IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 September 2023, accepted 19 September 2023, date of publication 26 September 2023,
date of current version 3 October 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3319335*

RESEARCH ARTICLE

# AntEater: When Arnold's Cat Meets Langton's Ant to Encrypt Images

**WASSIM ALEXAN[1], (Senior Member, IEEE), YOUSEF KORAYEM[2], (Student Member, IEEE), MOHAMED GABR[2], (Member, IEEE), MINAR EL-AASSER[3], (Senior Member, IEEE), ENGY ALY MAHER[1], (Senior Member, IEEE), DINA EL-DAMAK[4], (Senior Member, IEEE), AND AMR ABOSHOUSHA[5,6]**

[1]Communications Department, Faculty of Information Engineering and Technology, German University in Cairo (GUC), New Cairo 11835, Egypt
[2]Computer Science Department, Faculty of Media Engineering and Technology, German University in Cairo (GUC), New Cairo 11835, Egypt
[3]Networking Department, Faculty Information Engineering and Technology, German University in Cairo (GUC), New Cairo 11835, Egypt
[4]Electronics Department, Faculty Information Engineering and Technology, German University in Cairo (GUC), New Cairo 11835, Egypt
[5]Physics Department, Faculty of Basic Sciences, German University in Cairo (GUC), New Cairo 11835, Egypt
[6]Physics Department, Science Faculty, Cairo University, Giza 12613, Egypt

Corresponding author: Wassim Alexan (wassim.alexan@ieee.org)

**ABSTRACT** With the proliferation of digital images over open networks, secure and efficient encryption schemes are imperative for safeguarding image privacy. This paper proposes a novel 5-stage image encryption algorithm adhering to Shannon's confusion and diffusion paradigm. In the first stage, Langton's Ant is employed to induce chaos and perturb the pixel distribution of the original image. The second and fourth stages apply Mersenne Twister generated keys to confuse the image via XOR operations. An S-box substitution is utilized in the third stage to disrupt pixel statistical properties. Finally, Arnold's Cat map further scrambles and diffuses the pixel positions over the image. Extensive security analyses reveal the algorithm's robustness against various attacks such as visual, entropy, brute-force, statistical and differential attacks. Additionally, it successfully passes the NIST SP 800-22 test suite. Performance results demonstrate the proposed algorithm's efficacy for real-time secure image transmission with low computational overheads. The algorithm's security level combined with high-speed performance makes it well-suited for practical image encryption applications.

**INDEX TERMS** Arnold's cat map, chaos theory, image cryptosystem, image encryption, Langton's Ant, Mersenne Twister.

## I. INTRODUCTION

With the exponential growth of visual data transmission through digital networks, image encryption has become crucial for protecting confidential images against unauthorized access [19]. Traditional text encryption methods often prove inadequate for scrambling meaningful visual data. Thus, there is a pressing need to develop specialized image encryption algorithms that can provide robust security for sensitive images [16], [18]. Novel techniques like chaos-based encryption show promise for highly secure real-time image scrambling [20]. Rapid image encryption is essential for various entities, including medical systems transmitting patient scans, video conferencing services, and military operations sending reconnaissance photos [35]. The military in particular requires strong encryption to secure classified visual data during transmission. This article provides a comprehensive survey of recent image encryption techniques, with analysis of their cryptographic strength. Moreover, it highlights the most secure and efficient algorithms to advance the field and enable widespread adoption of secure image communication across military, commercial, and healthcare systems. Next, it identifies a gap in those algorithms and proceeds to propose a novel image cryptosystem that satisfies high security, efficiency, resistivity to attacks, and robustness against noise.

Recent literature on image cryptosystems describes the utilization of a plethora of mathematical constructs to

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Huang.

generate pseudo-random numbers, as encryption keys, and to construct substitution boxes (S-boxes), among other image transformations [4], [55]. Such constructs include ideas pooling from chaos theory, number theory, linear algebra, information theory, graph theory, group theory, combinatorics, as well as optimization theory. In fact, the literature mostly describes algorithms that make use of a combination of these constructs. However, the next few paragraphs attempt to briefly describe each, presenting some example implementations.

Chaos theory is an important mathematical concept that has found applications in various fields, including image encryption. Chaos theory deals with the behavior of physical and dynamical systems that are highly sensitive to initial conditions, which makes their behavior unpredictable [27]. In image encryption, chaos theory has been used to generate random sequences of numbers that are used to encrypt the image data. These random sequences are generated by chaotic maps or systems, which are mathematical functions that exhibit chaotic behavior [4], [5], [7]. The use of chaotic maps or systems in image encryption provides a high degree of randomness and makes the encryption process more secure. Furthermore, the sensitivity of chaotic maps or systems to initial conditions makes it difficult for an attacker to reconstruct the original image from the encrypted data. Therefore, chaos theory is an important tool in image encryption that provides a high degree of security and robustness. Recent works on image cryptosystems have increasingly attempted to utilize hyperchaotic systems of differential equations, as a means to easily and vastly increase the key space. For example, the authors of [4] employ the fractional-order numerical solutions of the Chen hyperchaotic system in conjunction with a number of cryptographically secure PRNGs to present an efficient image cryptosystem. The same idea of utilizing the Chen system is carried out in the work in [7], where the authors combine its use with a hybrid form of DNA coding, and a simple sine chaotic map for increasing security, while maintaining code efficiency. The authors of [38] propose a 3-stage image encryption algorithm, where chaos is introduced through the movement of a single Langton's Ant over the plain image for a very large number of iterations, as well as the use of deterministic noise formulae and a jigsaw transform. Another 3-stage image cryptosystem is proposed in [5], where in every stage an encryption key is XORed with the image data, and an S-box is applied. The encryption keys and the S-boxes are produced from the numerical solutions of 2 hyperchaotic systems, as well as the single neuron model (SNM). The choice of the hyperchaotic systems leads to expansion of the key space, while the simple SNM allows for a reduced algorithm complexity. The authors of [25] propose the utilization of the Henon map, the Circle map, and the Duffing map, in a confusion-diffusion algorithm that exhibits excellent performance. The most efficient image cryptosystem encountered in this literature review is that proposed in [34], where the authors carry out an FPGA realization of a fractional-order memristive chaotic system. Their work also incorporates the use of Arnold's cat map, and achieves an encryption rate of 0.396 Gbps. The authors of [37] propose a novel encryption algorithm for medical images that combines chaotic-hash scrambling, combined hash functions, block-based substitution, and dynamic bit shifting based on Josephus sequences. Their proposed technique complements the chaotic system, which alone only scrambles the pixel position, with additional operations to improve security against statistical attacks. Adequate testing is carried to showcase the resistivity of the algorithm to statistical and differential attacks, in addition to the possession of a sufficient key space to withstand brute force attacks. The method provided in [10] involves 4 chaos methods: the logistic map, the Lorenz system, the Henon map, and the Arnold transform, blended with hash operations to augment key space and sensitivity. Symmetric key generation is proposed to include diffused plain images to enhance sensitivity. Additionally, a novel combination model of confusion on row and column pixels, and image bits, is developed. The algorithm utilizes a two-step diffusion technique, executed before and after the confusion process. This combination of techniques is designed to improve encryption quality and resistance to various attacks, including brute force, statistical, and differential attacks. The authors of [46] propose a multi-interleaved encryption technique utilizing Zigzag, Hilbert, and Morton patterns to enhance the complexity of the confusion-diffusion process. The symmetric key for the encryption process is generated from an improved logistic map and a 6D hyperchaotic map, which are chosen for their sensitivity to initial conditions and control parameters, leading to highly random and nonlinear key-streams that are difficult to predict. The encryption process is carried out in four phases, alternating between diffusion and confusion, with SHA-512 used to increase the key space and sensitivity. The proposed technique has shown resilience against various attacks, including statistical, differential, brute force, NIST randomness tests, and data loss and noise attacks.

Number theory is the study of integers and their properties. It has been used in image encryption schemes for techniques such as modular arithmetic and prime numbers. Modular arithmetic involves performing arithmetic operations on remainders obtained after dividing numbers by a fixed integer. Prime numbers are used in some image encryption algorithms for their inherent difficulty in factorization, which makes them useful in ensuring secure encryption. For example, the author of [29] attempts to provide a more efficient implementation of the public-key cryptography algorithm RSA, which depends on prime numbers, for the secure transmission of image data over unsecured networks. Another work that involves the use of prime numbers is [36], which combines the use of DNA coding and a chaotic Chebyshev map, such that the initial conditions are chosen to be prime numbers. Multiple levels of encryption are applied

in [36], such that bit XORing is carried out, as well as row and column pixel permutation are applied, for improved induced randomness. The authors of [21] apply the use of modular arithmetic in 3D scale-invariant chaotic maps, as well as a diffusion step that utilizes the Hill map. These steps are applied at least twice, which ensures the expansion of the key space and the attainment of randomness in the final encrypted output image.

Linear algebra is a field of study that involves linear equations and their properties. In image encryption, concepts from linear algebra such as matrices and vector spaces have often been used in the literature on image encryption. Matrices can be used to represent image data and transformations, while vector spaces can be used to represent the image space. For example, the work presented in [52] utilizes the Sarrus model, as borrowed from linear algebra, to carry out pixel position scrambling. Next, a Fibonacci matrix is extended to 3D and utilized to further diffuse the image pixels, resulting in a highly encrypted output image. The work in [52] is shown to be efficient and secure, exhibiting a large key space and sensitivity to small changes in encryption keys. The authors of [14] apply the simple idea of matrix multiplication and inverse matrix multiplication, in conjunction with a number of simple chaotic maps, to carry out effective image encryption. The authors of [7] introduce a novel matrix, called the KAA map, and employ it alongside a number of chaotic maps for multiple symmetric key generation. The first key utilized in [7] is generated from the 2D Logistic sine map in conjunction with the Linear Congruential Generator (LCG), while the second key is generated from the use of both of the Tent and the Bernoulli chaotic maps. The authors of this work [7] carry out a plethora of tests that reflect the security and robustness of their proposed algorithm.

Information theory is the study of the quantification, storage, and communication of information. In image encryption, principles from information theory such as entropy have been utilized. Entropy is a measure of randomness and is used in some image encryption schemes to ensure that the encrypted image has a high degree of randomness. While the literature is rich with information theory-based encryption algorithms, such as the RSA and AES [9], [13], [43], [48], [54], as well as their improved implementations [22], such algorithms are usually not the best-suited for encrypting image data. First of all, because images have high correlation among their adjacent pixels, resulting in high redundancy, as well as the large size of image data, which effectively increases the computational complexity required to carry out the encryption [4]. Second, RSA and AES were not initially developed for the encryption of images [31]. This becomes very clear when comparing their efficiency at encrypting image data with other algorithms that are custom-developed for image encryption [4].

Graph theory is the study of graphs and their properties. In image encryption, graph theory concepts such as random walks and graph coloring have been used. Random walks involve traversing a graph randomly, while graph coloring involves assigning colors to the vertices of a graph in a way that neighboring vertices have different colors. For example, the authors of [2] present an algorithm relies on audio files as a foundation for generating a set of keys to encode color images. To create the key generation system, the audio signal is passed through multiple stages based on graph theory. In [51], the authors borrow the idea of Hamiltonian paths from graph theory, where a Hamiltonian path refers to the path that visits each vertex exactly once. In their proposed algorithm, a random Hamiltonian path is generated within plain images, as an equivalent to pixel permutation in image encryption. Moreover, the authors of [51] combine this idea with an application of an adjusted Bernoulli map for the logical XORing of the image pixels. The authors of [44] employ 3-partite graphs as a means to exchange plain image pixels. Chen's chaotic system is utilized to generate the vertices of the 3-partite directed graph, such that paths are established between each pair of vertices. The work proposed in [44] is effectively secure against plain-text attacks, as it uses the seeds of the chaotic Chen system to modulate the hash value of the plain images to be encrypted.

Group theory is a branch of mathematics that provides the Abelian Groups which are used to describe symmetries in geometry, particle physics, molecular physics, as well as DNA symmetries [49]. Such a definition allows researchers to include DNA coding schemes for image encryption [7], [36] underneath the umbrella of Group theory. For example, the authors of [45] make use of a 6D discrete hyperchaotic system as basis for the production of unique encryption keys. Next, they construct DNA matrices from the color components of plain RGB images. These matrices are then scrambled, producing 3 new matrices that are applied to each of the color channels. The combination of applying chaos theory alongside DNA coding is shown to exhibit good performance in [45]. The authors of [49] combine the utilization of both discrete and continuous chaotic systems with a novel inverse left almost semi-group. The conducted security analysis in [49] showcases excellent performance.

Combinatorics is the study of discrete structures and their properties. In image encryption, combinatorial techniques such as Latin squares and error-correcting codes have been used [56]. Latin squares are used to permute the image data, while error-correcting codes are used to ensure that the encrypted image can be recovered even if some errors occur during transmission [24], [26]. For example, in [41], the authors propose the introduction of time delay to non-linear systems as a means for improving their performance. More specifically, they utilize a time-delayed non-linear combinatorial hyperchaotic map (TD-NCHM). Their proposed algorithm carries out a simultaneous row-column shuffling-diffusion in addition to the use of encryption keys. The work in [41] is shown to be both highly secure as well as efficient. The work in [40] proposes the usage of cryptographic key providing encryption based on combinatorial structures that

offer non-crossing as well as non-nesting matching. The authors describe their work in a general sense, suitable for IoT applications, including image encryption. The authors of [24] realize that earlier implementations of Latin squares in image encryption algorithms were particularly inefficient. To increase efficiency, they introduce the utilization of orthogonal Latin square, in conjunction with a 2D chaotic system.

Optimization theory is the study of finding the best solution to a problem. In image encryption, optimization techniques such as linear programming have been used. Linear programming involves finding the best solution to a linear objective function subject to linear constraints. For example, the work in [11] aims to construct a deterministic and highly nonlinear S-box through the use of a generator that is based on an integer linear programming (ILP) formulation. This S-box is then utilized in an image cryptosystem jointly with PRNG that is based on elliptic curves. The authors of [11] showcase the security of their proposed confusion-diffusion cryptosystem and its high sensitivity to changes in keys. The authors of [12] attempt to combine a hybrid multi-objective particle swarm optimization (HMPSO) with block compressed sensing (BCS) and a Hessenberg decomposition (HD), for efficient image encryption. Furthermore, their implementation results in meaningful encrypted images, with secret keys generated from the Henon chaotic map.

Table 1 provides a classification of the reviewed recent literature as it relates to each of the constructs. It is clear that there is focus of researchers on the application of chaos theory, as seen in Table 1. This is justified in terms of the greatly expanded key spaces that could be achieved through the utilization of hyperchaotic systems. This is especially true whenever such hyperchaotic systems are numerically solved at the fractional-order, which immediately adds more variables to the key space of any image cryptosystem [4], [7]. Other works have specifically targeted improved efficiency, either in terms of software [7] or hardware [34] implementations. A third group of research works focused on increasing an image cryptosystem's resistivity and robustness against specific attacks, such as occlusion attacks [4], various noise attacks, chosen plain-text attacks [4], [44], and chosen cipher-text attacks [4]. However, it is rather hard to find a research work that attempts to cover all 3 aspects, namely security, efficiency and robustness. This research work attempts to cover this gap.

In this research work, the contributions of the proposed image cryptosystem are as follows:

1) A five-stage image cryptosystem is designed and software implemented, making use of ideas pooling from Chaos theory (in terms of utilizing Langton's ant), Mersenne Twister PRNGs and linear algebra (in terms of utilizing Arnold's cat map).
2) Highly resistive to brute-force attacks, with an achieved demonstrative key space of $2^{14244}$.
3) Highly efficient, encrypting images at an average rate of 3.7 Mbps..
4) Successfully passes all NIST SP 800-22 analysis tests, proving the randomness of its output encrypted images.

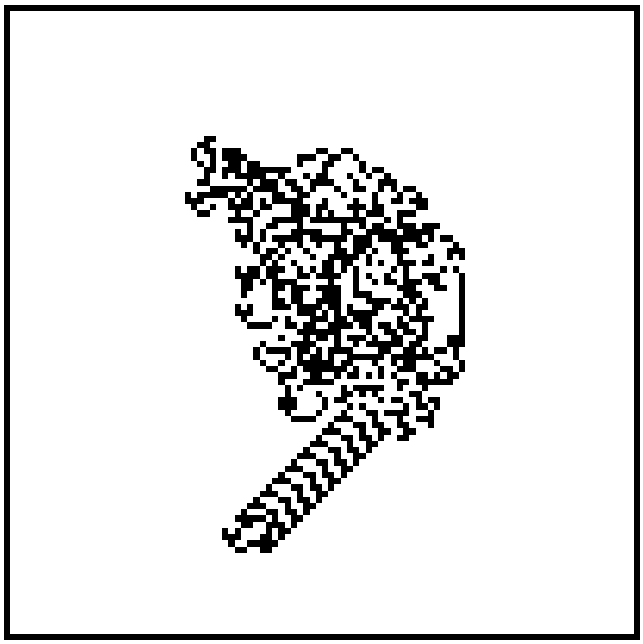


**FIGURE 1.** The original Langton's Ant configuration after 11000 steps, with one ant originally travelling up initially, turning left on black tiles, and right on white tiles; oscillation is clear after about 10000 steps, in the stepladder section at the bottom of the central mass.

The remainder of this article is organized as follows. Section II provides the preliminary mathematical constructs employed in the proposed image encryption algorithm. Section III outlines the sequence of steps carried out in the proposed image encryption algorithm. Section IV reports and discusses different performance evaluation metrics, as well as compares them to the state-of-the-art. Finally, Section V draws the conclusions and mentions future research directions that could be further pursued. Appendix is an image repository.

## II. PRELIMINARY OF THE PROPOSED IMAGE ENCRYPTION ALGORITHM

### A. LANGTON'S ANT VARIANT PRNG

Langton's Ant is the name generally attributed to a class of simple cellular automata originally proposed by Christopher Langton in [28]. Originally proposed in research on studying artificial life and the behavior of biomolecules, Langton's Ant shows how emergent behavior derived from simple rules can in fact be chaotic. The original form of Langton's Ant defines a large array of squares that can either be black or white, with an ant starting in the center. Each iteration of the system makes the ant step forward and, based on the color of the tile it is stepped on, turn left or right. The result is that after a certain number of iterations, patterns may emerge. For the default rule-set and configuration, Langton's Ant behaves somewhat unpredictably for nearly 10000 steps, after which it begins to oscillate in a predictable fashion. Figure 1 shows

**TABLE 1.** Classification of reviewed recent literature. Under the *Goal* column, the following acronyms are utilized: KS, E and RRA, signifying key space, efficiency, and robustness and resistivity to attacks.

| Ref \ Method | Chaos Theory | Number Theory | Linear Algebra | Information Theory | Graph Theory | Group Theory | Combinatorics | Optimization Theory | Goal |
|---|---|---|---|---|---|---|---|---|---|
| [2] | | | | | ✓ | | | | RRA |
| [4] | ✓ | | | | | | | | KE, E, RRA |
| [5] | ✓ | | | | | | | | KE, E, RRA |
| [11] | | | | | | | | ✓ | E, RRA |
| [25] | ✓ | | | | | | | | RRA |
| [34] | ✓ | | | | | | | | RRA |
| [36] | | ✓ | | | | ✓ | | | KE, RRA |
| [21] | | ✓ | | | | ✓ | | | E, RRA |
| [14] | ✓ | | ✓ | | | | | | RRA |
| [49] | ✓ | | | | | ✓ | | | E, RRA |
| [45] | ✓ | | | | | ✓ | | | RRA |
| [38] | ✓ | | | | | | | | KE, E, RRA |
| [41] | ✓ | | | | | | ✓ | | E, RRA |
| [24] | ✓ | | | | | | ✓ | | RRA |
| [12] | ✓ | | | | | | | ✓ | RRA |
| [52] | | | ✓ | | | | | | KE, E, RRA |
| [9] | | | | ✓ | | | | | RRA |
| [13] | | | | ✓ | | | | | RRA |
| [43] | | | | ✓ | | | | | RRA |
| [22] | | | | ✓ | | | | | E, RRA |
| [44] | ✓ | | | | ✓ | | | | RRA |
| [51] | ✓ | | | | ✓ | | | | E, KS |
| [40] | | | | | | | ✓ | | KS |
| [7] | ✓ | | ✓ | | | ✓ | | | E, RRA |
| [37] | ✓ | | | | | | | | RRA, KS |
| [10] | ✓ | | | | | | | | RRA |
| [46] | ✓ | | | | | | | | RRA, KS |
| [29] | | ✓ | | | | | | | E |
| [54] | | | | ✓ | | | | | RRA |

the behavior of Langton's Ant after 11000 steps exactly, showing seemingly chaotic behavior of the ant near the starting location, at the center of the grid, and the predictable ''tail'' formed later in the life-cycle, where the ant oscillates in the same pattern indefinitely.

While the original Langton's Ant only generates arrays of 2 unique colors, the rule-sets and configurations can be changed to produce greatly altered results. Variations of the produced Ant Field (AF) can be generated by altering the starting position of the ant on the field, the number of different colors that can be instigated by an ant contacting a tile, increasing the number of ants on the field, adding additional behavior when ants come into contact with each other, and different rules to account for situations when ants collide with the edge of the field. The actual rule-set obeyed by each ant has the greatest effect on the ant's behavior, which makes choosing effective rule-sets of great importance. Ant Field rule-sets will hereon be referred to by number. Since the rule-set effectively needs to define whether the ant turns left or right for each color, a rule can be represented as a list of bits of the same length as the number of colors, where each bit indicates whether to turn left or right on contacting that specific color. This can be represented in shorthand as a decimal number, with the highest possible usable number being $2^c$, where $c$ is the number of unique colors.

Figure 2 shows an example of a more complex AF. By relying on 4 unique colors instead of 2, a larger number of possible rule-sets can be employed, possibly generating fields that are more chaotic than previously possible using only 2.

With the generated fields becoming more and more variable, the generated fields gain some usability as encryption keys for image color channels. In theory, by increasing the number of colors to 256 some combination of the above factors should produce a field suitably chaotic to encrypt image color channels with. However, the expansion of the available color space also introduces a number of complications, such as the common appearance of non-chaotic rule-sets and the tendency of fields to produce somewhat homogeneous blobs near the ant's starting point.

The applied solution for generating a suitable AF for encryption relies on a combination of the above factors, as well as a few additional ones. The full size field is generated by constructing an array of smaller AFs. Each AF can be generated from multiple ants, each of which with predefined starting positions and velocities. Rule-sets for

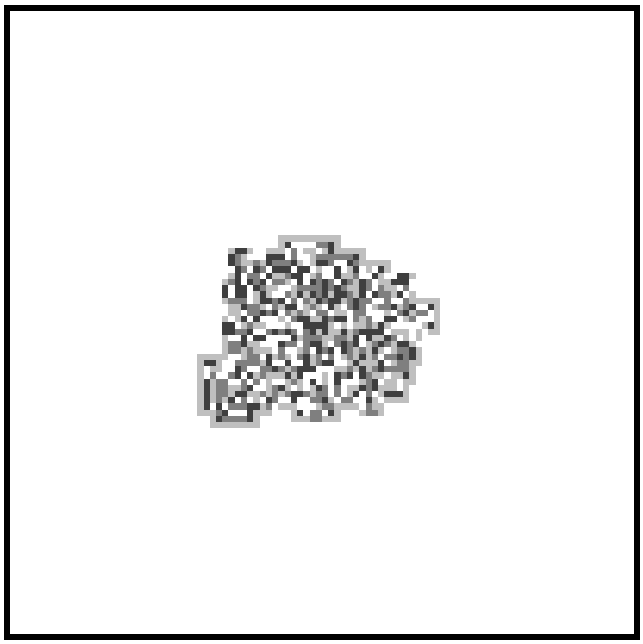


**FIGURE 2.** An example of a 100 × 100 ant field with 4 unique colors, using rule 7 (0,1,1,1). The ant, even after 11000 steps continues to oscillate in the center, leaving the rest of the field untouched.

each ant are also predefined. Ant collisions are also accounted for by having ants move like knights from chess, forward 2 steps then left once upon colliding with another ant in their field. Boundary crossing is resolved by looping the ant over to the opposite edge where the initial boundary collision occurred.

The issue that remains is that increasing the number of colors, and thus the number of possible rules, also increases the number of unreliable rules. Actual implementation of this system for encryption would involve using only rules known to generate Ant Fields that completely fill their space. Without accounting for those rules, ants may oscillate permanently in the center of their fields, like in Fig. 2 or Fig. 3a. Figures 3a, 3b, and 3c all show the effect of increasing the number of iterations in this system, which ends up filling some of the sub-fields while leaving others still blank. Figure 3d shows the successful result of curating the selected rules applied to each ant in every field.

### B. MERSENNE TWISTER PRNG AND S-BOX

The Mersenne Twister (MT) is a pseudo-random number generator (PRNG) that produces a sequence of numbers with a very long period and high order of equidistribution. Its advantages include fast generation speed [39], a huge period length of $2^{19937} - 1$ [32], and good statistical properties. Due to these qualities, the Mersenne Twister is well-suited for use in image encryption algorithms [1], [4], [15], where it can generate random-looking keys or initialization vectors to encrypt image pixels in a secure and efficient manner. Compared to other PRNGs, it has a longer period, better equidistribution, and faster speed, making the Mersenne Twister a gold standard for stochastic simulation and encryption tasks like image scrambling [42]. Moreover, the MT is easily implementable over a wide range of programming languages (e.g. Wolfram Mathematica®, Mathworks Matlab®, and MS Excel®), allowing it accessibility to many developers [33]. For illustrative purposes, an array plot of 100 × 50 random pixels is generated using the MT with a seed of 0123 and shown in Fig. 4.

### C. ARNOLD's CAT MAP

Arnold's Cat Map is a chaotic map that effectively serves as a source of permutation in an image encryption algorithm. The discrete form of the map, which is useful for application in encryption systems, is a 2D chaotic map that is defined as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} (modN), \tag{1}$$

where $x$ and $y$ represent a pixel's coordinates in an image, and $N$ represents the full size of the image. In essence, the map stretches the original image beyond the boundaries of the image's frame, and then reconstructs the image within those boundaries using the modulus operator. This process is repeated as many times as necessary to ensure a sufficient shuffling of the image's pixels is reached. The map can be further generalized to be defined as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & 1+ab \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} (modN), \tag{2}$$

where $a$ and $b$ are input variables that affect the transformation. Since the matrix used is a square matrix, its inverse can be computed, given $a$ and $b$ to revert the mapping. Figure 5 shows a sample image and the effect of various iterations of the Cat Map on it.

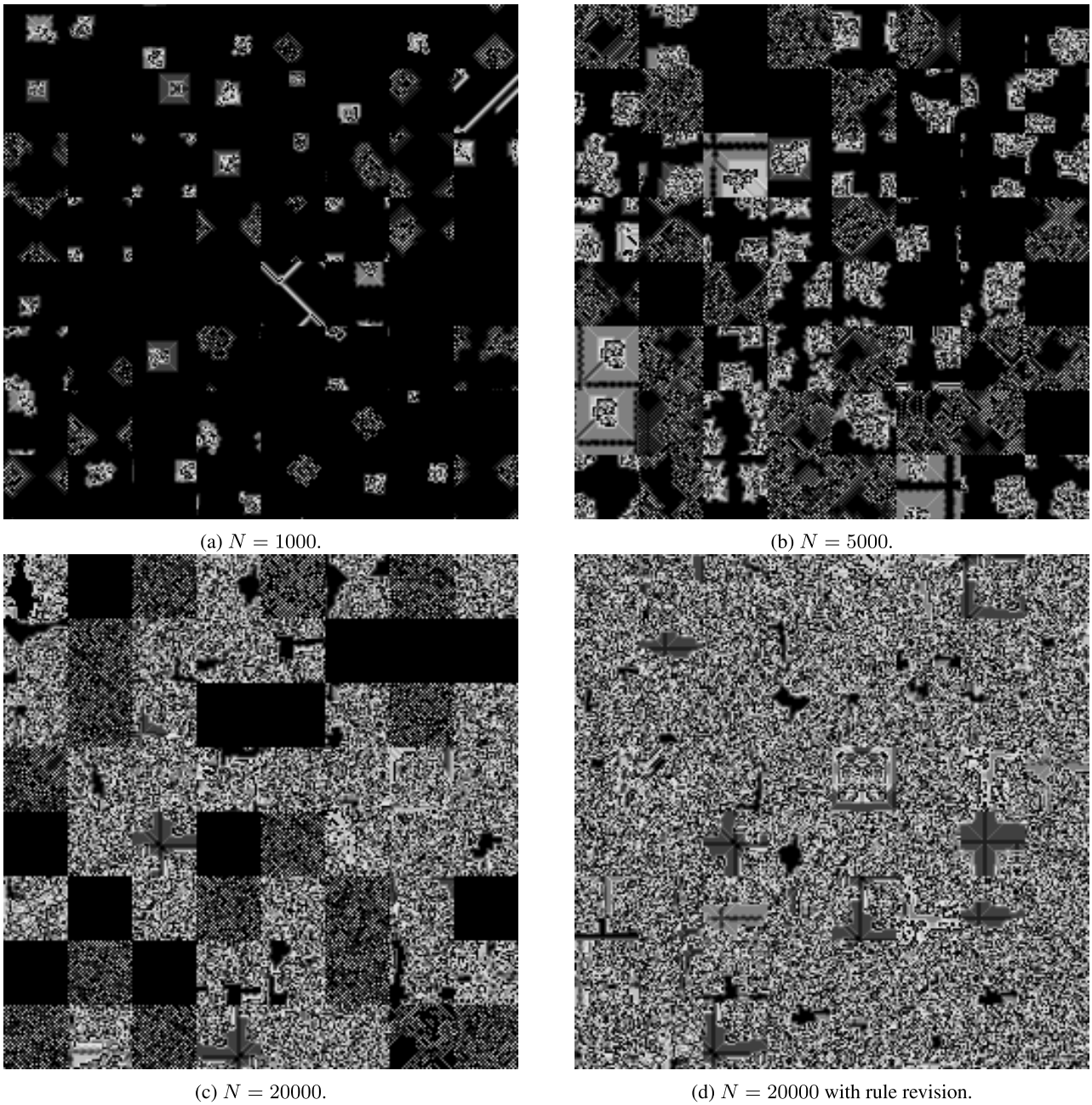
It should be noted that the Cat Map is a periodic transformation. For specific values of $a$ and $b$, repeatedly applying the Cat Map to an input image may return it to its original form, making additional applications of it redundant.

## III. METHODOLOGY OF THE PROPOSED COLOR IMAGE CRYPTOSYSTEM

This section describes in detail the process which the proposed scheme uses to encrypt and decrypt images. Subsection III-A describes the generation of the various keys used in a batch of encryptions, while the other sub-sections, III-B and III-C, discuss the actual encryption and decryption processes.

The complete encryption process utilizes 5 different stages, each of which requires different components that must be provided, or generated in the key generation stage prior to encryption. The stages are as follows:

1) Encryption of individual image color channels (red, green, and blue) using composite AFs of compatible sizes named $AF_r$, $AF_g$, and $AF_b$.
2) Permutation of image contents using the pre-generated S-Box, $SBox_{MT}$.



(a) $N = 1000$. (b) $N = 5000$.

(c) $N = 20000$. (d) $N = 20000$ with rule revision.

**FIGURE 3.** Four examples of a 256 × 256 Composite Ant Field (CAF) with 64 sub-fields, 8 unique colors, 1 ant per sub-field, random start locations, random start velocities, and random rule-sets. Figures 3a, 3b, 3c have no rule-set revision, and 3d does. Each CAF is generated with *N* steps.

3) Bit-wise XORing of the image contents with an encryption key $K_1$.
4) Cat Map Application.
5) Second Bit-wise XORing, using encryption key $K_2$.

### A. THE KEY GENERATION PROCESS

1) Ant Field Generation:
   a) Three Ant Fields of the same size as the input image are generated as described in section II-A. This requires the input of the following variables for every field:
      i) The number of sub-fields
      ii) The number of ants per sub-field
      iii) The starting positions of each ant in every field
      iv) The starting directions of each ant in every field
      v) The number of unique colors each cell in the field can have
      vi) The rules followed by every ant in every field
      vii) The number of steps that each ant in every field can take

**FIGURE 4.** A 0123 seeded-MT array plot of dimensions 100 × 50.

b) The generated composite AFs must be of the same size as the input image.
c) The generated composite AFs are referred to henceforth as $Ant_r$, $Ant_g$, and $Ant_b$.

2) Encryption Key Generation:
a) Two keys are generated using the Mersenne Twister pseudo-random number generator. The keys are each a number of bytes equivalent to the length of the image data.
b) Each key requires is its own seed value for the Mersenne Twister PRNG.
c) The generated keys are henceforth referred to as $K_1$ and $K_2$.

3) S-Box Construction:
a) An S-box is generated using Algorithm 1
b) The S-box is referred to as $SBox$

**Algorithm 1** S-Box Construction Algorithm

1) Use a seed value to initialize the Mersenne-Twister PRNG.
2) Generate enough values in the range $1-256$ to generate 10 S-boxes.
3) Test each of the S-boxes using the metrics in Section IV.
4) Choose the S-box with the best results.

### B. THE ENCRYPTION PROCESS

1) Stage 1: Ant Application.
a) The image byte data is separated into 3 color channels, named $I_r$, $I_g$, and $I_b$.
b) Channels $I_r$, $I_g$, and $I_b$ are mixed on the bit-level with the contents of $Ant_r$, $Ant_g$, and $Ant_b$ respectively, as follows:

$$\begin{aligned} I'_r &= I_r \oplus Ant_r, \\ I'_g &= I_g \oplus Ant_g, \\ I'_b &= I_b \oplus Ant_b. \end{aligned} \tag{3}$$

c) The 3 encrypted color channels are then recombined to form the image $I_{Ant}$.

2) Stage 2: S-Box Application.
a) The selected S-Box, $SBox$, is here used to permute the byte contents of the image $I_{Ant}$
b) The byte data is permuted across columns, rows, and different pixels.
c) The produced image is referred to as $I_{Ant,SBox}$

3) Stage 3: Key 1 Application.
a) Similar to with the first stage, the key $K_1$ is XORed bit-wise with the contents of $I_{Ant,SBox}$ as follows:

$$I_{Ant,SBox,K_1} = I_{Ant,SBox} \oplus K_1. \tag{4}$$

4) Stage 4: Cat Map Application.
a) The Cat Map effectively applies a permutation to the indicated image, shuffling the pixel contents in the image space deterministically.
b) The Cat Map requires 3 variables to determine the exact output of its transformation:
i) The number of iterations of the Cat Map transformation applied to the input image.
ii) The $a$ and $b$ values of the matrix.
c) With the above variables in hand, the Cat Map is applied as many times as required with the correct matrix to transform the contents of $I_{Ant,SBox,K_1}$ into $I_{Ant,SBox,K_1,Cat}$.

5) Stage 5: Second Key Application.
a) To ensure the security of the encryption, the final image contents are XORed once more, this time with $K_2$, as follows:

$$I_{Ant,SBox,K_1,Cat,K_2} = I_{Ant,SBox,K_1,Cat} \oplus K_2. \tag{5}$$

With the encryption process complete, the image data is recombined to form the encrypted image $I'$. The complete encryption is shown in Fig. 6 as a flow chart.

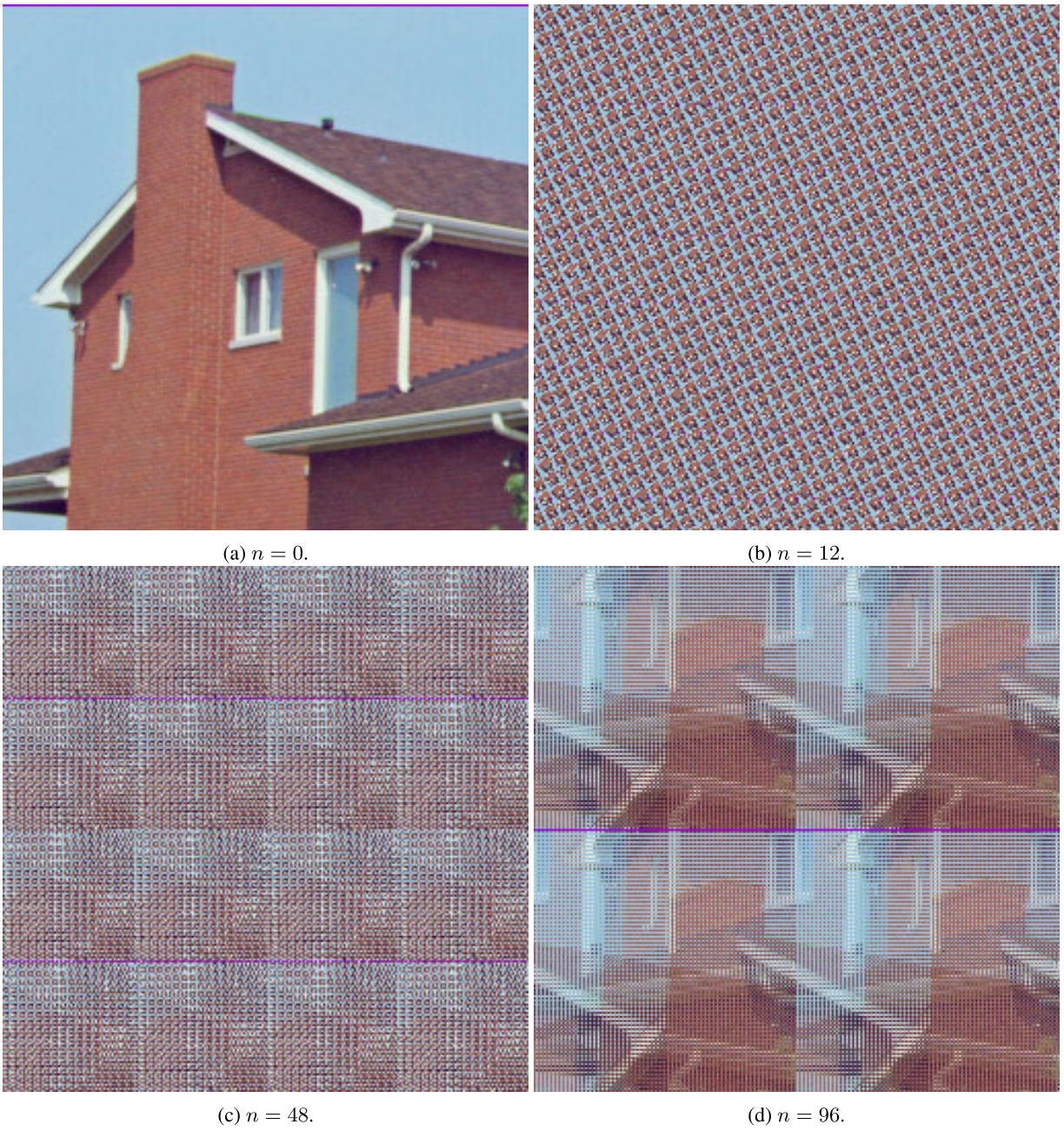
### C. THE DECRYPTION PROCESS

Decryption, being the inverse process of encryption, involves the dismantlement of the layers applied in encryption to reveal the original source image. This can only be done with the required keys, or the input variables used to generate those keys, having been securely transmitted to the intended recipient of the encrypted image. Some of the encryption layers may also require inverted keys or constructs to be generated and applied in turn. The complete decryption process is as follows, beginning with the encrypted image referred to as $I'$:

1) Stage 5: Key 2 Application.
a) Since the XOR operation is invertible, decrypting the final layer in the encryption is simply a matter of XORing the encrypted image with $K_2$ directly, as follows:

$$I'_{K_2} = I' \oplus K_2. \tag{6}$$

2) Stage 4: Cat Map Application.
a) There are 2 possible techniques for removing the Cat Map layer from the original encryption.



(a) $n = 0$. (b) $n = 12$.

(c) $n = 48$. (d) $n = 96$.

**FIGURE 5.** The House image upon application of different numbers of iterations of the Cat Map, with $a = 1$ and $b = 1$.

As described in Section II-C, the discrete form of the Cat Map is a transformation derived from an invertible matrix. Thus, the decryption may be performed by inverting the used matrix (as generated from the input variables $a$ and $b$) and iterating the process as many times as needed.

b) It may be possible for the decryption to be completed faster in certain cases where $a$ and $b$ form a well-known matrix. The Cat Map iterated operation can allow for some matrices to display periodicity upon repetition. If the period of the matrix is known, iterating forward with the original matrix until an integer multiple of the original period is reached may possibly be a faster operation.

c) In either case, inverting the matrix and iterating as many times as originally done is reliable and always produces $I'_{K_2,Cat}$

3) Stage 3: Key 1 Application.

a) Similar to with the first stage, the key $K_1$ is XORed bitwise with the contents of $I'_{K_2,Cat}$ as follows:

$$I'_{K_2,Cat,K_1} = I'_{K_2,Cat} \oplus K_1. \tag{7}$$

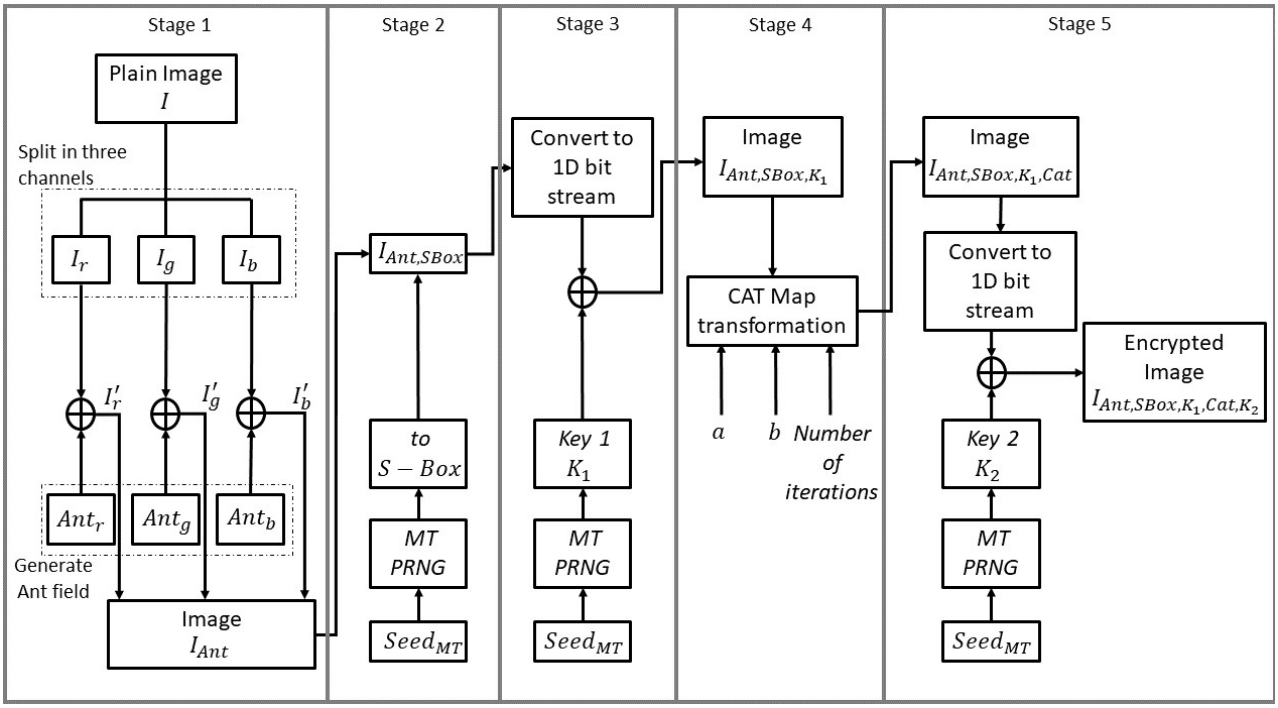



**FIGURE 6.** Flow chart of the proposed encryption process.

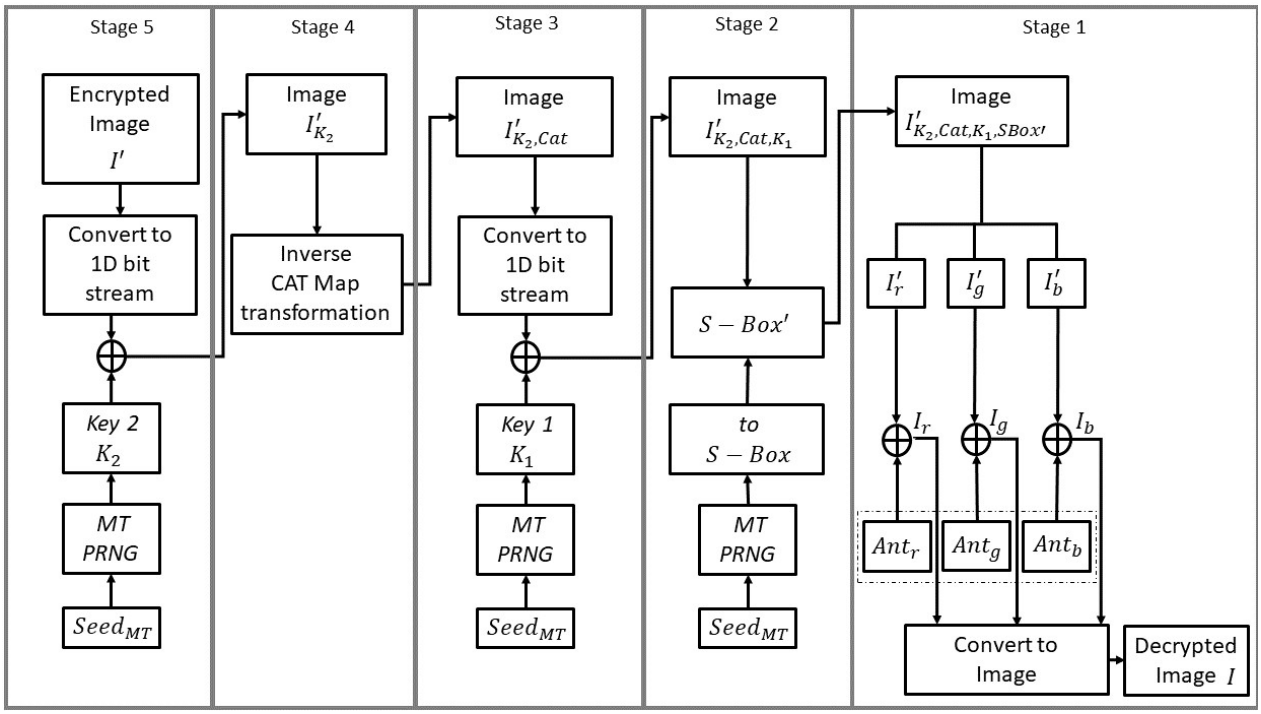
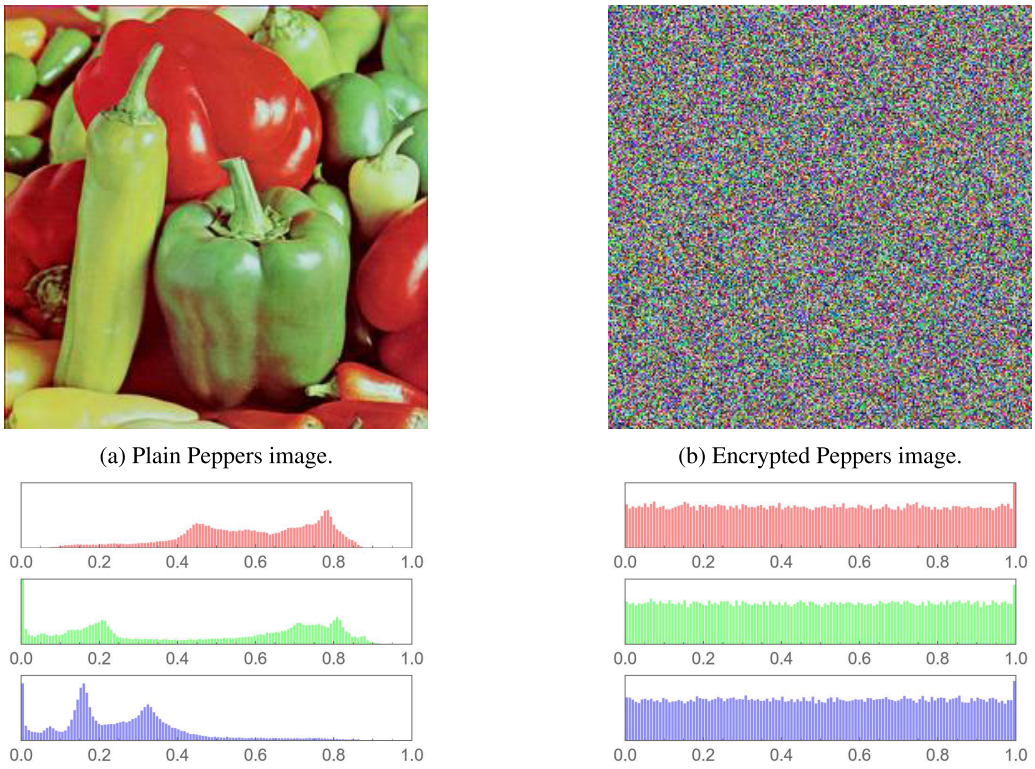



**FIGURE 7.** Flow chart of the proposed decryption process.

4) Stage 2: S-Box Application.
   a) Just as with the original encryption, the S-Box is applied at this stage to permute the pixel byte values of the original image.
   b) The original S-Box *SBox* must be inverted to form $SBox'$, which can provide the inverse of the original permutation during the encryption.
   c) Upon applying $SBox'$, $I'_{K_2,Cat,K_1,SBox'}$ is produced.
5) Stage 1: Ant Extermination.
   a) Just as with the first stage of the original image encryption, the partially decrypted image $I'_{K_2,Cat,K_1,SBox'}$ is split into 3 color channels, $I'_r$, $I'_g$, and $I'_b$.




(a) Plain Peppers image.

(b) Encrypted Peppers image.

(c) Histogram of plain Peppers image.

(d) Histogram of encrypted Peppers image.

**FIGURE 8.** Plain and encrypted versions of the Peppers image, and their respective histogram plots.

b) These 3 color channels are XORed with the same combined Ant Fields built during the key generation step, as follows:

$$\begin{aligned} I_r &= I'_r \oplus Ant_r, \\ I_g &= I'_g \oplus Ant_g, \\ I_b &= I'_b \oplus Ant_b. \end{aligned} \tag{8}$$

c) The 3 decrypted color channels are finally combined to form the completely decrypted image, $I$.

With the encryption process complete, the image data is restructured to form the decrypted image $I$. The complete decryption is shown in Fig. 7 as a flow chart.

## IV. SECURITY ANALYSIS AND NUMERICAL RESULTS

This section describes the various tests and measurements done on the proposed cryptosystem to measure its performance. A number of metrics often used in image encryption literature are applied to a number of images here, which are then compared with counterpart values from algorithms described in related research on image encryption. The images used here are all sourced from the University of Southern California's Signal and Image Processing Institute's miscellaneous image library. All of the images are taken at a size of $256 \times 256$, unless mentioned otherwise. The following tests and metrics are discussed in this section:

- Visual and Histogram Analyses (Section IV-A)
- Mean Squared Error (Section IV-B)
- Peak Signal-to-Noise Ratio (Section IV-C)
- Mean Absolute Error (Section IV-D)
- Information Entropy (Section IV-E)
- Correlation Coefficient (Section IV-F)
- Fourier Transformation Analysis (Section IV-G)
- Histogram Dependency Tests (Section IV-H)
- Differential Attack Analysis (Section IV-I)
- Key Space Analysis (Section IV-K)
- Execution Time Analysis (Section IV-L)
- The National Institute of Standards and Technology Analysis (Section IV-M)
- S-Box Performance Analysis (Section IV-N)

### A. VISUAL AND HISTOGRAM ANALYSES

The objective of any cryptographic system is to obscure the relationship between the plain-text and the cipher-text. For an image encryption system, this requires that as little as possible information about the contents of the original image be revealed by the contents of the encrypted image. The contents should be present in some form, otherwise decrypting the image would be impossible and the data would actually be lost, but whatever information is there should not be immediately obvious.

The first test to ensure the effectiveness of the proposed system relies on the Human Visual System (HVS) and its examination of both the encrypted images and the encrypted images' color histograms. Figure 8 shows the results of

**TABLE 2.** MSE values comparison of different images.

| | Proposed | [49] | [4] | [5] | [25] |
|---|---|---|---|---|---|
| Lena | 8913.31 | 4859.03 | 8912.4 | 8890.05 | 10869.73 |
| Peppers | 10144.4 | 7274.44 | 10,065.4 | 10,074.0 | N/A |
| Mandrill | 8316.4 | 6399.05 | 8320.41 | 8345.25 | 10930.33 |
| Girl | 12300.3 | N/A | 12104.2 | 12,152.8 | N/A |
| House | 8383.62 | N/A | 8395.53 | 8361.44 | N/A |
| House2 | 9172.77 | N/A | 9142.54 | 9190.27 | N/A |
| Sailboat | 10050.4 | N/A | 10071.9 | 10063.3 | N/A |
| Tree | 9993.67 | N/A | 9873.24 | 9931.63 | N/A |
| Average | 9659.4 | 6177.5 | 9610.65 | 9626.09 | 10900 |

**TABLE 3.** PSNR values comparison of different images.

| | Proposed | [49] | [4] | [5] | [25] |
|---|---|---|---|---|---|
| Lena | 8.63042 | 11.30 | 8.63086 | 8.64176 | 7.7677 |
| Peppers | 8.06855 | 9.55 | 8.10248 | 8.09877 | N/A |
| Mandrill | 8.93145 | 10.10 | 8.92936 | 8.91641 | 7.7447 |
| Girl | 7.23164 | N/A | 7.30144 | 7.28403 | N/A |
| House | 8.89649 | N/A | 8.89032 | 8.90799 | N/A |
| House2 | 8.5058 | N/A | 8.52013 | 8.49752 | N/A |
| Sailboat | 8.10895 | N/A | 8.0997 | 8.10339 | N/A |
| Tree | 8.13355 | N/A | 8.18621 | 8.1606 | N/A |
| Average | 8.3134 | 10.3167 | 8.33256 | 8.32631 | 7.7562 |

**TABLE 4.** MAE values comparison of various images.

| Image | Proposed | [49] | [4] | [5] | [25] |
|---|---|---|---|---|---|
| Lena | 77.5407 | 79.22 | 77.4877 | 77.409 | 87 |
| Peppers | 82.3332 | N/A | 81.9832 | 82.0156 | 92 |
| Mandrill | 75.1741 | N/A | 75.1632 | 75.3335 | N/A |
| Girl | 90.6731 | N/A | 89.9807 | 90.1646 | N/A |
| House | 75.4068 | N/A | 75.4983 | 75.3132 | N/A |
| House2 | 78.5196 | N/A | 78.3327 | 78.5675 | N/A |
| Sailboat | 81.9821 | N/A | 82.1003 | 82.0101 | N/A |
| Tree | 81.7889 | N/A | 82.1003 | 81.4948 | N/A |

**TABLE 5.** Comparison of entropy values of the Lena image RGB channels.

| Algorithm | Entropy values of channels | | |
|---|---|---|---|
| | Red | Green | Blue |
| Proposed | 7.99522 | 7.99523 | 7.99514 |
| [49] | 7.9965 | 7.9970 | 7.9971 |
| [4] | N/A | N/A | N/A |
| [5] | N/A | N/A | N/A |
| [25] | 7.9973 | 7.9972 | 7.9975 |
| [47] | 7.9948 | 7.9958 | 7.9950 |
| [45] | 7.9974 | 7.9970 | 7.9969 |

encrypting the plain Peppers image, as well as the effect on that image's distribution. It is clear that the encrypted image reveals no information about the contents of the original image, lacking any clear edges or defining features: there is no visual symmetry or correlation between the plain image and the encrypted one.

The color histograms indicate that the encryption scheme successfully obscures any color related data as well. Statistical cryptanalysis techniques may take advantage of an image's unique color distribution to reveal some information about it given the encrypted image. However, the encrypted image's color histogram shows a completely homogenized distribution, with no discerning features to be taken advantage of in an attack.

These same properties are visible not just in the encrypted Peppers image, but also in the House, House2, Mandrill, Tree, Sailboat, Girl, and Lena images, all displayed in the figures located in Appendix.

### B. MEAN SQUARED ERROR

Although a quick examination using the HVS is useful to tell at a glance whether the encryption system is working as intended, a more thorough mathematical examination is necessary to ensure that the encrypted image actually differs strongly from its source image. One such metric used to determine this level of difference is the Mean Squared Error (MSE). The MSE is a widely used metric in image encryption literature and is expressed as follows:

$$MSE = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(P_{(i,j)} - E_{(i,j)})^2}{M \times N}, \tag{9}$$

where $P_{(i,j)}$ and $E_{(i,j)}$ are pixels from the plain and encrypted images, respectively, at location $(i, j)$, in an image of size $M \times N$. The MSE effectively takes the average of the squared difference between 2 respective pixels in 2 different images. Ideally this value would be as high as possible, signifying as great a difference as possible between the 2 images, and a cryptosystem that is highly resistant to statistical attacks.

The larger the computed MSE value between a plain image and its encrypted image, the better is the performance of an image cryptosystem, as this would reflect better resilience against statistical attacks. Table 2 displays the computed MSE values for the proposed cryptosystem and the values computed by algorithms from similar research. The table shows that the MSE values of the various tested images perform comparably to those in similar research.

### C. PEAK SIGNAL-TO-NOISE RATIO

Related to the MSE is a metric called the Peak Signal-to-Noise Ratio (PSNR). Just as the MSE presents the average difference between respective pixels in 2 images, the PSNR reflects the average difference between an encrypted image's average difference from its source and its maximum possible intensity. The PSNR is calculated as follows:

$$PSNR = 10\log\left(\frac{I_{max}^2}{MSE}\right), \tag{10}$$

where $I_{max}$ represents the maximum pixel intensity of a grayscale image. In the case of the single-channel images used in this research, $I_{max} = 255$. Since the PSNR relies on the inverse of the MSE for its calculation, a lower PSNR value typically indicates better performance of the cryptosystem as a whole. Table 3 shows the calculated PSNR values for a number of images tested using the proposed encryption system, as well as the values from similar research articles. Table 3 clearly shows performance that is similar to that of the literature cited, if not better at times.

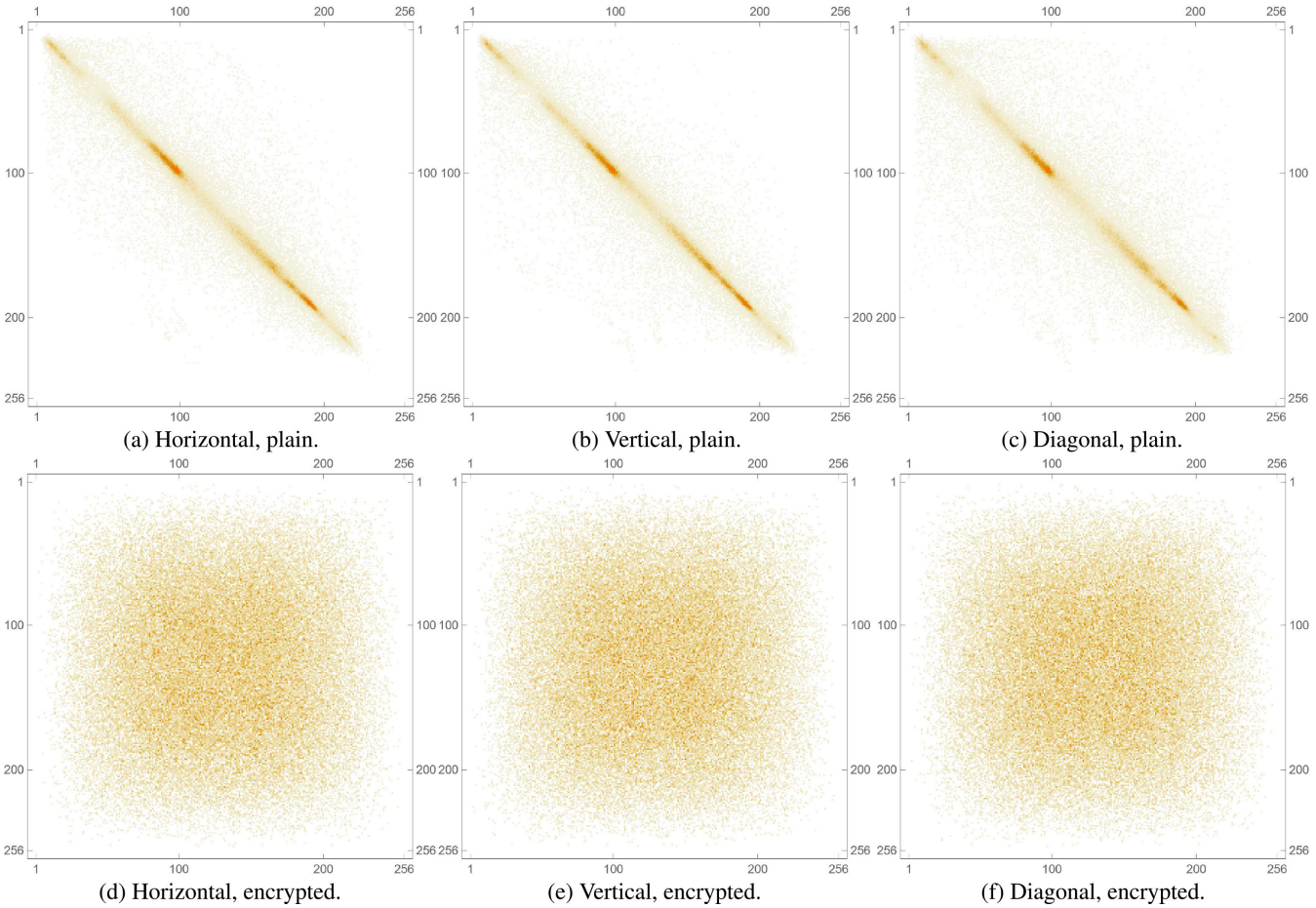



**FIGURE 9.** Correlation Coefficient diagrams of the plain and encrypted Peppers images.

**TABLE 6.** Comparison of the entropy values of the Lena image of the proposed cryptosystem and various algorithms from the literature.

| Algorithm | Entropy value |
|---|---|
| Proposed | 7.99702 |
| [49] | 7.9968 |
| [4] | 7.99887 |
| [5] | 7.999 |
| [25] | 7.9990 |
| [47] | 7.9985 |

**TABLE 7.** Horizontal, diagonal, and vertical calculations of the correlation coefficients in various plain images.

| Plain image | H | D | V |
|---|---|---|---|
| Lena | 0.959422 | 0.930426 | 0.79088 |
| Peppers | 0.959422 | 0.930426 | 0.966795 |
| Mandrill | 0.848778 | 0.750624 | 0.79088 |
| Girl | 0.974013 | 0.974013 | 0.965671 |
| House | 0.978232 | 0.952926 | 0.952926 |
| House2 | 0.907074 | 0.907074 | 0.92309 |
| Sailboat | 0.950138 | 0.919872 | 0.950138 |
| Tree | 0.968153 | 0.929967 | 0.919872 |

### D. MEAN ABSOLUTE ERROR

A third test that can indicate an encrypted image's difference from its source is the Mean Absolute Error (MAE). Also useful for ensuring an encryption system's resistance to differential attacks, the MAE acts similarly to the MSE, comparing respective pixels across different images and taking the difference between them. However, where the MSE takes the square of the difference in its mean computation, the MAE takes the absolute value of that difference, as defined here:

$$MAE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} P_{(i,j)} - E_{(i,j)}, \tag{11}$$

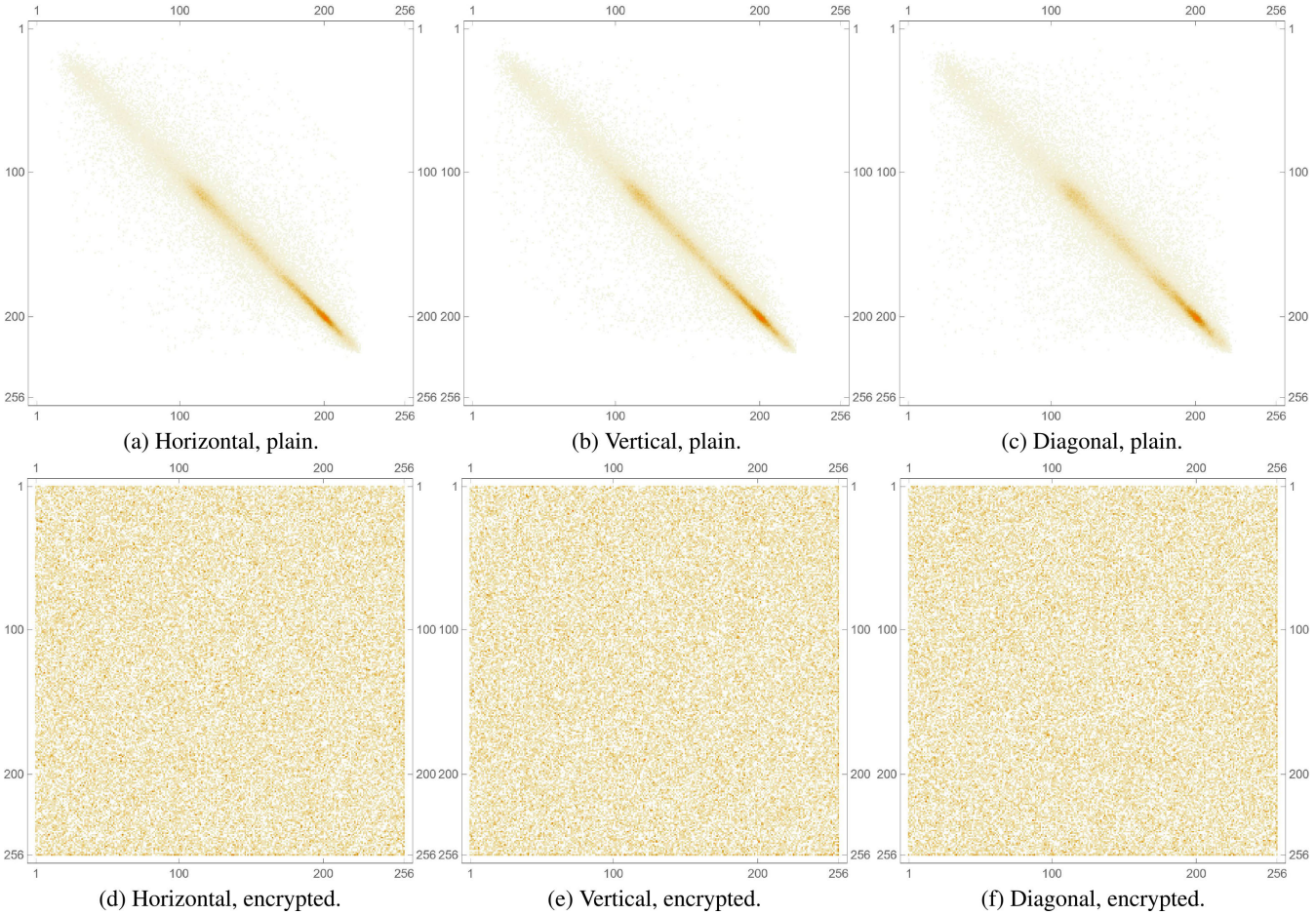
where the plain image's pixels are represented by $P_{(i,j)}$, the encrypted image's pixels are represented by $E_{(i,j)}$, and both

**TABLE 8.** Horizontal, diagonal, and vertical calculations of the correlation coefficients in various encrypted images.

| Enc. image | H | D | V |
|---|---|---|---|
| Lena | −0.00581238 | −0.00191576 | −0.00153603 |
| Peppers | 0.00157365 | 0.00494262 | −0.0067555 |
| Mandrill | −0.00286339 | −0.00280182 | −0.00290341 |
| Girl | 0.00527438 | 0.00157293 | 0.00477274 |
| House | 0.000702559 | 0.0021423 | −0.000369936 |
| House2 | −0.00712483 | −0.00430762 | 0.00725773 |
| Sailboat | 0.00410397 | −0.0052046 | 0.0012498 |
| Tree | −0.00391053 | −0.0000389741 | 0.00307199 |

**TABLE 9.** Correlation coefficient comparison of plain and encrypted Lena image color channels with the literature.

| Schemes | R | | | G | | | B | | |
|---|---|---|---|---|---|---|---|---|---|
| | H | D | V | H | D | V | H | D | V |
| Plain | 0.952474 | 0.928029 | 0.975913 | 0.935628 | 0.910534 | 0.966647 | 0.917439 | 0.888482 | 0.947961 |
| Proposed | −0.000174528 | −0.000804767 | −0.00357241 | −0.00259291 | −0.00294638 | −0.00122533 | −0.000445844 | −0.00342736 | −0.00365199 |
| [49] | 0.0084 | 0.0016 | 0.0052 | 0.0028 | 0.0012 | 0.0066 | 0.0072 | 0.0013 | 0.0098 |
| [4] | 0.00771152 | 0.003263 | 0.00199022 | 0.000053 | 0.0026447 | 0.003507 | 0.000962 | 0.004093 | 0.00259674 |
| [5] | 0.00266725 | 0.00564219 | 0.0008351 | 0.00307568 | 0.0020740 | 0.0011823 | 0.00046821 | 0.0025489 | 0.0053944 |
| [25] | 0.0017 | 0.0049 | 0.0004 | 0.0011 | 0.0002 | 0.0076 | 0.0030 | 0.0049 | 0.0050 |



(a) Horizontal, plain. (b) Vertical, plain. (c) Diagonal, plain.

(d) Horizontal, encrypted. (e) Vertical, encrypted. (f) Diagonal, encrypted.

**FIGURE 10.** Correlation Coefficient diagrams of the plain and encrypted red channels of the Peppers image.

**TABLE 10.** Correlation coefficient comparison between plain and encrypted Lena images.

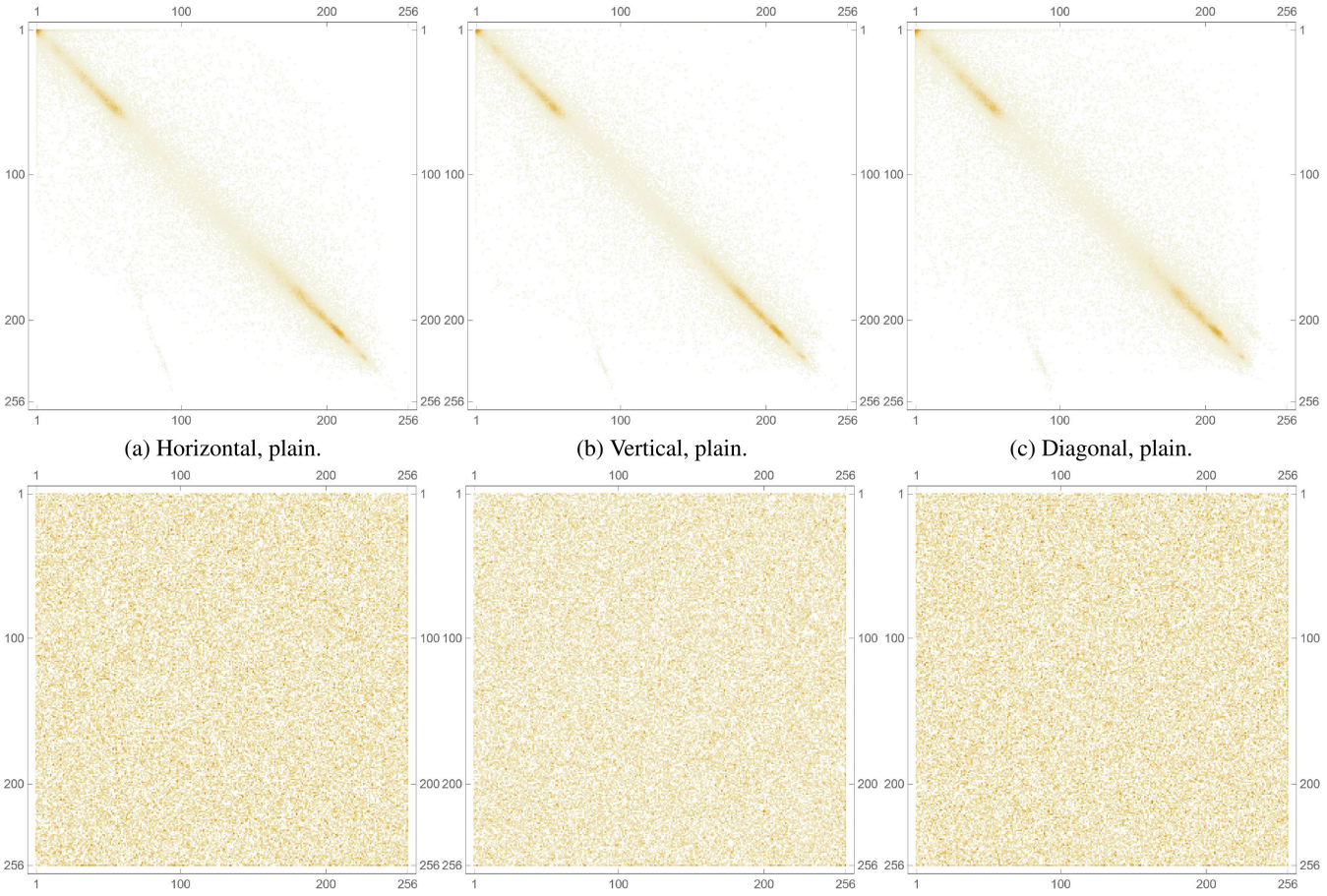
| Direction | Horizontal | Diagonal | Vertical |
|---|---|---|---|
| Plain | 0.938611 | 0.913175 | 0.96833 |
| Proposed | −0.00581238 | −0.00191576 | −0.00153603 |
| [49] | 0.0043 | 0.00310 | 0.0090 |
| [4] | 0.0064113 | 0.0015143 | 0.000568333 |
| [5] | 0.00180128 | 0.000991502 | 0.000186079 |
| [25] | 0.0054 | 0.0054 | 0.0016 |

images are of dimensions $M \times N$. Just as with the MSE, a larger MAE value indicates a greater difference between the source image and its encrypted counterpart, which implies better performance for the encryption system as a whole. Table 4 shows the MAE for the same images tested in IV-B and IV-C, as well as MAE values from similar research. Once again, the table shows that the encryption system performs comparably to the other algorithms, if not better in some cases.

### E. INFORMATION ENTROPY

To further ensure the effectiveness of the encryption algorithm, an analysis of the data actually produced by the encryption algorithm is carried out. One such analysis is a test of the Shannon Information Entropy of the image, which is generally calculated as follows:

$$H(m) = \sum_{i=1}^{M} p(m_i) \log_2 \frac{1}{p(m_i)}, \tag{12}$$



(a) Horizontal, plain. (b) Vertical, plain. (c) Diagonal, plain.

(d) Horizontal, encrypted. (e) Vertical, encrypted. (f) Diagonal, encrypted.

**FIGURE 11.** Correlation Coefficient diagrams of the plain and encrypted green channels of the Peppers image.

where $p(m_i)$ represents the probability of occurrence of each symbol $m$ in the total number of $M$ symbols in an image. This is performed over the 3 separate color channels of the image. In essence, information entropy describes just how many bits are needed, on average, to store the data in the set as a whole. For a perfectly random RGB image, that value would be 8, the maximum size given for storing color data in any pixel [53]. Tables 5 and 6 compare the entropy values from the Lena image with those of similar research articles, indicating a fair amount of randomness in the encrypted images. While not as similar as the results for the MSE and PSNR metrics, the values still remain somewhat comparable.

### F. CORRELATION COEFFICIENT ANALYSIS

In further ensuring the randomness and the distribution of the input images' visual data in their encryption, a computation of the image's correlation coefficients is carried out. The correlation coefficient $r$ helps assess the relations between adjacent pixels in any of the horizontal (H), vertical (V), or diagonal (D) directions across an image. Since typical images will contain some kind of cross-correlation between pixels in close proximity, any strong encryption system must be capable of eradicating those relations and hiding any identifying visual data. Equations (13) to (16) describe the method by which the correlation coefficient is calculated. Equation (16) is first used to calculate the average pixel distribution in each image, followed by (15) calculating the dispersion of those distributions. Equation (14) computes the covariance of each image's distribution and, with all those values in hand, (13) calculates the correlation coefficient:

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{13}$$

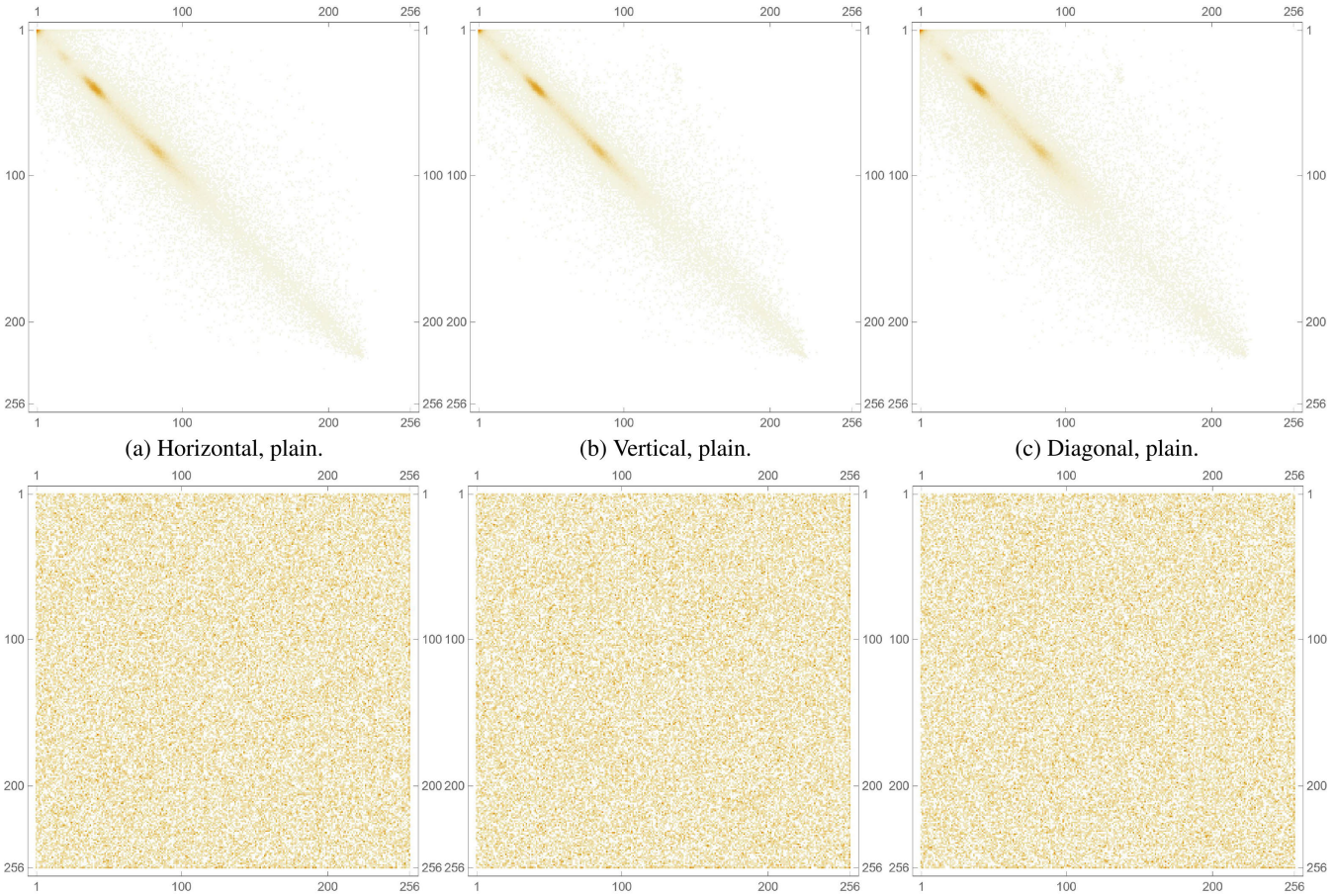
where

$$cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \tag{14}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2, \tag{15}$$

and

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i). \tag{16}$$



(a) Horizontal, plain. (b) Vertical, plain. (c) Diagonal, plain.

(d) Horizontal, encrypted. (e) Vertical, encrypted. (f) Diagonal, encrypted.

**FIGURE 12.** Correlation Coefficient diagrams of the plain and encrypted blue channels of the Peppers image.

where $x$ and $y$ are 2 images, while $N$ is the total number of bits representing all the pixels in one of the 2 images.

Ideally, a well encrypted image would have all its correlation coefficients as 0 indicating no relationships between adjacent pixels of any type. Values close to $\pm 1$ indicate a very strong correlation, which should only appear in plain images that have strong substructures. An encrypted image should show $r$ values close to 0 in every possible direction.

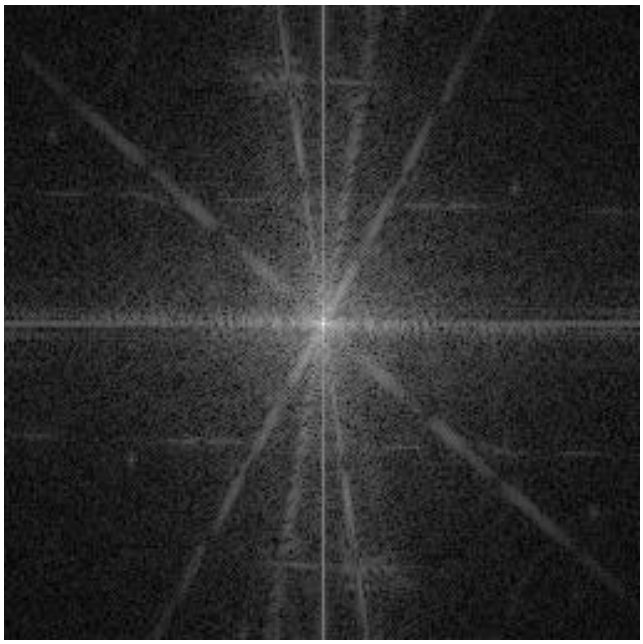
Tables 7 and 8 show the calculated $r$ values for the plain and encrypted images analyzed previously. As expected, the $r$ values for the plain images in Table 7 are quite high, representing normal pixel correlation in the plain images, while the $r$ values in Table 8 are all quite low, indicating that the correlations were dispersed upon encryption. This dispersion is also made clear visually in Fig. 9, where sub-figures 9a, 9b, and 9c show normal pixel behavior in plain images, and sub-figures 9d, 9e, and 9f show the uniform, scattered distribution in the plots expected from correlation coefficient analysis on encrypted images. Similar plots are displayed in Figs. 10, 11, and 12, which show the same relationships but for the individual color channels. Once again, the plain images' correlation plots show strong lines down the center, indicating high pixel correlation, while the encrypted correlation coefficients show highly scattered plots that indicate a complete lack of pixel correlation.

Tables 9 and 10 compare the correlation coefficient values calculated for the Lena image with those from similar research. It is clear that the proposed algorithm encrypts images with an effectiveness that is comparable to counterpart algorithms from the literature.

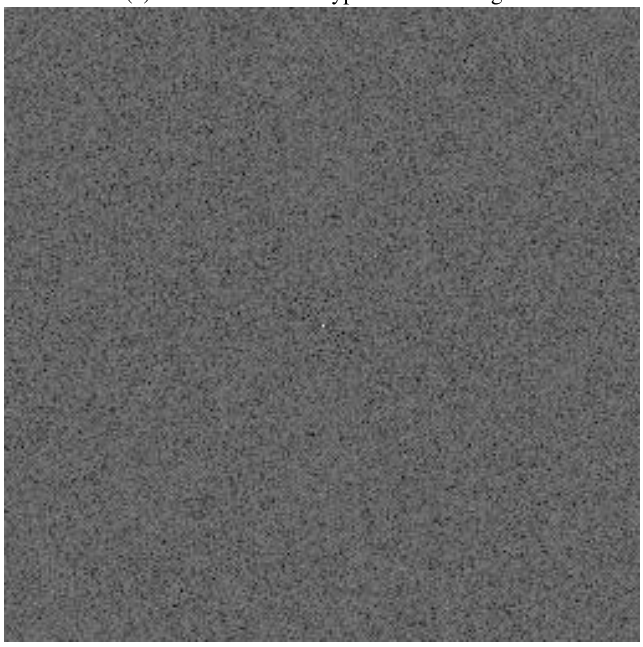
### G. FOURIER TRANSFORMATION ANALYSIS

Another technique that can be used to measure just how diffused the content of the encrypted image is, is the Discrete Fourier Transform (DFT). The DFT is a mathematical method that transforms any signal from the time domain into the frequency domain, effectively splitting it into many other periodic signals. This same technique can be applied to images, examining the shape of their DFT before and after encryption to check if any artifacts remain. The DFT of an image of $N \times N$ dimensions, where each pixel is represented as $f(i, j)$, into the frequency domain is calculated as follows:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-i2\pi\left(\frac{ki}{N} + \frac{li}{N}\right)}. \tag{17}$$



(a) DFT of the unencrypted House image.



(b) DFT of the encrypted House image.

**FIGURE 13.** Fourier transform of the plain and encrypted House images.

In the equation, $f(a, b)$ indicates the representation of the image in the spatial domain, and the exponential term is the basis function that relates to every point $F(k, l)$ in the Fourier space. The basis functions are trigonometric waves with increasing frequencies. Thiss indicates that $F(0, 0)$ is the DC-component of the image, and translates into average brightness, while $F(N-1, N-1)$ represents the highest frequency.

Figure 13 represents the effect of applying the DFT to plain and encrypted versions of the House image. It is clear that the DFT of the plain image in Fig. 13a contains broad streaks in the middle, forming a cross-like structure that represents the high concentration of pixels at dominant brightness. The connectedness of the cross also points to the natural source of the image, with most pixels possessing the same or similar levels of brightness. In the DFT of the encrypted image, seen in Fig. 13b, no such structure exists, and instead a uniform distribution of pixel values is displayed. No singular brightness or group of brightness is dominant with all pixel values possessing distributed brightness. This indicates that any and all identifiable characteristics in the original image were lost upon encryption.

### H. HISTOGRAM DEPENDENCY TESTS

Although section IV-A briefly discussed the differences between plain and encrypted images' color histograms, a more in-depth analysis of those differences is warranted to ensure that the proposed cryptosystem is secure against advanced cryptanalysis techniques as well. As such, a number of methods can be used to evaluate the linear dependency between a plain image's color histogram and the encrypted histogram. Dependency levels are generally calculated in the range $[-1, 1]$, where a value as close to 0 as possible is preferred, indicating a lack of dependence. Values of 1 and $-1$ both indicate strong dependence and inverse dependence respectively. The following 5 linear correlation evaluation techniques are used:

- Blomqvist's $\beta$ tests the correlation between 2 histogram distributions $x$ and $y$ as a medial correlation coefficient (given the medians of the distributions as $\overline{x}$ and $\overline{y}$). It is typically defined as follows:

$$\beta = \{(X - \overline{x})(Y - \overline{y}) > 0\} - \{(X - \overline{x})(Y - \overline{y}) < 0\}. \tag{18}$$

- Goodman-Kruskal's $\gamma$ measures the monotonic association of pairs of values in both histograms. Any pair of values in both histograms might either support or undermine the correlation. The metric is defined as:

$$\gamma = \frac{n_c - n_d}{n_c + n_d}. \tag{19}$$

- Kendall's $\tau$ evaluates the relation between the 2 histograms in relation to the size of the sample by employing a similar concept of pairs supporting or undermining the correlation. The correlation metric is defined as:

$$\tau = \frac{n_c - n_d}{\frac{n(n-1)}{2}}. \tag{20}$$

- The Spearman rank correlation $\rho$ test compares the position of an element in a sorted list of elements from the histogram to the mean rank value as follows:

$$\rho = \frac{\sum(R_{ix} - \overline{R}_x)(R_{iy} - \overline{R}_y)}{\sqrt{\sum(R_{ix} - \overline{R}_x)^2 \sum(R_{iy} - \overline{R}_y)^2}}. \tag{21}$$

**TABLE 11.** Histogram dependency tests for various images.

| Image | Color | $\beta$ (18) | $\gamma$ (19) | $\tau$ (20) | $\rho$ (21) | $r$ (22) |
|---|---|---|---|---|---|---|
| Lena | Red | 0.106949 | 0.105487 | 0.10211 | 0.146919 | 0.0403091 |
| | Green | −0.0637901 | −0.048005 | −0.0474069 | −0.0680551 | −0.113892 |
| | Blue | −0.0711512 | −0.0203785 | −0.0191742 | −0.0269984 | −0.0640328 |
| | Combined | 0.031497 | 0.0178605 | 0.0177536 | 0.0286113 | −0.0856135 |
| House | Red | 0.078125 | 0.0859684 | 0.0835202 | 0.11977 | 0.0807738 |
| | Green | 0 | 0.00676946 | 0.00669293 | −0.0309308 | 0.00661287 |
| | Blue | 0.00817277 | 0.00628381 | 0.00599914 | 0.00979905 | −0.0119619 |
| | Combined | 0.0589406 | 0.0284135 | 0.0282226 | 0.0422026 | −0.0264467 |
| Peppers | Red | −0.0749514 | −0.0592419 | −0.0579266 | −0.0841925 | −0.109961 |
| | Green | 0.11068 | 0.0434755 | 0.0430283 | 0.0580687 | −0.0104401 |
| | Blue | −0.0478183 | −0.0168001 | −0.0165313 | −0.0313893 | −0.0249592 |
| | Combined | 0.114405 | 0.0444589 | 0.0441494 | 0.0666489 | −0.0335524 |
| House2 | Red | −0.0828501 | 0.00773131 | 0.00760268 | 0.01731 | −0.0173777 |
| | Green | 0.031498 | 0.0345216 | 0.0341389 | 0.0468312 | −0.00162966 |
| | Blue | −0.0672985 | −0.0532556 | −0.0523536 | −0.079756 | −0.0866643 |
| | Combined | 0.0553416 | 0.00974127 | 0.00968057 | −0.0242042 | 0.0168709 |
| Mandrill | Red | −0.134401 | −0.110416 | −0.109278 | −0.161772 | −0.177218 |
| | Green | 0.03125 | 0.00581673 | 0.00571037 | 0.00217523 | −0.027869 |
| | Blue | 0.0238133 | 0.0319737 | 0.0316567 | −0.0215724 | 0.0437854 |
| | Combined | −0.0862752 | −0.0449685 | −0.0447107 | −0.0656276 | −0.109782 |
| Sailboat | Red | −0.0512881 | −0.00822554 | −0.00790485 | −0.00939942 | −0.0540556 |
| | Green | 0.0437483 | −0.0284361 | −0.0648114 | −0.0389433 | −0.0648114 |
| | Blue | 0.0318788 | 0.0201655 | 0.0199 | 0.0289103 | −0.0587563 |
| | Combined | −0.027615 | −0.0246806 | −0.0245334 | −0.0352627 | −0.118076 |
| Tree | Red | −0.0838465 | −0.0377731 | −0.0370505 | −0.0550062 | −0.0574733 |
| | Green | −0.119549 | −0.0573285 | −0.056508 | −0.0834913 | −0.0807761 |
| | Blue | 0.0869626 | −0.00989442 | −0.00950424 | −0.0166167 | −0.0523116 |
| | Combined | −0.0196469 | −0.0271243 | −0.0269496 | −0.0351106 | −0.0659488 |
| Girl | Red | −0.0299803 | −0.0317055 | −0.0266928 | −0.0330007 | −0.0341154 |
| | Green | 0.0343558 | 0.00591292 | 0.00488824 | 0.00681652 | 0.00303486 |
| | Blue | −0.137245 | −0.154145 | −0.124688 | −0.168672 | −0.139118 |
| | Combined | −0.0986249 | −0.100618 | −0.0961034 | −0.142417 | −0.119876 |

- The Pearson correlation $r$ connects components of each distribution to each distribution's mean. It can be calculated as:

$$r = \frac{\sum(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum(X_i - \overline{X})^2 \sum(Y_i - \overline{Y})^2}}. \tag{22}$$

The results of applying the 5 discussed tests to images in the test library are shown in Table 11. Most values are close to 0 in all tested color channels and images, indicating little to no dependence between the plain-text and cipher-text images used.

### I. DIFFERENTIAL ATTACK ANALYSIS

Differential attacks are a class of cryptographic attacks that attempt to retrieve the encryption key by modifying the source image slightly and analyzing the resulting encryption. An encryption scheme resistant to cryptographic attacks would diffuse any change in the plain-text throughout the cipher-text in an unpredictable manner, to avoid leaking any information to statistical attackers attempting to execute differential attacks. The Number of Pixel Changing Rate (NPCR) and the Unified Average Change Intensity (UACI) are the 2 measures typically employed to determine an image encryption system's strength against differential attacks. The NPCR is defined as:

$$NPCR = \frac{\sum_{i,j} D_{i,j}}{M \times N} \times 100, \tag{23}$$

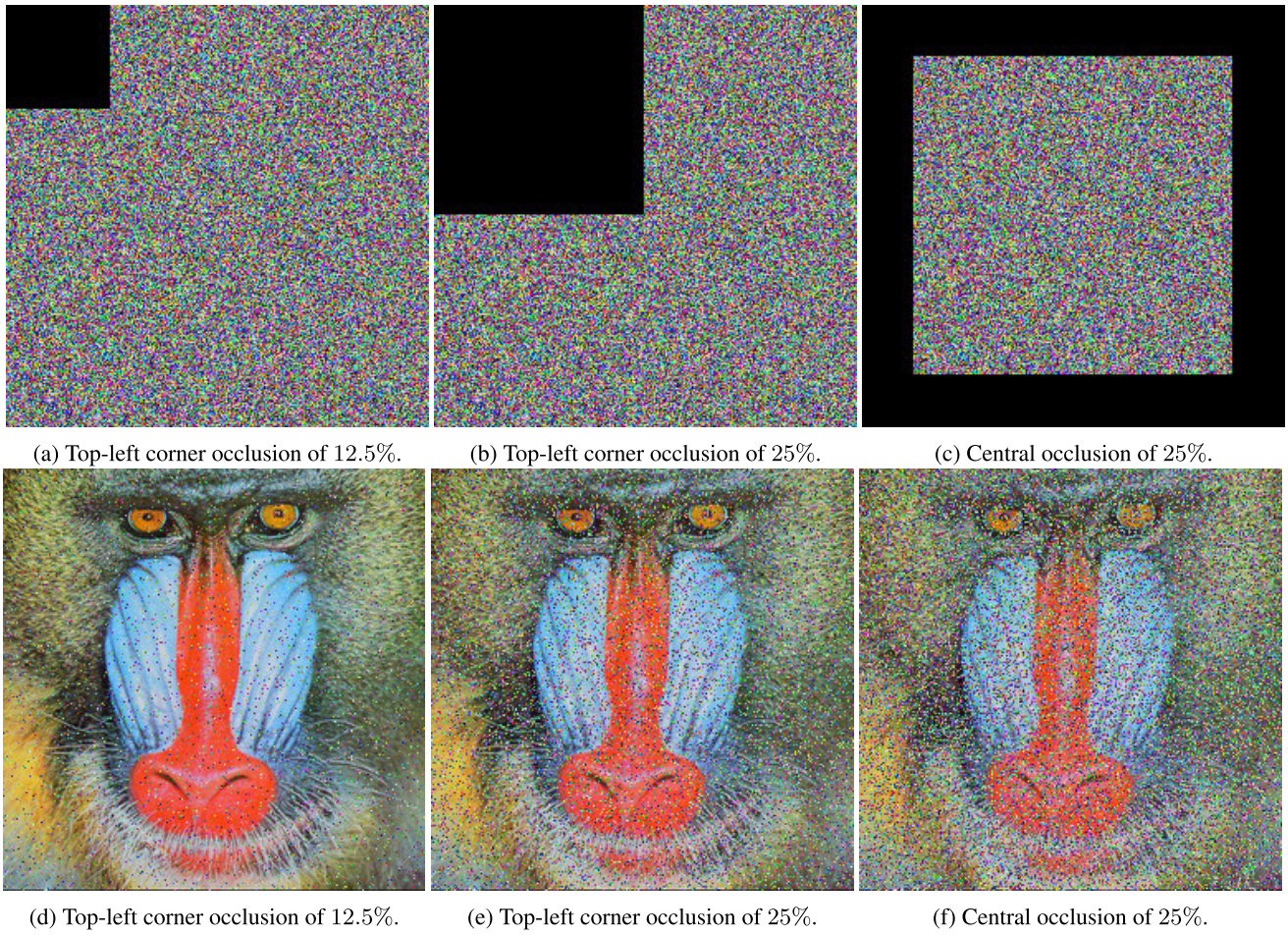
where $D_{i,j}$ is given by

$$D_{i,j} = \begin{cases} 0, & C_{1(i,j)} = C_{2(i,j)}, \\ 1, & C_{1(i,j)} \neq C_{2(i,j)}. \end{cases} \tag{24}$$

Simply put, the NPCR is a computation resulting in the number of differing pixels between a plain image and its encrypted version. The UACI is mathematically expressed as

$$UACI = \frac{1}{M \times N} \sum_{i,j} \frac{C_{1(i,j)} - C_{2(i,j)}}{255}, \tag{25}$$

where $C_{1(i,j)}$ and $C_{2(i,j)}$ are 2 images of dimensions $M \times N$. The UACI calculates the difference between the average intensity in an image and its encrypted version. The results for the NPCR and UACI calculation on the encrypted Lena image are provided in Tables 12 and 13. The ideal value of the NPCR is 100% although few encryption algorithms actually reach that value, and the ideal value of the UACI is 33.35%. The tables show that, while the NPCR values are quite close, the UACI values are somewhat lacking, but still comparable to those generated by the other schemes in similar literature on image encryption.



(a) Top-left corner occlusion of 12.5%. (b) Top-left corner occlusion of 25%. (c) Central occlusion of 25%.

(d) Top-left corner occlusion of 12.5%. (e) Top-left corner occlusion of 25%. (f) Central occlusion of 25%.

**FIGURE 14.** A visual representation of three different types of occlusion attacks, as well as the resultant decryption of those occlusions, clearly showing no recognizable pattern, as well as preservation of most of the visual data from the source image.

**TABLE 12.** NPCR values for the RGB channels of the Lena image.

| RGB | Proposed | [49] | [4] | [5] | [25] |
|---|---|---|---|---|---|
| R | 99.6109 | 99.592 | 99.5712 | 99.6231 | 99.58 |
| G | 99.6353 | 99.595 | 99.5758 | 99.614 | 99.56 |
| B | 99.5605 | 99.600 | 99.6094 | 99.5972 | 99.64 |

**TABLE 13.** UACI values for the RGB channels of the Lena image.

| RGB | Proposed | [49] | [4] | [5] | [25] |
|---|---|---|---|---|---|
| R | 32.8335 | 33.497 | 33.1056 | 32.8621 | 33.27 |
| G | 30.6768 | 33.325 | 30.5178 | 30.6466 | 33.36 |
| B | 27.7141 | 33.223 | 27.5385 | 27.5607 | 33.50 |

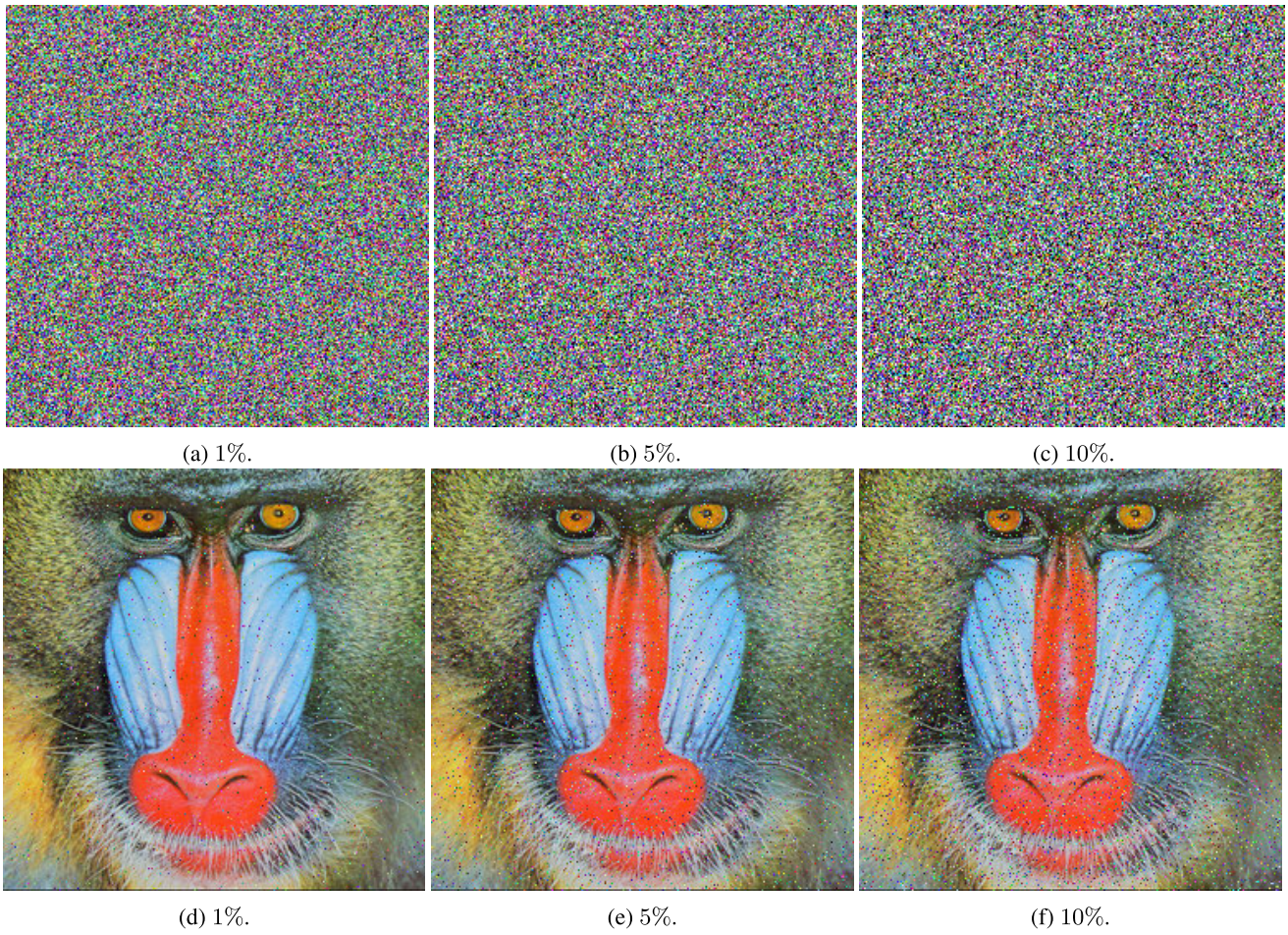
### J. OCCLUSION AND NOISE ATTACKS

Two other types of attacks, besides differential attacks, that strong encryption schemes should be capable of counteracting are occlusion and noise attacks. Poor encryption schemes may show patterns or recognizable artifacts upon encryption or decryption when controlled alterations are made to the plain-text or cipher-text respectively. These artifacts may reveal information about the scheme and the encryption key when statistical techniques are applied. Thus, it becomes of great importance to ensure that no such artifcats show when those attacks are made.

The two types rely on the same principle of altering the plain-text and examining the decryption to attempt to gain information. Occlusion attacks do this by blocking parts of the plain-text with known information and examining the propagation of the additional data in the encryption, while noise attacks add information by applying different types and levels of noise to the plain-text. Figures 14, 15, and 16 show the effects of occlusion, salt-and pepper-noise, and Gaussian noise attacks respectively. Figure 14 shows that, even when blocking up to 25% of the encryption, the decrypted image remains very much recognizable and the altered pixels are widely distributed and randomly scattered throughout the image. Figure 15 shows the same, with strong resistance to salt-and-pepper noise, while Fig. 16 shows a lack of any organized reaction to the input Gaussian noise, at all magnitudes. In general, the proposed algorithm is shown not to react in any organized manner when affected by occlusion, salt-and-pepper noise, and Gaussian noise, making it resistant to those attacks.

### K. KEY SPACE ANALYSIS

Key space analysis is a technique that gives an analysis of the number of possible keys that can be used in an encryption



(a) 1%. (b) 5%. (c) 10%.

(d) 1%. (e) 5%. (f) 10%.

**FIGURE 15.** A visual representation of salt-and-pepper attacks of different magnitudes on the Mandrill image's encryption; very little effect is seen, and most of the original visual data is preserved.

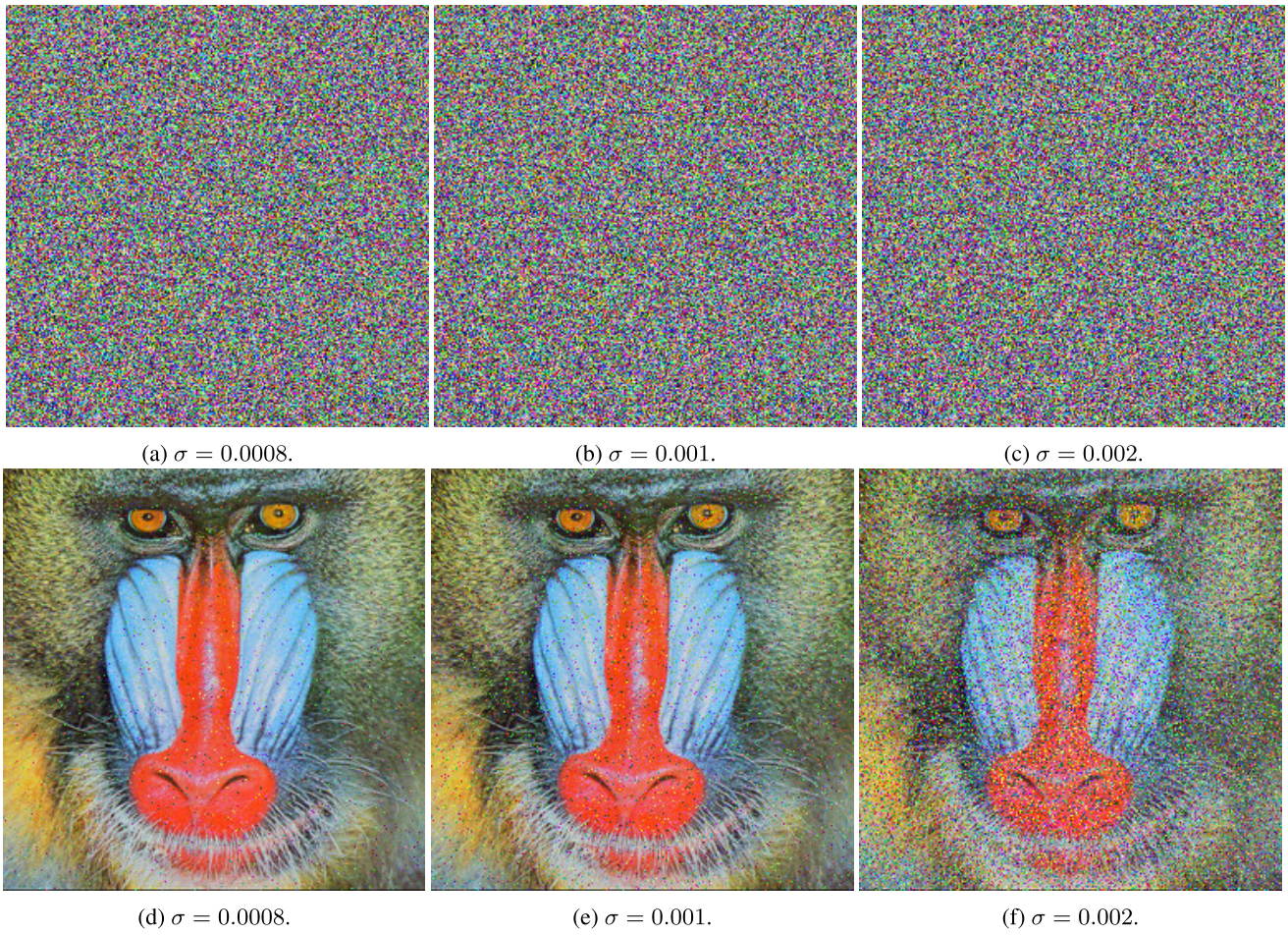
scheme. It is typically used to get an understanding of just how resistant an encryption scheme is to brute force attacks. A larger key space is preferable, making it more difficult for brute force attackers to cryptanalyze the required key. Unlike encryption schemes, the key space of the presented scheme is highly variable, since the number of required variables relies on the actual value of the control variables themselves. As such, the following key space analysis will consider the key space given the input variables used throughout section IV. Those values are as follows:

- Generation of 3 Composite Ant Fields, each requiring:
  - Field dimensions: 2 variables.
  - Number of sub-fields: 1 variable.
  - Number of ants per sub-field: 1 variable.
  - Starting position and direction of every ant in every sub-field. For the sample calculations, this was taken as 64 sub-fields, with 1 ant in each: $64 \times 1 \times 3$ variables.
  - Number of colors: 1 variable.
  - Unique rule for each ant in every field. For the sample calculation: $64 \times 1 \times 1$ variables.
  - Number of steps for each sub-field: $64 \times 1$.
- This totals 325 values per Composite Ant Field.
- With 3 fields (one field per RGB color channel), this presents a total of $325 \times 3 = 975$ variables.
- Generation of 3 Mersenne-Twister PRNG sequences (2 directly for encryption, and one for S-box construction), each requiring a single seed value.
- Cat Map specification requires 3 variables, the $a$ and $b$ values specific to the matrix, and the number of iterations $n$.
- In total, $975 + 3 + 3 = 981$ variables are required.

Assuming that each value is taken as a real value by the brute force computation, where each real value has a maximum machine precision of $10^{-16}$, the total key space becomes $10^{981\times16} = 10^{15696} \approx 2^{52141}$. This value is high enough to consider the scheme resistant to brute force attacks according to [30]. Table 14 displays the achieved key space values of the proposed system and that of others in similar literature. The proposed system actually exceeds the calculated key space for most of the provided algorithms showing very high resistance to brute force attacks.

### L. ENCRYPTION TIME ANALYSIS

Since the objective of any image encryption system is to be capable of encrypting images for actual real usage, the



(a) $\sigma = 0.0008$. (b) $\sigma = 0.001$. (c) $\sigma = 0.002$.

(d) $\sigma = 0.0008$. (e) $\sigma = 0.001$. (f) $\sigma = 0.002$.

**FIGURE 16.** A visual representation of Gaussian noise attacks of different standard deviations on the Mandrill image's encryption; very little organization in the generated changes is seen, and most of the original visual data is preserved.

**TABLE 14.** Key space values comparison.

| Scheme | Key space |
|---|---|
| Proposed | $10^{15696} \approx 2^{52141}$ |
| [6] | $10^{128}$ |
| [8] | $2^{299}$ |
| [17] | $2^{372}$ |
| [23] | $10^{169}$ |
| [4] | $2^{1658}$ |
| [5] | $2^{2551}$ |

**TABLE 15.** Execution time of the proposed image cryptosystem when applied to the 256 × 256 House image, with respect to the number of iterations of the Cat Map during encryption. MSE and IE values are also displayed.

| Cat Map Iterations | $t_{Enc}$ [s] | $MSE$ | $IE$ |
|---|---|---|---|
| 5 | 0.4270318 | 8372.11 | 7.99698 |
| 20 | 0.7223997 | 8410.69 | 7.99702 |
| 40 | 1.2314058 | 8403.23 | 7.99694 |
| 90 | 2.7974703 | 8398.63 | 7.9968 |
| 98 | 2.6152572 | 8358.49 | 7.9971 |

encryption time of the provided system should be within a certain acceptable range. Table 15 shows the encryption time for the proposed image encryption system when applied to an image of dimensions $256 \times 256$. Experimentation with the scheme's input variables revealed that the Cat Map's input iterations value had the greatest effect on the execution time. As such, Table 15 shows the effect on execution time at different values of the Cat Map iterations input, as well as the effect on 2 other metrics, the MSE (described in section IV-B) and Information Entropy (described in section IV-E) to give some perspective on the effect of the Cat Map on the encryption as a whole.

As described in Section II-C the Cat Map is periodic. The period for the matrix applied in the experiments performed in Table 15 is 196, so iteration values up to half the period were taken. Very little change in the MSE or Information Entropy was measured upon changing the iterations of the Cat Map, so the minimal value of 5 is taken for the purposes of comparison with other encryption schemes in Table 16. Table 16 shows that the proposed scheme operates relatively quickly, encrypting images of dimensions $256 \times 256$ in about 0.4 seconds. In addition, AES is used to encrypt an identically

**TABLE 16. Encryption time comparison of the Lena image of dimensions 256 × 256.**

| Scheme | Time [s] | Machine specifications (CPU and RAM) |
|---|---|---|
| Proposed | 0.427 | AMD® Ryzen™ 5600H Mobile 3.3 GHz, 16 GB |
| AES | 0.838 | AMD® Ryzen™ 5600H Mobile 3.3 GHz, 16 GB |
| [6] | 2.582 | 2.9 GHz Intel® Core™ i9, 32 GB |
| [17] | 1.425 | 2.9 GHz Intel® Core™ i9, 32 GB |
| [8] | 0.25 | N/A |
| [51] | 1.112 | 3.4 GHz Intel® Core™ i3, 4 GB |

sized image on the same machine, as shown in Table 16, to demonstrate that the proposed image encryption system is more efficient. While the proposed cryptosystem is somewhat slower than the fastest scheme presented, specifically that presented in [8], it is important to remember that encryption time is also often the function of the machine operating the encryption and its current state. The achieved encryption time of the proposed cryptosystem translates into an average encryption rate of 3.7 Mbps.

### M. THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY ANALYSIS

The United States National Institute of Standards and Technology (NIST) has a number of statistical documents and analysis tools devoted to cryptology. Once such tool is the SP 800-22 Statistical Test Suite, which is a set of analyses and tests designed to evaluate the efficacy of pseudo-random number generators. This can be useful when applied directly to the output of encryption produced by the proposed scheme. While not exactly suited for measuring the strength of the scheme's encryption, any encryption data that passes the NIST SP 800-22 tests would present enough chaotic behavior to be considered reliable encryption. Table 17 shows the results of all the suite's tests when applied to the image data from the encryption of the $256 \times 256$ sized Sailboat image. All of the table's values exceed the minimum pass requirement of 0.01, showing that the results of an encryption produced by the proposed scheme are encrypted strongly enough to act as an effective PRNG.

### N. S-BOX PERFORMANCE ANALYSIS

The generated S-box is one of 2 key elements in the proposed encryption scheme that act as sources for Shannon's property of confusion. As such, it becomes of some importance that the S-box be assessed on its efficiency independently. Five tests are usually applied to S-Boxes to ensure their efficacy:

- Non-linearity (NL), which measures the number of bits in a Boolean function's truth table that needs to be changed, for it to reach the nearest affine function.
- Linear approximation probability (LAP), which indicates the probability for an S-box to be biased.
- Differential approximation probability (DAP), which displays the effect of changes in the input on the output.
- Bit independence criterion (BIC), which measures the relationship between encryption technique and resulting repeated patterns in the output.
- Strict avalanche criterion (SAC), which measures the rate of change in the output relative to the rate of change in the input on a bit-by-bit basis.

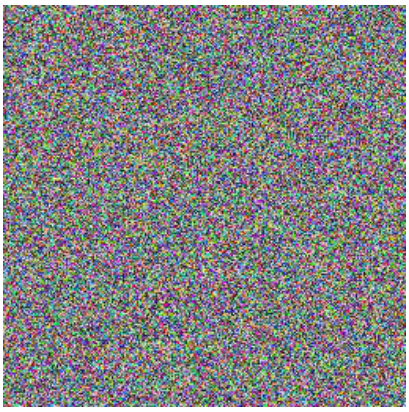
**TABLE 17. NIST analysis of the image-data bitstream from the encrypted Sailboat image of size 256 × 256.**

| Test Name | $p$-value | Remarks |
|---|---|---|
| Frequency | 0.959301 | Success |
| Block Frequency | 0.872927 | Success |
| Runs | 0.122278 | Success |
| Longest run of ones | 0.482247 | Success |
| Rank | 0.605649 | Success |
| FFT | 0.084990 | Success |
| Non overlapping T.M. (000000001) | 0.987797 | Success |
| Overlapping T.M. | 0.211766 | Success |
| Maurer's Universal | 0.826660 | Success |
| Linear complexity | 0.108833 | Success |
| Serial 1 | 0.328055 | Success |
| Serial 2 | 0.578563 | Success |
| Approx. entropy | 0.100840 | Success |
| Cum. sums forward | 0.505831 | Success |
| Cum. sums reverse | 0.548599 | Success |
| Random ex. 1 | 0.943580 | Success |
| Random ex. 2 | 0.462019 | Success |
| Random ex. 3 | 0.686487 | Success |
| Random ex. 4 | 0.288324 | Success |
| Random ex. 5 | 0.152853 | Success |
| Random ex. 6 | 0.123802 | Success |
| Random ex. 7 | 0.251100 | Success |
| Random ex. 8 | 0.846973 | Success |
| Random ex. var. 1 | 0.669353 | Success |
| Random ex. var. 2 | 0.869015 | Success |
| Random ex. var. 3 | 0.670038 | Success |
| Random ex. var. 4 | 0.715614 | Success |
| Random ex. var. 5 | 0.990818 | Success |
| Random ex. var. 6 | 0.906512 | Success |
| Random ex. var. 7 | 0.853014 | Success |
| Random ex. var. 8 | 0.749789 | Success |
| Random ex. var. 9 | 0.769176 | Success |
| Random ex. var. 10 | 0.437285 | Success |
| Random ex. var. 11 | 0.583596 | Success |
| Random ex. var. 12 | 0.871108 | Success |
| Random ex. var. 13 | 0.871108 | Success |
| Random ex. var 14 | 0.777986 | Success |
| Random ex. var. 15 | 0.962639 | Success |
| Random ex. var. 16 | 0.656142 | Success |
| Random ex. var. 17 | 0.342446 | Success |
| Random ex. var. 18 | 0.231160 | Success |

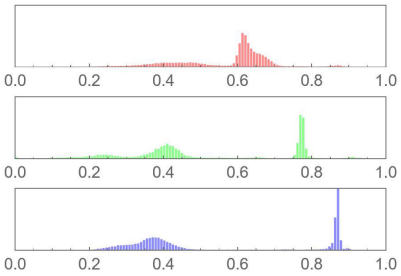
**TABLE 18. Comparison between the proposed S-box (generated using Mersenne-Twister *Seed* = 456) and those provided in the literature.**

| S-Box | NL | SAC | BIC | LAP | DAP |
|---|---|---|---|---|---|
| Ideal | 112 | 0.5 | 112 | 0.0625 | 0.0156 |
| Proposed | 108 | 0.494141 | 108 | 0.078125 | 0.015625 |
| [3] | 112 | 0.4998 | 112 | 0.0625 | 0.0156 |
| [4] MT | 108 | 0.503662 | 92 | 0.140625 | 0.015625 |
| [4] OSSL | 108 | 0.499023 | 112 | 0.0625 | 0.015625 |
| [4] IMKL | 108 | 0.499268 | 104 | 0.09375 | 0.015625 |
| [5] SNM | 106 | 0.499268 | 104 | 0.09375 | 0.015625 |
| [5] HC 4D | 108 | 0.500977 | 108 | 0.078125 | 0.015625 |
| [5] HC 7D | 108 | 0.506592 | 108 | 0.078125 | 0.015625 |
| [7] | 106 | 0.47266 | 68 | 0.23438 | 0.015625 |
| [50] | 107 | 0.497 | 103.5 | 0.1560 | 0.039 |

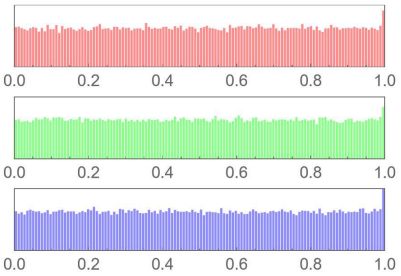
(a) Plain House image.



(b) Encrypted House image.



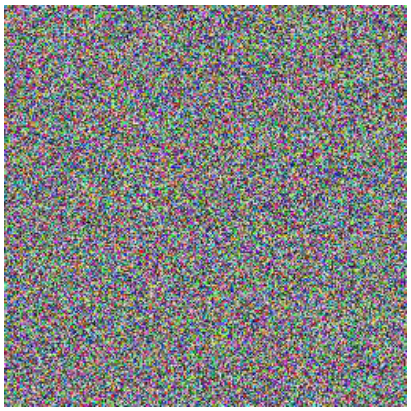

(c) Histogram of plain House image.



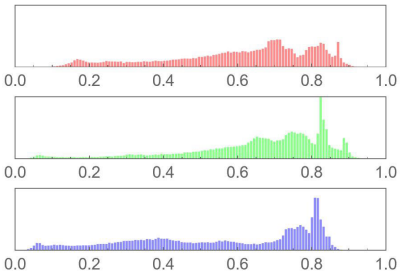

(d) Histogram of encrypted House image.

**FIGURE 17.** Plain and encrypted versions of the House image, and their respective histogram plots.

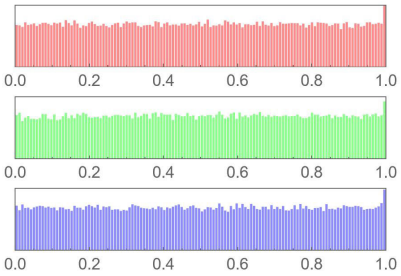
(a) Plain House2 image.



(b) Encrypted House2 image.




(c) Histogram of plain House2 image.

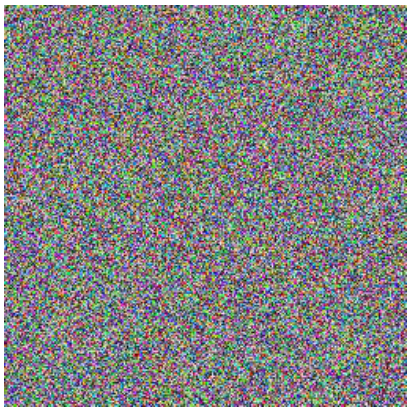



(d) Histogram of encrypted House2 image.

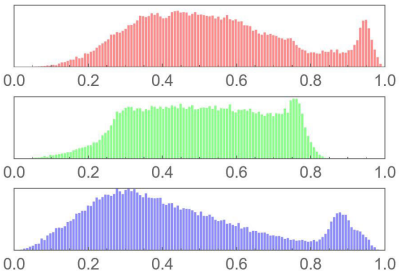
**FIGURE 18.** Plain and encrypted versions of the House2 image, and their respective histogram plots.

The S-box used in the proposed scheme is generated dynamically using the Mersenne-Twister PRNG. A long sequence of values are taken from the PRNG and used to generate a number of S-boxes. Those S-boxes are evaluated using the 5 tests mentioned above, and the test scoring the clothest to the optimal values is adopted. The results

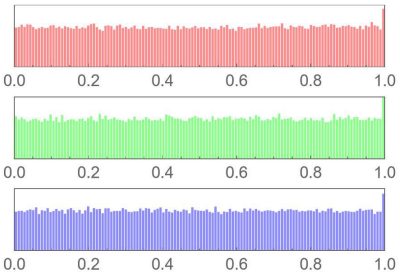
(a) Plain Mandrill image.



(b) Encrypted Mandrill image.



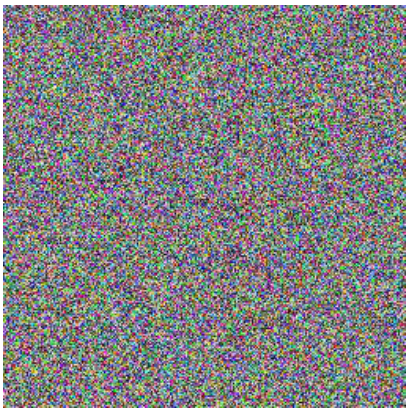

(c) Histogram of plain Mandrill image.



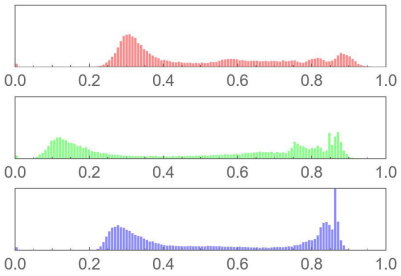

(d) Histogram of encrypted Mandrill image.

**FIGURE 19.** Plain and encrypted versions of the Mandrill image, and their respective histogram plots.

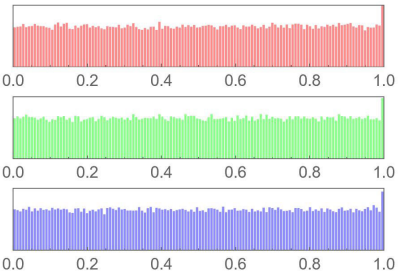
(a) Plain Tree image.



(b) Encrypted Tree image.




(c) Histogram of plain Tree image.

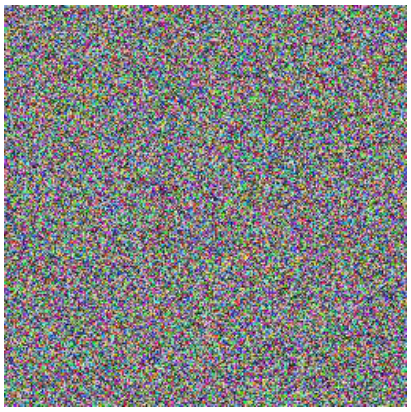



(d) Histogram of encrypted Tree image.

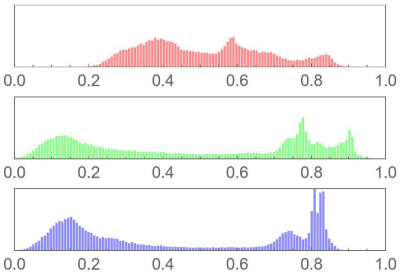
**FIGURE 20.** Plain and encrypted versions of the Tree image, and their respective histogram plots.

for one such S-box are shown in Table 18, where they are also compared with the optimal values for each test, as well as those of S-boxes used in similar literature. While not exactly ideal, the values produced by the S-box are at least somewhat comparable with those produced by similar literature. Additionally, the S-box is generated dynamically

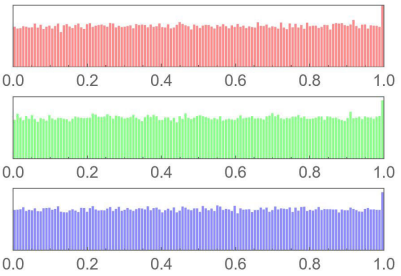
(a) Plain Sailboat image.



(b) Encrypted Sailboat image.



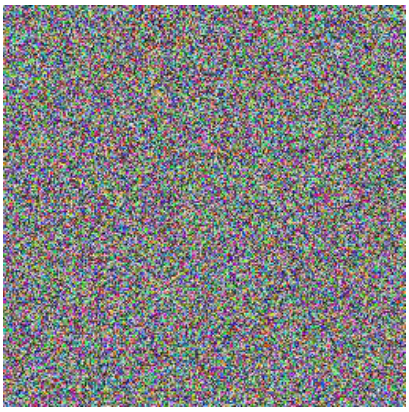

(c) Histogram of plain Sailboat image.



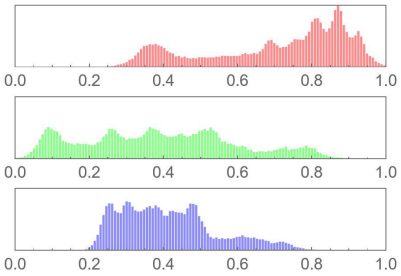

(d) Histogram of encrypted Sailboat image.

**FIGURE 21.** Plain and encrypted versions of the Sailboat image, and their respective histogram plots.

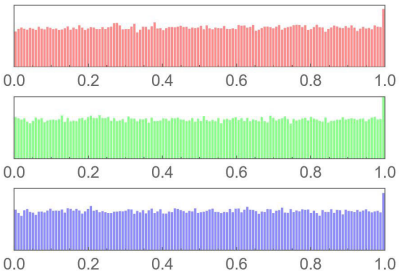
(a) Plain Lena image.



(b) Encrypted Lena image.



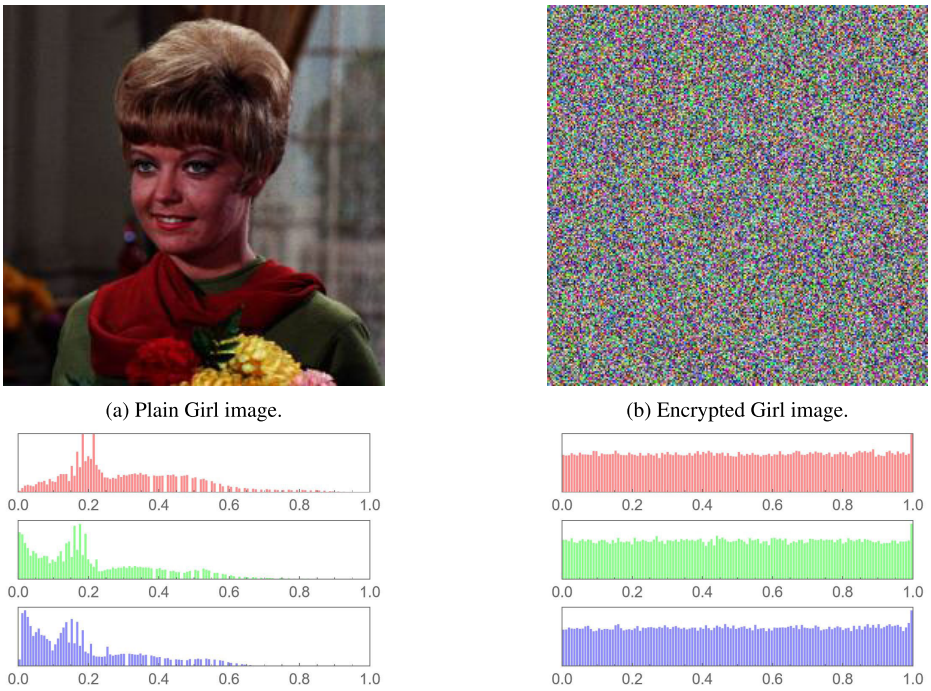

(c) Histogram of plain Lena image.




(d) Histogram of encrypted Lena image.

**FIGURE 22.** Plain and encrypted versions of the Lena image, and their respective histogram plots.

and with self-revision, so adjustment of the pseudo-random sequence produced by the Mersenne-Twister generator may produce even better outputs. In any case, this method of S-box generation generally foregoes the intricacies of S-box design in favor of randomness in the contents, making it perform somewhat poorly on some of the tests in Table 18.




(a) Plain Girl image.

(b) Encrypted Girl image.

(c) Histogram of plain Girl image.

(d) Histogram of encrypted Girl image.

**FIGURE 23.** Plain and encrypted versions of the Girl image, and their respective histogram plots.

## V. CONCLUSIONS AND FUTURE WORK

In conclusion, this paper has presented a novel five-stage image encryption algorithm that provides an efficient and secure mechanism for real-time image encryption applications. The proposed technique incorporates a multitude of chaotic and randomness-inducing mathematical operations, such as Langton's Ant, Mersenne Twister PRNG, S-box substitution, and Arnold's Cat map for instigating confusion and diffusion. Security analyses indicate the algorithm's resilience to various attacks including brute-force, statistical, differential, as well as various other forms of attacks. Additionally, it has passed the stringent NIST SP 800-22 statistical test suite. Compared to existing image encryption schemes, the proposed method exhibits lower computational complexity allowing real-time encryption of HD images. The security evaluation and performance results validate the effectiveness of the proposed encryption stages and their combination. This demonstrates the algorithm's suitability for practical secure image transmission over open networks. As future work, hardware implementations of the algorithm on FPGA and ASIC can be explored to enable very high-speed real-time encryption. The security can be further analyzed under a wider array of attacks. Overall, this work adds a new high-speed image encryption scheme to the literature providing an optimal balance between security and the measured real-time performance.

## APPENDIX
## VISUAL IMAGE ANALYSIS REPOSITORY

See Figures 17–23.

## REFERENCES

[1] R. I. Abdelfatah, "A color image authenticated encryption using conic curve and Mersenne twister," *Multimedia Tools Appl.*, vol. 79, nos. 33–34, pp. 24731–24756, Sep. 2020.
[2] I. Q. Abduljaleel, S. A. Abdul-Ghani, and H. Z. Naji, "An image of encryption algorithm using graph theory and speech signal key generation," *J. Phys., Conf. Ser.*, vol. 1804, no. 1, Feb. 2021, Art. no. 012005.
[3] J. A. Aboytes-González, J. S. Murguía, M. Mejía-Carlos, H. González-Aguilar, and M. T. Ramírez-Torres, "Design of a strong s-box based on a matrix approach," *Nonlinear Dyn.*, vol. 94, no. 3, pp. 2003–2012, 2018.
[4] W. Alexan, N. Alexan, and M. Gabr, "Multiple-layer image encryption utilizing fractional-order Chen hyperchaotic map and cryptographically secure PRNGs," *Fractal Fractional*, vol. 7, no. 4, p. 287, Mar. 2023.
[5] W. Alexan, Y.-L. Chen, L. Y. Por, and M. Gabr, "Hyperchaotic maps and the single neuron model: A novel framework for chaos-based image encryption," *Symmetry*, vol. 15, no. 5, p. 1081, May 2023.
[6] W. Alexan, M. ElBeltagy, and A. Aboshousha, "RGB image encryption through cellular automata, S-box and the Lorenz system," *Symmetry*, vol. 14, no. 3, p. 443, Feb. 2022.
[7] W. Alexan, M. Gabr, E. Mamdouh, R. Elias, and A. Aboshousha, "Color image cryptosystem based on sine chaotic map, 4D Chen hyperchaotic map of fractional-order and hybrid DNA coding," *IEEE Access*, vol. 11, pp. 54928–54956, 2023.
[8] T. S. Ali and R. Ali, "A new chaos based color image encryption algorithm using permutation substitution and Boolean operation," *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 19853–19873, Jul. 2020.
[9] D. M. Alsaffar, A. S. Almutiri, B. Alqahtani, R. M. Alamri, H. F. Alqahtani, N. N. Alqahtani, G. M. Alshammari, and A. A. Ali, "Image encryption based on AES and RSA algorithms," in *Proc. 3rd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, Mar. 2020, pp. 1–5.
[10] P. N. Andono and D. R. I. M. Setiadi, "Improved pixel and bit confusion-diffusion based on mixed chaos and hash operation for image encryption," *IEEE Access*, vol. 10, pp. 115143–115156, 2022.
[11] N. A. Azam, J. Zhu, U. Hayat, and A. Shurbevski, "Towards provably secure asymmetric image encryption schemes," *Inf. Sci.*, vol. 631, pp. 164–184, Jun. 2023.
[12] X. Chai, J. Fu, Z. Gan, Y. Lu, and Y. Zhang, "An image encryption scheme based on multi-objective optimization and block compressed sensing," *Nonlinear Dyn.*, vol. 108, no. 3, pp. 2671–2704, May 2022.

[13] T. Chuman, W. Sirichotedumrong, and H. Kiya, "Encryption-then-compression systems using grayscale-based image encryption for JPEG images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1515–1525, Jun. 2019.

[14] M. T. Elkandoz and W. Alexan, "Image encryption based on a combination of multiple chaotic maps," *Multimedia Tools Appl.*, vol. 81, no. 18, pp. 25497–25518, Jul. 2022.

[15] M. Gabr, W. Alexan, K. Moussa, B. Maged, and A. Mezar, "Multi-stage RGB image encryption," in *Proc. Int. Telecommun. Conf. (ITC-Egypt)*, Jul. 2022, pp. 1–6.

[16] M. Gabr, H. Younis, M. Ibrahim, S. Alajmy, and W. Alexan, "Visual data enciphering via DNA encoding, S-box, and tent mapping," in *Proc. IEEE 5th Int. Conf. Image Process. Appl. Syst. (IPAS)*, vol. 5, Dec. 2022, pp. 1–6.

[17] M. Gabr, H. Younis, M. Ibrahim, S. Alajmy, I. Khalid, E. Azab, R. Elias, and W. Alexan, "Application of DNA coding, the Lorenz differential equations and a variation of the logistic map in a multi-stage cryptosystem," *Symmetry*, vol. 14, no. 12, p. 2559, Dec. 2022.

[18] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li, and X. Tang, "EFR-CSTP: Encryption for face recognition based on the chaos and semi-tensor product theory," *Inf. Sci.*, vol. 621, pp. 766–781, Apr. 2023.

[19] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li, C. Wang, and X. Tang, "Asynchronous updating Boolean network encryption algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 4388–4400, Aug. 2023.

[20] S. Gao, R. Wu, X. Wang, J. Wang, Q. Li, C. Wang, and X. Tang, "A 3D model encryption scheme based on a cascaded chaotic system," *Signal Process.*, vol. 202, Jan. 2023, Art. no. 108745.

[21] H. Ghazanfaripour and A. Broumandnia, "Designing a digital image encryption scheme using chaotic maps with prime modular," *Opt. Laser Technol.*, vol. 131, Nov. 2020, Art. no. 106339.

[22] A. Hafsa, A. Sghaier, J. Malek, and M. Machhout, "Image encryption method based on improved ECC and modified AES algorithm," *Multimedia Tools Appl.*, vol. 80, no. 13, pp. 19769–19801, May 2021.

[23] E. Hasanzadeh and M. Yaghoobi, "A novel color image encryption algorithm based on substitution box and hyper-chaotic system with fractal keys," *Multimedia Tools Appl.*, vol. 79, pp. 7279–7297, Dec. 2019.

[24] Z. Hua, Z. Zhu, Y. Chen, and Y. Li, "Color image encryption using orthogonal Latin squares and a new 2D chaotic system," *Nonlinear Dyn.*, vol. 104, no. 4, pp. 4505–4522, Jun. 2021.

[25] M. Khan and F. Masood, "A novel chaotic image encryption technique based on multiple discrete dynamical maps," *Multimedia Tools Appl.*, vol. 78, no. 18, pp. 26203–26222, Sep. 2019.

[26] K. Kondou and M. Noda, "Uniform Latin square interleaving for correcting two-dimensional burst errors," *IEEE Trans. Magn.*, vol. 41, no. 10, pp. 2962–2964, Oct. 2005.

[27] M. Kumari and S. Gupta, "Performance comparison between chaos and quantum-chaos based image encryption techniques," *Multimedia Tools Appl.*, vol. 80, no. 24, pp. 33213–33255, Oct. 2021.

[28] C. G. Langton, "Studying artificial life with cellular automata," *Phys. D, Nonlinear Phenomena*, vol. 22, nos. 1–3, pp. 120–149, Oct. 1986.

[29] L. Li, "Application of data image encryption technology in computer network information security," *Math. Problems Eng.*, vol. 2022, pp. 1–7, Jul. 2022.

[30] H. Liu, X. Wang, and A. Kadir, "Chaos-based color image encryption using one-time keys and Choquet fuzzy integral," *Int. J. Nonlinear Sci. Numer. Simul.*, vol. 15, no. 1, pp. 1–10, Feb. 2014.

[31] P. Mahajan and A. Sachdeva, "A study of encryption algorithms AES, DES and RSA for security," *Global J. Comput. Sci. Technol.*, vol. 13, no. 15, pp. 15–22, 2013.

[32] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, Jan. 1998.

[33] G. Mélard, "On the accuracy of statistical procedures in Microsoft excel 2010," *Comput. Statist.*, vol. 29, no. 5, pp. 1095–1128, Oct. 2014.

[34] S. M. Mohamed, W. S. Sayed, A. H. Madian, A. G. Radwan, and L. A. Said, "An encryption application and FPGA realization of a fractional memristive chaotic system," *Electronics*, vol. 12, no. 5, p. 1219, Mar. 2023.

[35] R. Nithya and D. Dhanasekaran, "Novel dominant color subband image encryption in visual sensor network for smart military surveillance system," *Traitement du Signal*, vol. 39, no. 3, pp. 951–960, Jun. 2022.

[36] S. Patel, K. P. Bharath, and M. R. Kumar, "Symmetric keys image encryption and decryption using 3D chaotic maps with DNA encoding technique," *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 31739–31757, Nov. 2020.

[37] D. R. I. M. Setiadi, E. H. Rachmawanto, and R. Zulfiningrum, "Medical image cryptosystem using dynamic Josephus sequence and chaotic-hash scrambling," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 9, pp. 6818–6828, Oct. 2022.

[38] A. Romero-Arellano, E. Moya-Albor, J. Brieva, I. Cruz-Aceves, J. G. Avina-Cervantes, M. A. Hernandez-Gonzalez, and L. M. Lopez-Montero, "Image encryption and decryption system through a hybrid approach using the jigsaw transform and Langton's ant applied to retinal fundus images," *Axioms*, vol. 10, no. 3, p. 215, Sep. 2021.

[39] M. Route, "Radio-flaring ultracool dwarf population synthesis," *Astrophys. J.*, vol. 845, no. 1, p. 66, Aug. 2017.

[40] M. H. Saracevic, S. Z. Adamovic, V. A. MiS. Patel, K. P. Bharath, and R. Kumarkovic, M. Elhoseny, N. D. Macek, M. M. Selim, and K. Shankar, "Data encryption for Internet of Things applications based on Catalan objects and two combinatorial structures," *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 819–830, Jun. 2021.

[41] Y. Shen, J. Huang, L. Chen, T. Wen, T. Li, and G. Zhang, "Fast and secure image encryption algorithm with simultaneous shuffling and diffusion based on a time-delayed combinatorial hyperchaos map," *Entropy*, vol. 25, no. 5, p. 753, May 2023.

[42] R Simard, "Testu01: AC library for empirical testing of random number generators P. L'Ecuyer," *Les Cahiers du GERAD ISSN*, vol. 711, p. 2440, Nov. 2006.

[43] W. Sirichotedumrong, Y. Kinoshita, and H. Kiya, "Privacy-preserving deep neural networks using pixel-based image encryption without common security keys," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2019, pp. 1756–1761.

[44] A. ur Rehman, X. Liao, and H. Wang, "An innovative technique for image encryption using tri-partite graph and chaotic maps," *Multimedia Tools Appl.*, vol. 80, no. 14, pp. 21979–22005, Jun. 2021.

[45] Y. Wang, C. Wu, S. Kang, Q. Wang, and V. I. Mikulovich, "Multi-channel chaotic encryption algorithm for color image based on DNA coding," *Multimedia Tools Appl.*, vol. 79, nos. 25–26, pp. 18317–18342, Jul. 2020.

[46] E. Winarno, K. Nugroho, P. W. Adi, and D. R. I. M. Setiadi, "Combined interleaved pattern to improve confusion-diffusion image encryption based on hyperchaotic system," *IEEE Access*, vol. 11, pp. 69005–69021, 2023.

[47] F. Yang, J. Mou, K. Sun, Y. Cao, and J. Jin, "Color image compression-encryption algorithm based on fractional-order memristor chaotic circuit," *IEEE Access*, vol. 7, pp. 58751–58763, 2019.

[48] S. Yasser, A. Hesham, M. Hassan, and W. Alexan, "AES-secured bit-cycling steganography in sliced 3D images," in *Proc. Int. Conf. Innov. Trends Commun. Comput. Eng. (ITCE)*, Feb. 2020, pp. 227–231.

[49] I. Younas and M. Khan, "A new efficient digital image encryption based on inverse left almost semi group and Lorenz chaotic system," *Entropy*, vol. 20, no. 12, p. 913, Nov. 2018.

[50] A. Zahid, M. Arshad, and M. Ahmad, "A novel construction of efficient substitution-boxes using cubic fractional transformation," *Entropy*, vol. 21, no. 3, p. 245, Mar. 2019.

[51] W. Zhang, S. Wang, W. Han, H. Yu, and Z. Zhu, "An image encryption algorithm based on random Hamiltonian path," *Entropy*, vol. 22, no. 1, p. 73, Jan. 2020.

[52] X. Zhang, M. Liu, and J. Tian, "Multiple-image encryption algorithm based on Sarrus rule and 3D Fibonacci matrix," *Phys. Scripta*, vol. 98, no. 5, May 2023, Art. no. 055208.

[53] X. Zhang and X. Wang, "Multiple-image encryption algorithm based on DNA encoding and chaotic system," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 7841–7869, Mar. 2019.

[54] Y. Zhang, "Test and verification of AES used for image encryption," *3D Res.*, vol. 9, no. 1, pp. 1–27, Mar. 2018.

[55] S. Zhou, X. Wang, Y. Zhang, B. Ge, M. Wang, and S. Gao, "A novel image encryption cryptosystem based on true random numbers and chaotic systems," *Multimedia Syst.*, vol. 28, pp. 95–112, May 2021.

[56] B. Zolfaghari and K. Bibak, "Combinatorial cryptography and Latin squares," in *Perfect Secrecy IoT: A Hybrid Combinatorial-Boolean Approach*. Cham, Switzerland: Springer, 2022, pp. 37–55.

**WASSIM ALEXAN** (Senior Member, IEEE) was born in Alexandria, Egypt, in 1987. He received the B.Sc., M.Sc., and Ph.D. degrees in communications engineering and the M.B.A. degree from German University in Cairo (GUC), Egypt, in 2010, 2012, 2017, and 2019, respectively.

He was with the Mathematics Department, from 2010 to 2017. Since 2017, he has been an Assistant Professor with the Faculty of Information Engineering and Technology, GUC, teaching various courses in relation to wireless communications, modulation and coding, information theory, digital logic design, circuit theory, and mathematics. He has been an Adjunct Assistant Professor with the Mathematics Department, German International University (GIU), New Administrative Capital, Egypt, since 2019. He is the author or coauthor of more than 70 journal articles and conference papers. His research interests lie in the fields of wireless communications, information security, image, and signal processing.

Dr. Alexan is a member of the ACM and has been granted the Best Paper Award at the 19th and 26th IEEE Conference on Signal Processing Algorithms, Architectures, Arrangements and Applications (SPA'2015 and SPA'2023, respectively), Poznan, Poland; the AEG Writer of the Year Award from the American University in Cairo (AUC), Egypt, in 2019; and the Best Poster Award at the 37th IEEE National Radio Science Conference, Cairo, Egypt, in 2020.

**YOUSEF KORAYEM** (Student Member, IEEE) was born in Cairo, Egypt, in 2001. He received the B.Sc. degree in computer science and engineering from German University in Cairo (GUC), Egypt, in 2023. His research interests lie in the fields of information security and image processing. He has a few IEEE conference papers published.

**MOHAMED GABR** (Member, IEEE) was born in Cairo, Egypt, in 1989. He received the B.Sc., M.Sc., and Ph.D. degrees in computer science and engineering from German University in Cairo (GUC), Egypt, in 2011, 2013, and 2023, respectively.

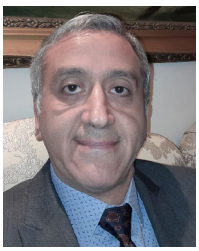
He has been with the Computer Science and Engineering Department, since 2011. He is teaching various courses in relation to computer vision, artificial intelligence, compilers, theory of computation, and computer graphics. He is the author or coauthor of various journal articles and conference papers. His research interests lie in the fields of computer vision and information security.

**MINAR EL-AASSER** (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from German University in Cairo (GUC), in 2008, 2013, and 2021, respectively.

She has been a Researcher and a Lecturer Assistant, since 2008. She is currently an Assistant Professor with Networking Department, Faculty of Information and Engineering Technology, GUC. She has several publications recognized in both national and international conferences and journals. Her research interests lie in the areas of modeling and simulation of trending networking technologies. After attaining her Ph.D., she focused her research on information, networks, and cyber security, modeling their threats on key network technologies, such as V2V, LPWANs, and SDNs.

**ENGY ALY MAHER** (Senior Member, IEEE) was born in Cairo, Egypt, in 1990. She received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees in information engineering and technology (IET) from Communications Department, German University in Cairo (GUC), Cairo, in 2012, 2013, and 2019, respectively. From 2012 to 2019, she was a Teaching and a Research Assistant with Electrical Engineering Department, GUC, where since 2019, she has been an Assistant Professor and a Researcher. Her research interests lie in the areas of mobile and wireless communications, resource allocation, power control, radio resource management, spectral efficiency, and signal processing. Her work is recognized by several publications at both national and international conferences and journals in this field.

**DINA EL-DAMAK** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering and computer science from MIT, in 2012 and 2015, respectively, under the supervision of Prof. Anantha Chandrakasan. After receiving her Ph.D. degree, she spent one year as a Postdoctoral Associate with MIT Microsystems Technology Laboratories. From September 2020 to August 2022, she was an Assistant Professor at Zewail City, Egypt. From August 2016 to August 2020, she was an Assistant Professor of electrical and computer engineering with the University of Southern California. She has been an Assistant Professor with German University in Cairo (GUC), since September 2022.She has published in major conferences and journals, including ISSCC, VLSI Circuits, and Technology, IEDM, IEEE JOURNAL OF SOLID-STATE CIRCUITS, *Nano Letters*, and *Nature Biomedical Engineering*. She holds three U.S. patents. She was a recipient of the Texas Instruments Graduate Woman's Fellowship for Leadership in Microelectronics at MIT. She was selected as one of the women rising stars in electrical engineering and computer sciences (EECS) at UC Berkeley, in 2014. She is one of the organizers of the 2023 IEEE Power Supply on Chip (PwrSoC) Conference, Germany.



**AMR ABOSHOUSHA** was born in Cairo, Egypt, in 1967. He received the B.Sc. degree (Hons.) in physics and the M.Sc. degree in theoretical physics from the Science Faculty, Cairo University, in 1988 and 1995, respectively, and the Dr. Rer. Nat. degree in theoretical physics from Friedrich Alexander University Erlangen-Nuremberg, Germany.

He started to work as a Physics Lecturer with the Physics Department, Science Faculty, Cairo University, from 2002 to 2010. During this period, he instructed and supervised various physics courses, such as thermodynamics, electromagnetism, statistical mechanics, computational physics, and optical communications. From 2004 to 2005, he was a Reviewer of the Physics Exam Committee of the High School Certificate, Ministry of Education. Since 2010, he has been with the Physics Department, Faculty of Engineering, German University in Cairo (GUC), as an Assistant Professor, teaching different physics-related courses to engineering, biotechnology, and pharmacy students. His research interests include applications of theoretical physics models in different areas, such as information technology and biophysics.