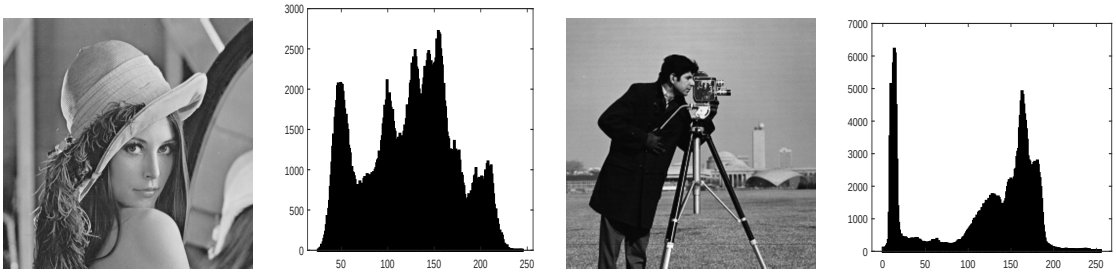



**Figure 13.** Original images of Lena (**a**) and Cameraman (**c**), along with their respective histograms (**b**,**d**).

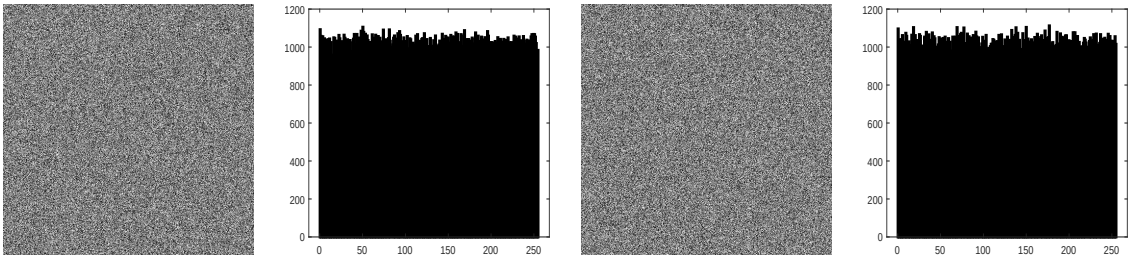



**Figure 14.** Encrypted images of Lena (**a**) and Cameraman (**c**), along with their corresponding histograms (**b**,**d**).

The Chi-square test is considered successful when the $p$-value exceeds the significance level $l_s \in (0, 1)$, indicating that the histogram distribution is more uniform.

Table 3 presents a comparison of the Chi-square test results (mean, variance, and success rate) for the evaluated algorithms, based on a preset significance level of 0.05. From this table, it can be inferred that the proposed algorithm, along with the one from [43], achieves a success rate of 98%, outperforming the other competing methods. These results demonstrate the strong resistance of our proposed scheme against histogram-based attacks.

**Table 3.** Chi-square test results for the histograms, showing the mean, variance, and success rate for the compared algorithms.

| | $\chi^2$ **Test (*p*-Value)** | | | | |
|---|---|---|---|---|---|
| | **[44]** | **[45]** | **[43]** | **[46]** | **Proposed** |
| Mean | 0.4887 | 0.5432 | 0.4973 | 0.4544 | 0.5206 |
| Variance | 0.0781 | 0.0849 | 0.0720 | 0.0907 | 0.0823 |
| Success rate (%) | 95 | 96 | 98 | 95 | 98 |

#### 6.3.4. Global Entropy

Entropy, originally introduced by Shannon [51], serves as a fundamental measure of randomness within a random variable [46]. The global entropy (expressed in bits) of a source $X$ is mathematically defined as

$$H(X) = -\sum_{i=1}^{L} p(x_i) \log_2(p(x_i)), \tag{22}$$

where $L$ represents the number of distinct symbols in $X$, and $p(x_i)$ denotes the probability of occurrence of each symbol, $x_i$.

A higher entropy value signifies a greater level of uncertainty or randomness in the data. For grayscale images, the maximum achievable entropy is 8 bits. Therefore, a robust

encryption scheme should inject sufficient randomness into the image, driving its entropy value close to this theoretical maximum of 8.

Table 4 presents a comparison of the global entropy (mean and variance) between the proposed scheme and other algorithms from the literature. As observed, the entropy values for all compared algorithms are nearly optimal. Notably, the proposed algorithm, along with those in [43–45], demonstrates slightly higher entropy than the algorithm in [46]. This finding suggests that the proposed method effectively mitigates information leakage, ensuring a high degree of security.

**Table 4.** Global entropy (mean and variance) outcomes.

| | **Global Entropy** | | | | |
|---|---|---|---|---|---|
| | **[44]** | **[45]** | **[43]** | **[46]** | **Proposed** |
| Mean | 7.9993 | 7.9993 | 7.9993 | 7.9908 | 7.9993 |
| Variance ($\times 10^{-9}$) | 3.7935 | 3.8254 | 3.1727 | $0.0072 \times 10^{9}$ | 3.8729 |

6.3.5. Local Entropy

While global entropy is a valuable metric for assessing overall randomness, it often fails to capture deficiencies in randomness within localized regions of an image. This limitation can be particularly problematic when evaluating the security of encryption schemes, as patterns or predictable structures may still exist in specific areas of the image. To address this issue, a more granular measure known as local entropy was introduced in [52]. Local entropy provides a deeper understanding of randomness by focusing on specific regions within the image, thus offering a more comprehensive evaluation of the encryption scheme's effectiveness.

Mathematically, the local entropy can be expressed as

$$H_{k,TB}(\mathbf{S}) = \frac{1}{k}\sum_{i=1}^{k} H(\mathbf{S}_i), \tag{23}$$

where the following applies:

- $k$ is the number of non-overlapping image blocks.
- $TB$ is the number of pixels within each image block.
- $H(\mathbf{S}_i)$ is the Shannon entropy of the image block $\mathbf{S}_i$.

Table 5 presents the results of the local entropy test, comparing the performance of the proposed algorithm against those reported in [43–46]. The findings indicate that the proposed scheme performs exceptionally well in maintaining high local entropy. Additionally, the algorithms from [43–45] also demonstrate strong local entropy performance, with all approaches achieving values close to the optimal entropy of 7.9024693 as identified in [52].

These results underscore the effectiveness of the proposed encryption scheme not only in achieving high global entropy but also in ensuring that localized regions within the image exhibit significant randomness. This characteristic is crucial for enhancing the security of the encrypted image, as it minimizes the risk of localized attacks that could exploit any residual patterns. Overall, the proposed scheme demonstrates a robust capability in both global and local entropy measures, providing a strong defense against potential vulnerabilities.

**Table 5.** Local entropy (mean and variance) outcomes.

| | **Local Entropy** | | | | |
|---|---|---|---|---|---|
| | **[44]** | **[45]** | **[43]** | **[46]** | **Proposed** |
| Mean | 7.9025 | 7.9025 | 7.9024 | 7.8924 | 7.9025 |
| Variance ($\times 10^{-7}$) | 2.7439 | 3.8913 | 3.5318 | $0.0097 \times 10^{7}$ | 4.1654 |

#### 6.3.6. Randomness of Cipher-Images

For an encrypted image to effectively resist statistical attacks, the pixel values must be uniformly distributed. To assess the robustness of the proposed encryption scheme against such attacks, we employed the statistical test suite developed by NIST [53]. This suite consists of 15 tests that examine various aspects of randomness in long binary sequences, aiming to identify any non-random patterns within the sequences. Each test is designed to evaluate a specific null hypothesis ($H_0$), which posits that the sequence being tested is random.

The probability that a test fails to reject the null hypothesis when the sequence is genuinely random is referred to as the level of significance. For cryptographic applications, a significance level of 1%, also known as the decision threshold, is recommended in [53].

In our experiments, we used a significance level of $\alpha = 1\%$ and applied the NIST test suite to two image datasets. The first dataset (Dataset #1) consists of 100 8-bit grayscale images, each with a resolution of $512 \times 512$, sourced from the WAID dataset [41]. The second dataset (Dataset #2) also contains 100 8-bit grayscale images of the same size, drawn from the CVG-UGR dataset [42].

The evaluation process was conducted in three steps. First, both datasets were encrypted using the proposed cryptosystem with the same key. Next, each encrypted image was converted into a binary sequence of 2,097,152 bits. Finally, each group of 100 binary sequences underwent the 15 NIST tests.

The results of the NIST tests for both datasets are summarized in Table 6. As the table shows, all datasets successfully passed all 15 tests, demonstrating that the proposed cryptosystem generates highly random encrypted images. This high level of randomness significantly strengthens the scheme's resilience against statistical attacks, further validating its effectiveness in securing image data.

**Table 6.** Results of the NIST SP 800-22 test suite for two image datasets, each containing 100 images encrypted using the proposed cryptosystem.

| | Dataset #1 | | | Dataset #2 | | |
|---|---|---|---|---|---|---|
| **Test** | ***p*-Value** | **Success Rate** [(1)] | **Decision** [(2)] | ***p*-Value** | **Success Rate** [(3)] | **Decision** [(2)] |
| Frequency (Monobit) | 0.1206 | 0.9952 | Success | 0.0324 | 0.9952 | Success |
| Block Frequency | 0.7647 | 0.9904 | Success | 0.8210 | 0.9809 | Success |
| Cumulative Sums (1) | 0.1996 | 0.9952 | Success | 0.7248 | 1.0000 | Success |
| Cumulative Sums (2) | 0.3153 | 0.9856 | Success | 0.4895 | 0.9952 | Success |
| Runs | 0.6527 | 1.0000 | Success | 0.1516 | 0.9856 | Success |
| Longest Run of Ones in a Block | 0.8869 | 0.9856 | Success | 0.4993 | 0.9856 | Success |
| Binary Matrix Rank | 0.9329 | 0.9904 | Success | 0.0224 | 0.9856 | Success |
| Discrete Fourier Transform (Spectral) | 0.4235 | 0.9904 | Success | 0.4702 | 0.9809 | Success |
| Non-overlapping Template Matching [(4)] | 0.0479 | 0.9713 | Success | 0.8120 | 0.9569 | Success |
| Overlapping Template Matching | 0.0746 | 1.0000 | Success | 0.5904 | 0.9761 | Success |
| Maurer's Universal Statistical Test | 0.1010 | 0.9952 | Success | 0.6319 | 0.9904 | Success |
| Approximate Entropy | 0.2656 | 0.9856 | Success | 0.9329 | 0.9856 | Success |
| Random Excursions [(4)] | 0.9915 | 0.9667 | Success | 0.1253 | 0.9764 | Success |
| Random Excursions Variant [(4)] | 0.8486 | 0.9833 | Success | 0.1379 | 0.9764 | Success |
| Serial (1) | 0.8120 | 0.9904 | Success | 0.2656 | 0.9856 | Success |
| Serial (2) | 0.3385 | 0.9856 | Success | 0.2338 | 0.9904 | Success |
| Linear Complexity | 0.7147 | 1.0000 | Success | 0.4702 | 0.9952 | Success |

[(1)] The minimum pass rate for each statistical test, except for the random excursion (variant) test, is around 0.9665. For the random excursion (variant) test, the minimum pass rate is approximately 0.9583. [(2)] Decision made at the 1% significance level. [(3)] The minimum pass rate for all statistical tests, excluding the random excursion (variant) test, is approximately 0.9665. For the random excursion (variant) test, the minimum pass rate is around 0.9606. [(4)] The lowest *p*-value and the success rate are taken into account.

### *6.4. Sensitivity Analysis*

#### 6.4.1. Key Sensitivity

Key sensitivity is a critical factor in evaluating the security of an encryption scheme [49]. It ensures that even the slightest modification in the secret key results in significant changes to both the encrypted and decrypted images. This property is vital for preventing potential attackers from successfully decrypting the image with a similar, but incorrect, key.

To assess key sensitivity, we examine the relationship between the mean Hamming distances and the number of altered bits in the secret key, as defined in (8). Figure 15a presents this analysis using a boxplot. The indices of the $n$ $(n = 1, \ldots, 10)$ altered bits, along with the images used in the study, are randomly generated over 200 iterations. The results indicate that, on average, 99.3% of the observed Hamming distances fall within the range of $[0.4991, 0.5009]$, which is remarkably close to the ideal value of 0.5.

This near-optimal performance shows that the proposed encryption scheme is highly sensitive to key variations, even with a single-bit change. This ensures that any minor key alteration results in a completely different encrypted output, greatly enhancing the scheme's security and robustness against decryption attempts with incorrect keys.

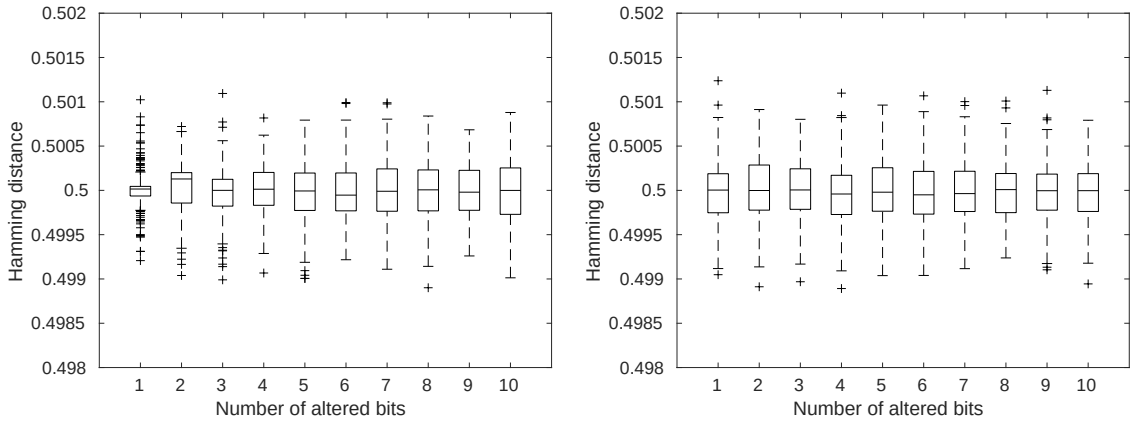



**Figure 15.** Boxplots depict key sensitivity (**a**) and plaintext sensitivity (**b**) using parameters $sd = 100$, $x_0 = 0.1435546875$, $\alpha_0 = 2.0595703125$, $x_1 = 0.41650390625$, and $\alpha_1 = 3.03076171875$.

#### 6.4.2. Plaintext Sensitivity

For optimal security, a cryptosystem must exhibit extreme sensitivity to even minute alterations in the original image. Specifically, modifying a mere single bit in the encrypted image should preclude the recovery of the original image [47]. Cryptographic protocols with robust plaintext sensitivity prove to be highly resistant to chosen plaintext attacks, where an adversary attempts to deduce the original image by analyzing variations in the encrypted output.

Figure 15b displays boxplots of the average Hamming distances as a function of the number of bits altered in the original image. During this experiment, both the images and the indices of the $n$ $(n = 1, \ldots, 10)$ changed bits were randomly selected across 200 iterations. A detailed examination of Figure 15b shows that, on average, 99.3% of the Hamming distances range between $[0.4990, 0.5010]$, closely approximating the ideal midpoint of 0.5.

These findings affirm that the algorithm under consideration is profoundly sensitive to even minor changes in the original image, ensuring that no meaningful information is extractable from the encrypted data. Such heightened sensitivity significantly enhances the security of the cryptosystem against various attacks.

6.4.3. UACI and NPCR

Differential attacks, a subset of chosen plaintext or known plaintext attacks, involve a cryptanalyst introducing slight modifications to an original image to observe resultant changes in its encrypted counterpart. This process can potentially expose correlations between the original and encrypted images, thereby compromising the secrecy of the key used for decryption. To evaluate a cryptosystem's resilience against such differential attacks, quantitative measures like the unified average changing intensity (UACI) and the number of changing pixels rate (NPCR) are employed [49]. Higher NPCR and UACI values signify a stronger defense against differential attacks. The NPCR and UACI metrics for two encrypted images, $\mathbf{E}_1$ and $\mathbf{E}_2$, each modified by altering one pixel in their original images, are calculated using (24) and (25), respectively.

$$\texttt{NPCR} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\mathbf{D}(i,j), \tag{24}$$

$$\texttt{UACI} = \frac{1}{255 \times MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|\mathbf{E}_1(i,j) - \mathbf{E}_2(i,j)|, \tag{25}$$

where $\mathbf{D}(i,j)$ denotes the difference between $\mathbf{E}_1$ and $\mathbf{E}_2$ as follows:

$$\mathbf{D}(i,j) = \begin{cases} 0, & \text{if } \mathbf{E}_1(i,j) = \mathbf{E}_2(i,j) \\ 1, & \text{if } \mathbf{E}_1(i,j) \neq \mathbf{E}_2(i,j) \end{cases}$$

Tables 7 and 8 present the NPCR and UACI scores for the proposed algorithm, comparing these results with those from other studies [43–46]. The NPCR and UACI scores for the proposed algorithm closely approximate the ideal values of 99.6% and 33.4%, respectively [47]. This performance aligns with the outcomes reported for competing methods [43–45], substantiating the robustness of the proposed scheme against differential attacks.

**Table 7.** NPCR (mean and variance) for encrypted images subjected to a one-bit alteration in the original image.

| | **NPCR (%)** | | | | |
|---|---|---|---|---|---|
| | **[44]** | **[45]** | **[43]** | **[46]** | **Proposed** |
| Mean | 99.6081 | 99.6083 | 99.6072 | 99.6106 | 99.6094 |
| Variance | 0.0002 | 0.0001 | 0.0001 | 0.0001 | 0.0001 |

**Table 8.** UACI (mean and variance) for encrypted images subjected to a one-bit alteration in the original image.

| | **UACI (%)** | | | | |
|---|---|---|---|---|---|
| | **[44]** | **[45]** | **[43]** | **[54]** | **Proposed** |
| Mean | 33.4639 | 33.4714 | 33.4597 | 33.4262 | 33.4697 |
| Variance | 0.0027 | 0.0025 | 0.0022 | 0.1218 | 0.0020 |

*6.5. Known Plaintext and Chosen Plaintext Attacks*

Known plaintext attacks (KPAs) occur when a cryptanalyst has access to both the original and encrypted images, enabling them to derive secret information and compromise the cipher [55]. Conversely, in chosen plaintext attacks (CPAs), the cryptanalyst selects specific images for encryption and studies their encrypted forms, aiming to decipher or estimate decryption methods without the key. In the method we propose, the encryption key is derived from the input image itself, resulting in a unique, one-time key for each image. Thus, our algorithm robustly defends against both known and chosen plaintext attacks.

Additionally, it is well documented that cryptanalysts often prefer to encrypt monochromatic images, such as all-white or all-black, to bypass the permutation and/or substitution phases of an encryption algorithm. Figure 16 illustrates the encrypted monochromatic images and their histograms, while Figure 17 displays the correlation among neighboring pixel pairs in various directions. These figures indicate uniformly distributed histograms and a random distribution of pixel pairs across the phase plane.

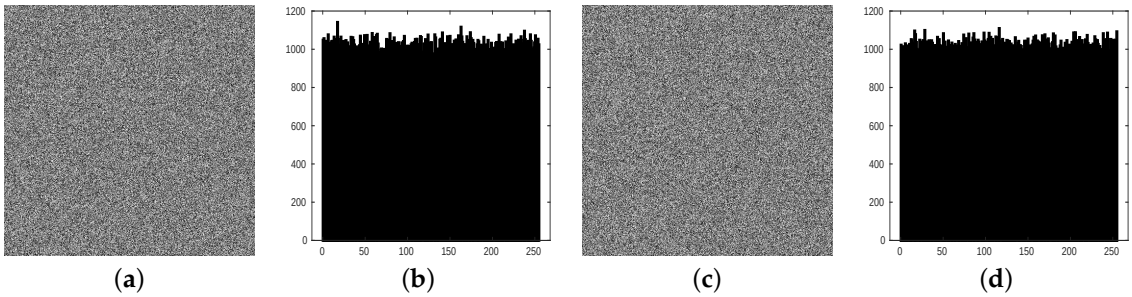



**Figure 16.** Encrypted representations of the (**a**) all-white and (**c**) all-black images, along with their respective histograms (**b**,**d**).

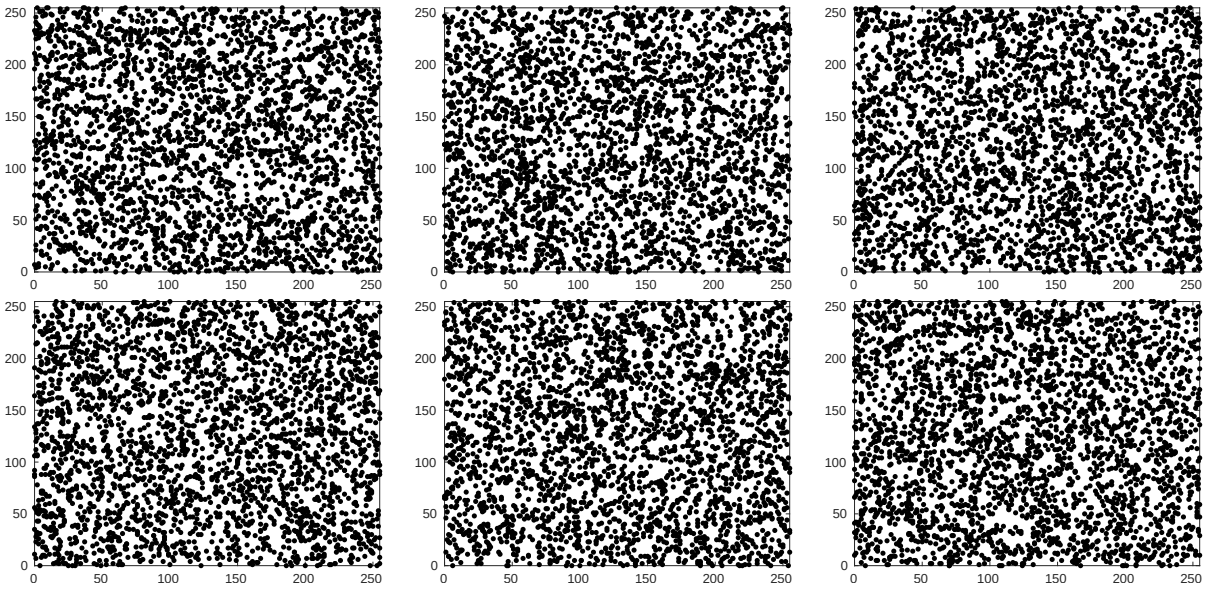


**Figure 17.** Correlation among adjacent pixel pairs in horizontal, vertical, and diagonal orientations for both all-white and all-black images, arranged in row-major sequence.

Quantitative assessments of this behavior are detailed in Tables 9 and 10, covering various statistical and differential tests on the monochromatic images. These tables also compare our method against others documented in [43–46]. The statistical tests confirm the randomness, uniform histograms, and negligible correlation among adjacent pixels as produced via our method. Notably, both global and local entropies approach optimal values of 8 and 7.9024693, respectively. Sensitivity tests, including NPCR and UACI, yield results close to the ideal scores of 99.6% and 33.4%, respectively, aligning with or surpassing the performance of the referenced schemes [43–46]. Particularly, our algorithm excels over the algorithms of [44,45], which struggle with encrypting an all-black image, and equates the performance of [43,46] in resisting both known and chosen-plaintext attacks on monochromatic images. This analysis underscores another notable achievement of our proposed cryptographic approach.

**Table 9.** Statistical evaluation of the encrypted all-white and all-black images.

| Image | Algorithm | $\chi^2$ Test of Histogram | Correlation | | | Global Entropy | Local Entropy |
|---|---|---|---|---|---|---|---|
| | | *p*-Value | Horizontal | Vertical | Diagonal | | |
| All-white | [44] | 0.1511 | −0.0008 | −0.0004 | 0.0014 | 7.9992 | 7.9026 |
| | [45] | 0.7908 | −0.0050 | 0.0006 | 0.0016 | 7.9993 | 7.9008 |
| | [43] | 0.2275 | 0.0032 | 0.0003 | 0.0029 | 7.9993 | 7.9022 |
| | [46] | 0.7645 | −0.0014 | −0.0010 | −0.0018 | 7.9993 | 7.9028 |
| | Proposed | 0.7219 | 0.0029 | −0.0026 | −0.0004 | 7.9993 | 7.9019 |
| All-black | [44] | 0 | 0.0055 | 0.0022 | −0.0035 | 1.0000 | 0.9996 |
| | [45] | 0 | 0.0043 | −0.0036 | −0.0040 | 1.0000 | 0.9996 |
| | [43] | 0.7901 | −0.0027 | 0.0026 | −0.0006 | 7.9993 | 7.9026 |
| | [46] | 0.5702 | 0.0004 | 0.0006 | −0.0002 | 7.9993 | 7.9023 |
| | Proposed | 0.9386 | −0.0038 | −0.0013 | −0.0041 | 7.9994 | 7.9034 |

**Table 10.** Differential analysis of the encrypted all-white and all-black images.

| Image | Algorithm | NPCR (%) | UACI (%) |
|---|---|---|---|
| All-white | [44] | 99.5758 | 33.5269 |
| | [45] | 99.6132 | 33.4725 |
| | [43] | 99.6082 | 33.4887 |
| | [46] | 99.6037 | 33.4982 |
| | Proposed | 99.6078 | 33.4594 |
| All-black | [44] | 49.9306 | 0.1958 |
| | [45] | 49.9172 | 0.1958 |
| | [43] | 99.5922 | 33.4019 |
| | [46] | 99.6117 | 33.4495 |
| | Proposed | 99.6143 | 33.4417 |

### *6.6. Time Complexity Analysis*

The computational time complexities and their corresponding orders of magnitude for a $512 \times 512$ grayscale image are detailed in Table 11. Notably, the proposed algorithm demonstrates the lowest computational complexity order. Additionally, it outperforms the other algorithms in terms of the execution speed. A comprehensive analysis of the computational characteristics of the proposed algorithm is presented in Appendix A, whereas the complexities of the comparative algorithms are thoroughly discussed in [46].

**Table 11.** Time complexity orders and their corresponding magnitudes for a grayscale image of dimensions $M \times N = 512 \times 512$.

| | Complexity Order | Order of Magnitude ($\times 10^6$) |
|---|---|---|
| [44] | $O(212MN + 4M\log(M) + 4N\log(N))$ | 55 |
| [45] | $O(72MN + 116L^4 + 224L^2 + 32L^2\log(L))$ | 49 |
| [43] | $O(54MN + 20M)$ | 14 |
| [46] | $O(61MN + 3 \times (M + N))$ | 15 |
| Proposed | $O(45.25MN + 6.75N)$ | 11 |

This analysis is crucial in evaluating the overall performance of a cryptosystem, as it directly impacts the system's efficiency and practicality in real-world applications. A lower computational complexity translates to faster encryption and decryption processes, making the cryptosystem more viable for large-scale or time-sensitive applications. By demonstrating both robust security features and efficient computational performance, the

proposed algorithm not only ensures the protection of sensitive data but also maintains the usability and scalability essential for practical deployment.

## 7. Security and Performance Validation

The integration of 2D MCA and 1D MCA addresses diffusion and confusion from complementary perspectives. The 2D MCA leverages spatial relationships across a two-dimensional grid, which broadens diffusion by spreading pixel values throughout the encrypted image. This spatial arrangement ensures that changes in one pixel affect multiple surrounding pixels, thereby obscuring the original image structure and strengthening the encryption's resistance to cryptanalysis. Additionally, varying neighbor configurations in the 2D MCA provide diverse pixel interactions, making it harder for an attacker to detect patterns and predict key elements of the encryption.

The 1D MCA complements this approach by introducing sequence complexity and non-linear transformations, enhancing confusion. This one-dimensional sequence creates a layer of non-linear operations that obfuscates pixel relationships in a manner that increases the scheme's resistance to analysis attacks. As the 1D MCA operates in a sequential, line-by-line approach, it also maintains a relatively low computational overhead, ensuring that the encryption remains efficient even with added complexity.

We performed extensive security and performance analyses to rigorously validate the efficacy of this approach, encompassing the following tests:

1. Keyspace analysis: The combination of 1D and 2D MCA significantly expands the keyspace by incorporating diverse spatial and sequential transformations. The multi-dimensional approach prevents brute-force attacks, as the increased complexity requires a much larger keyspace to cover all possible configurations.
2. Statistical evaluation: The use of 2D MCA with varying neighbor configurations minimizes statistical biases by distributing pixel values across multiple configurations. This results in a more uniform pixel distribution, as confirmed through tests like histogram analysis and correlation coefficients. The 1D MCA further complicates patterns, ensuring statistical characteristics that make plaintext-image patterns indistinguishable.
3. Sensitivity assessment: The layered encryption approach enhances sensitivity, meaning that minor changes to the plaintext or key produce substantial variations in the encrypted image. This was demonstrated through tests such as NPCR and UACI, which confirmed the system's strong sensitivity to both key and plaintext variations. The combination of 1D and 2D MCA ensures that any change cascades through both spatial and sequential elements, maximizing sensitivity.
4. Robustness testing: The structure provided via the 2D MCA adds resilience to attacks by spreading pixel changes widely, making the system more robust against attacks that try to analyze or disrupt pixel structures. The 1D MCA's sequential, non-linear transformations make it resilient to differential attacks, where attackers attempt to predict output patterns based on small input variations.
5. Complexity analysis: Despite the robust security features, the combination of 1D and 2D MCA maintains low computational complexity. Both components are inherently low-cost in terms of processing, and their integration does not introduce significant overhead. The algorithm was tested for efficiency, confirming its suitability for real-time applications and systems with limited computational resources, such as drones.

This combination of 1D and 2D MCA achieves a balance between high security and efficient performance, ensuring robust protection while maintaining the computational efficiency needed for practical, real-time applications.

## 8. Conclusions

This paper has introduced a novel encryption algorithm specifically designed to secure wildlife data captured via drones, which is crucial for conservation efforts. The proposed method effectively integrates 1D and 2D memory cellular automata (MCA) with a bitwise XOR operation to enhance both diffusion and confusion, providing robust protection

against cryptographic attacks. The use of varying neighbor configurations in 2D MCA, along with chaotic rules derived from the sine-exponential (SE) map, further strengthens the encryption process by diversifying and complicating the encryption patterns. Additionally, the incorporation of the SHA-256 hash of the input image to derive encryption parameters ensures unique keystreams for each image, significantly enhancing the resistance to known and chosen plaintext attacks. Performance evaluations confirm that the proposed scheme achieves a well-balanced trade-off between security and complexity, making it an efficient and resilient solution for safeguarding sensitive ecological data.

**Author Contributions:** A.B. conceived the image encryption algorithm. A.B. performed numerical experiments. A.B. and H.M. analyzed the data. A.B. wrote the original draft. H.M. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Spanish Ministry of Science and Innovation and the Research State Agency under Grant PID2021-123627OB-C55, cofinanced by FEDER funds (MCIN/AEI/FEDER/UE).

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A. An In-Depth Analysis of Time Complexity for the Proposed Image Encryption Algorithm

Table A1 enumerates the frequency of each elementary operation relative to the total number of image pixels, $MN$. The random sequences employed within the proposed algorithm are generated using a pseudorandom number generator (PRNG), with the associated complexity cost calculated by referencing the well-established linear congruential generator (LCG) as a model for uniform random number generation [56]. It is also noteworthy that the SHA-256 hash algorithm processes inputs of 512 bits with a constant number of operations. After compiling the counts of occurrences shown in Table A1 and omitting lower-order terms, the computational complexity of the proposed algorithm is determined as follows.

$$O(45.25MN + 6.75N) \tag{A1}$$

**Table A1.** Time complexity assessment of the proposed image encryption algorithm.

| | |
|---|---|
| **Addition** | $24.25MN + 0.75N + 1384$ |
| **Multiplication** | $8MN + 3.75N + 4384$ |
| **Exponentiation** | $MN + 0.75N + 1000$ |
| **Sinus function** | $MN + 0.75N + 1000$ |
| **Mod** | $4MN + 0.75N + 384$ |
| **XOR** | $7MN$ |
| **SHA-256 operations** | $\frac{8}{512}MN$ |

## References

1. Koh, L.P.; Wich, S.A. Dawn of drone ecology: Low-cost autonomous aerial vehicles for conservation. *Trop. Conserv. Sci.* **2012**, *5*, 121–132. [CrossRef]
2. Gonzalez, L.F.; Montes, G.A.; Puig, E.; Johnson, S.; Mengersen, K.; Gaston, K.J. Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation. *Sensors* **2016**, *16*, 97. [CrossRef] [PubMed]
3. Christie, K.S.; Gilbert, S.L.; Brown, C.L.; Hatfield, M.; Hanson, L. Unmanned aircraft systems in wildlife research: Current and future applications of a transformative technology. *Front. Ecol. Environ.* **2016**, *14*, 241–251. [CrossRef]
4. Pimm, S.L.; Alibhai, S.; Bergl, R.; Dehgan, A.; Giri, C.; Jewell, Z.; Joppa, L.; Kays, R.; Loarie, S. Emerging technologies to conserve biodiversity. *Trends Ecol. Evol.* **2015**, *30*, 685–696. [CrossRef] [PubMed]
5. Rees, A.F.; Avens, L.; Ballorain, K.; Bevan, E.; Broderick, A.C.; Carthy, R.R.; Christianen, M.J.; Duclos, G.; Heithaus, M.R.; Johnston, D.W.; et al. The potential of unmanned aerial systems for sea turtle research and conservation: A review and future directions. *Endanger. Species Res.* **2018**, *35*, 81–100. [CrossRef]

6. Anderson, K.; Gaston, K.J. Lightweight unmanned aerial vehicles will revolutionize spatial ecology. *Front. Ecol. Environ.* **2013**, *11*, 138–146. [CrossRef]
7. Wich, S.A.; Koh, L.P. *Conservation Drones: Mapping and Monitoring Biodiversity*; Oxford University Press: Oxford, UK, 2018. [CrossRef]
8. Pareek, N.K.; Patidar, V.; Sud, K.K. Image encryption using chaotic logistic map. *Image Vis. Comput.* **2006**, *24*, 926–934. [CrossRef]
9. Chen, G.; Mao, Y.; Chui, C.K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **2004**, *21*, 749–761. [CrossRef]
10. Kocarev, L.; Jakimoski, G. Logistic map as a block encryption algorithm. *Phys. Lett. A* **2001**, *289*, 199–206. [CrossRef]
11. Bouteghrine, B.; Tanougast, C.; Sadoudi, S. Novel image encryption algorithm based on new 3-d chaos map. *Multimed. Tools Appl.* **2021**, *80*, 25583–25605. [CrossRef]
12. Li, S.; Mou, X.; Cai, Y. Pseudo-random bit generator based on couple chaotic systems and its application in stream-cipher cryptography. In *Progress in Cryptology—INDOCRYPT 2001, Proceedings of the Second International Conference on Cryptology in India, Chennai, India, 16–20 December 2001*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2247, pp. 316–329. [CrossRef]
13. Wolfram, S. Cryptography with cellular automata. In *Advances in Cryptology*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1986; Volume 218, pp. 429–432. [CrossRef]
14. Gutowitz, H.A. Cryptography with dynamical systems. In *Cellular Automata and Cooperative Systems*; Springer: Dordrecht, The Netherlands, 1995; pp. 237–274. [CrossRef]
15. Roy, S.; Rawat, U.; Karjee, J. A lightweight cellular automata based encryption technique for IoT applications. *IEEE Access* **2019**, *7*, 39782–39793. [CrossRef]
16. Kheiri, H.; Dehghani, R. A hybrid model of recursive cellular automata, DNA sequences, and chaotic system for image encryption. *Multimed. Tools Appl.* **2024**, 1–27. [CrossRef]
17. Kumari, E.; Mukherjee, S. A Review on Encryption Techniques based on Cellular Automata. In *Artificial Intelligence and Communication Technologies*; SCRS: New Delhi, India, 2022; pp. 225–234 . [CrossRef]
18. Xu, L.; Li, Z.; Li, J.; Hua, W. A novel bit-level image encryption algorithm based on chaotic maps. *Opt. Lasers Eng.* **2016**, *78*, 17–25.. [CrossRef]
19. Kumar, A.; Raghava, N.S An efficient image encryption scheme using elementary cellular automata with novel permutation box. *Multimed. Tools Appl.* **2021**, *80*, 21727–21750. [CrossRef]
20. Zhang, Y.Q.; He, Y.; Li, P.; Wang, X.Y. A new color image encryption scheme based on 2DNLCML system and genetic operations. *Opt. Lasers Eng.* **2020**, *128*, 106040. [CrossRef]
21. Liu, W.; Sun, K.; Zhu, C. A fast image encryption algorithm based on chaotic map. *Opt. Lasers Eng.* **2016**, *84*, 26–36. [CrossRef]
22. Bhat, I.K.; Qadir, F.; Neshat, M.; Gandomi, A.H. Exploring Cellular Automata Learning: An Innovative Approach for Secure and Imperceptible Digital Image Watermarking. *IEEE Access* **2024**, *12*, 159748–159774. [CrossRef]
23. Zhu, H.; Qi, W.; Ge, J.; Liu, Y. Analyzing Devaney chaos of a sine–cosine compound function system. *Int. J. Bifurc. Chaos* **2018**, *28*, 1850176. [CrossRef]
24. Souyah, A.; Faraoun, K.M. Secure image encryption scheme using cellular automata and chaotic maps. *Nonlinear Dyn.* **2016**, *86*, 639–653. [CrossRef]
25. Jeyaram, B.; Raghavan, R. New cellular automata-based image cryptosystem and a novel non-parametric pixel randomness test. *Secur. Commun. Netw.* **2016**, *9*, 3365–3377. [CrossRef]
26. Roy, S.; Shrivastava, M.; Rawat, U.; Pandey, C.V.; Nayak, S.K. IESCA: An efficient image encryption scheme using 2-D cellular automata. *J. Inf. Secur. Appl.* **2021**, *61*, 102919. [CrossRef]
27. Wang, X.; Luan, D. A novel image encryption algorithm using chaos and reversible cellular automata. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 3075–3085. [CrossRef]
28. Habibipour, M.; Maarefdoust, R.; Yaghobi, M.; Rahati, S. An image encryption system by 2D Memorized Cellular Automata and chaos mapping. In Proceedings of the 6th International Conference on Digital Content, Multimedia Technology and Its Applications, Seoul, Republic of Korea, 16–18 August 2010; pp. 331–336.
29. Haque, A.; Abdulhussein, T.A.; Ahmad, M.; Falah, M.W.; Abd El-Latif, A.A. A strong hybrid S-box scheme based on chaos, 2D cellular automata and algebraic structure. *IEEE Access* **2022**, *10*, 116167–116181. [CrossRef]
30. Ismail, I.A.; Abdo, A.A.; Amin, M.; Diab, H. Self-Adaptive Image Encryption Based on Memory Cellular Automata. *Int. J. Inf. Acquis.* **2011**, *8*, 227–241. [CrossRef]
31. Pokkuluri, K.S.; Usha, D.N. A secure cellular automata integrated deep learning mechanism for health informatics. *Int. Arab J. Inf. Technol.* **2021**, *18*, 782–788. [CrossRef]
32. Neumann, J.V. *Theory of Self-Reproducing Automata*; University of Illinois Press: Champaign, IL, USA, 1966.
33. Clarke, K.C.; Cellular Automata and Agent-Based Models. In *Handbook of Regional Science*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1751–1766.
34. Wolfram, S. Random sequence generation by cellular automata. *Adv. Appl. Math.* **1986**, *7*, 123–169. [CrossRef]
35. Wolfram, S. *A New Kind of Science*; Wolfram Media: Champaign, IL, USA, 2002; Volume 5, Chapter 6.

36. Jiménez López, J.; Mulero-Pázmány, M. Drones for conservation in protected areas: Present and future. *Drones* **2019**, *3*, 10. [CrossRef]
37. Seier, G.; Hödl, C.; Abermann, J.; Schöttl, S.; Maringer, A.; Hofstadler, D.N.; Pröbstl-Haider, U.; Lieb, G.K. Unmanned aircraft systems for protected areas: Gadgetry or necessity? *J. Nat. Conserv.* **2021**, *64*, 126078. [CrossRef]
38. Wich, S.A.; Hudson, M.; Andrianandrasana, H.; Longmore, S.N. Drones for conservation. In *Conservation Technology*; Oxford University Press: Oxford, UK, 2021; Volume 35.
39. Vermeulen, C.; Lejeune, P.; Lisein, J.; Sawadogo, P.; Bouche, P. Unmanned aerial survey of elephants. *PLoS ONE* **2013**, *8*, e54700. [CrossRef]
40. Sindiramutty, S.R.; Jhanjhi, N.Z.; Tan, C.E.; Yun, K.J.; Manchuri, A.R.; Ashraf, H.; Murugesan, R.K.; Tee, W.J.; Hussain, M. Data Security and Privacy Concerns in Drone Operations. In *Cybersecurity Issues and Challenges in the Drone Industry*; IGI Global: Hershey, PA, USA, 2024; pp. 236–290. [CrossRef]
41. Mou, C.; Liu, T.; Zhu, C.; Cui, X. Waid: A large-scale dataset for wildlife detection with drones. *Appl. Sci.* **2023**, *13*, 10397. [CrossRef]
42. Computer Vision Group-University of Granada (CVG-UGR) Image Database. Available online: http://decsai.ugr.es/cvg/dbimagenes/ (accessed on 16 August 2024).
43. Hua, Z.; Xu, B.; Jin, F.; Huang, H. Image encryption using Josephus problem and filtering diffusion. *IEEE Access* **2019**, *7*, 8660–8674. [CrossRef]
44. Hua, Z.; Jin, F.; Xu, B.; Huang, H. 2D Logistic-Sine-coupling map for image encryption. *Signal Process.* **2018**, *149*, 148–161. [CrossRef]
45. Hua, Z.; Zhou, Y.; Huang, H. Cosine-transform-based chaotic system for image encryption. *Inf. Sci.* **2019**, *480*, 403–419. [CrossRef]
46. Belazi, A.; Kharbech, S.; Aslam, M.N.; Talha, M.; Xiang, W.; Iliyasu, A.M.; Abd El-Latif, A.A. Improved Sine-Tangent chaotic map with application in medical images encryption. *J. Inf. Secur. Appl.* **2022**, *66*, 103131. [CrossRef]
47. Abd el Latif, A.A.; Abd-el Atty, B.; Amin, M.; Iliyasu, A.M. Quantum-inspired cascaded discrete-time quantum walks with induced chaotic dynamics and cryptographic applications. *Sci. Rep.* **2020**, *10*, 1930. [CrossRef]
48. Nestor, T.; De Dieu, N.J.; Jacques, K.; Yves, E.J.; Iliyasu, A.M.; El-Latif, A.; Ahmed, A. A multidimensional hyperjerk oscillator: Dynamics analysis, analogue and embedded systems implementation, and its application as a cryptosystem. *Sensors* **2020**, *20*, 83. [CrossRef]
49. Tsafack, N.; Kengne, J.; Abd-El-Atty, B.; Iliyasu, A.M.; Hirota, K.; Abd EL-Latif, A.A. Design and implementation of a simple dynamical 4-D chaotic circuit with applications in image encryption. *Inf. Sci.* **2020**, *515*, 191–217. [CrossRef]
50. Zhang, W.; Wong, K.W.; Yu, H.; Zhu, Z.L. A symmetric color image encryption algorithm using the intrinsic features of bit distributions. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 584–600. [CrossRef]
51. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
52. Wu, Y.; Zhou, Y.; Saveriades, G.; Agaian, S.; Noonan, J.P.; Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inf. Sci.* **2013**, *222*, 323–342. [CrossRef]
53. Bassham, L.E., III; Rukhin, A.L.; Soto, J.; Nechvatal, J.R.; Smid, M.E.; Barker, E.B.; Leigh, S.D.; Levenson, M.; Vangel, M.; Banks, D.L.; et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2010. [CrossRef]
54. Hua, Z.; Li, J.; Li, Y.; Chen, Y. Image Encryption Using Value-Differencing Transformation and Modified ZigZag Transformation. *Nonlinear Dyn.* **2021**, *106*, 3583–3599. [CrossRef]
55. Wang, X.; Teng, L.; Qin, X. A novel colour image encryption algorithm based on chaos. *Signal Process.* **2012**, *92*, 1101–1108. [CrossRef]
56. L'Ecuyer, P. Uniform random number generation. *Ann. Oper. Res.* **1994**, *53*, 77–120. [CrossRef]