applied sciences

MDPI

Article

Enhancing Image Encryption with the Kronecker xor Product, the Hill Cipher, and the Sigmoid Logistic Map

Dani Elias Mfungo, Xianping Fu *, Xingyuan Wang and Yongjin Xian

School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; danimfungo@gmail.com (D.E.M.); matxyj@163.com (Y.X.)
* Correspondence: fxp@dlmu.edu.cn

Abstract: In today's digital age, it is crucial to secure the flow of information to protect data and information from being hacked during transmission or storage. To address this need, we present a new image encryption technique that combines the Kronecker xor product, Hill cipher, and sigmoid logistic Map. Our proposed algorithm begins by shifting the values in each row of the state matrix to the left by a predetermined number of positions, then encrypting the resulting image using the Hill Cipher. The top value of each odd or even column is used to perform an xor operation with all values in the corresponding even or odd column, excluding the top value. The resulting image is then diffused using a sigmoid logistic map and subjected to the Kronecker xor product operation among the pixels to create a secure image. The image is then diffused again with other keys from the sigmoid logistic map for the final product. We compared our proposed method to recent work and found it to be safe and efficient in terms of performance after conducting statistical analysis, differential attack analysis, brute force attack analysis, and information entropy analysis. The results demonstrate that our proposed method is robust, lightweight, and fast in performance, meets the requirements for encryption and decryption, and is resistant to various attacks.

Keywords: image encryption; Hill cipher; chaotic map; logistic map; Kronecker product

Citation: Mfungo, D.E.; Fu, X.; Wang, X.; Xian, Y. Enhancing Image Encryption with the Kronecker xor Product, the Hill Cipher, and the Sigmoid Logistic Map. *Appl. Sci.* **2023**, *13*, 4034. https://doi.org/10.3390/app13064034

Academic Editor: Byung-Gyu Kim

Received: 6 January 2023
Revised: 17 March 2023
Accepted: 18 March 2023
Published: 22 March 2023

1. Introduction

Recently, a lot of digital data has been transmitted over the Internet. This data can be in the form of semi-structured, unstructured, or structured data. In this digitally driven world, information security is of great importance. Various organizations and institutions invest a lot of money in the security of their resources, especially in protecting the privacy of employees, online transactions, and confidential information stored or transmitted in the form of video, text, audio, or image.

This means that in today's era of the Internet of Things, everything is connected to the Internet, and therefore, the security of information must be ensured. Here, it is imperative to preserve the integrity, confidentiality, and availability of information system resources on all communication channels, starting from hardware, software, data, or telecommunications. There are various protection methods, such as antivirus programs, firewalls, intrusion detection systems, biometric verification processes, routing controls, and cryptographic techniques.

Text encryption is a widely used technique to secure sensitive information from unauthorized access, which is usually achieved through the use of cryptography. However, when it comes to image encryption, additional techniques are required to preserve the visual content and quality of the image while ensuring its confidentiality. In this context of image encryption, both chaotic and non-chaotic methods have been used, where chaotic methods have been found to be more effective due to their ability to provide higher security, faster encryption, and better resistance to attacks. Some studies, such as [1], have incorporated chaotic systems into their image encryption algorithms with other cryptography techniques.

The complexity of chaotic systems can provide a higher level of security for encrypted information, and key attributes to consider when using chaos for encryption include initial condition sensitivity, ergodicity, simplicity, and randomness. Generally, different techniques of image encryption have been introduced and used, such as Chaos-Based Encryption [1], which is a modern and innovative method for securing digital images. This technique utilizes the principles of chaos theory to scramble the original image in a way that is not only secure but also visually pleasing. Unlike traditional image encryption methods that rely on mathematical algorithms, chaos-based image encryption uses the inherent randomness and unpredictability of chaotic systems to generate a key that can be used to encrypt and decrypt the image. This means that the encrypted image is not susceptible to attacks based on the prediction or manipulation of algorithmic patterns. The use of chaos theory in image encryption has several advantages over traditional methods because they provide a high level of security due to their unpredictability and retain their visual quality, which makes them ideal for applications such as medical imaging and digital art. There are several chaotic maps that have been used to generate random sequences of numbers in chaotic systems, including the Sine map [2], Henon map [3], and logistic map [4]. The logistic map is a one-dimensional chaotic system that can be modified to create more complex confusion in two- or three-dimensional space.

Transform Domain Encryption [5] is a method of securing digital images by transforming the image from the spatial domain to a different domain, such as the frequency or wavelet domain, before encryption. The goal of this technique is to utilize the unique properties of different transform domains to enhance the security of the encrypted image. Transform Domain image encryption has several advantages over traditional image encryption methods. Firstly, the transformed image often has a lower correlation between its adjacent pixels, which makes it more difficult for attackers to use statistical attacks to reverse the encryption. Secondly, the transformed image often has a more robust representation of the image content, which makes it more resilient to attacks based on image manipulation. Examples include discrete cosine transform (DCT) and discrete wavelet transform (DWT). This method has the drawback of being vulnerable to chosen-text attacks, in which the attacker examines the coefficients of the plaintext and encrypted image in order to recover the secret keys. Another weakness is the susceptibility to brute-force attacks, which take place when the key is too small. Although we did not incorporate any transform domain encryption techniques into our proposed study, we did integrate multiple techniques with long and complex secret keys.

Cryptographic Hash Functions play an important role in the field of image encryption by providing a secure method of verifying the authenticity and integrity of an encrypted image. A cryptographic hash function is a mathematical algorithm that takes an input, such as an image, and produces a fixed-length output, called a hash, that is unique to that input. When an image is encrypted, the hash of the original image can be computed and stored alongside the encrypted image. When the encrypted image is decrypted and the original image is retrieved, a new hash of the recovered image can be computed and compared to the stored hash. If the two hashes match, it is evidence that the encrypted image has not been tampered with and is a secure representation of the original image. The use of cryptographic hash functions in image encryption provides several advantages. Firstly, it allows for the verification of the authenticity and integrity of the encrypted image without the need to compare the entire image to the original. Secondly, it provides a way to detect any changes made to the encrypted image, even if the changes are small and go unnoticed by the human eye. Collision attacks, pre-image attacks, salt attacks, and rainbow table attacks are some of the drawbacks of hash functions in image encryption. Although we have not used a hash function, our proposed technique ensures that it is not possible to use a fake image to obtain the secret key or tamper with the final product by comparing its hash value with the encrypted image.

Visual Cryptography [6] is a method of encrypting images using techniques that are based on visual perception, rather than purely mathematical algorithms. The goal of visual

image cryptography is to make the encrypted image appear as a random, meaningless pattern to an unauthorized observer, while still preserving the original information when decrypted by the intended recipient. Visual image cryptography can be more secure and robust against attacks, as it takes advantage of the human visual system and its limitations. Another approach used in image encryption is the use of steganography techniques. Steganography is the practice of hiding secret information [7] within an image in a way that is not noticeable to the human eye. The goal of image steganography is to provide a covert communication channel that can be used to transmit sensitive information without being detected by unauthorized parties. The secret information can be in the form of text, images, or other digital data, and it is embedded into the host image in such a way that it does not alter the visual appearance of the image. The ability to hide sensitive data within digital media files that are resistant to file compression, cropping, and scaling is one of the advantages of using steganography. However, its weaknesses include easy detection of the original information when skilled attackers use specialized tools to detect the presence of hidden data and extract information, and the technique also has a limited amount of data that can be hidden within other digital files. Despite these weaknesses, steganography remains an effective way to hide sensitive data, as it allows users to embed private information within a digital file in an inconspicuous way.

A research study titled "A Secure Image Encryption Method Using Chaotic Map" by [8] explored the use of chaotic maps for image encryption. The authors proposed a technique that involves dividing the image into blocks and encrypting each block using a chaotic map. The effectiveness of this method was tested on various images and found to provide security for the images. One of the benefits of the image encryption technique proposed in this study is that it is simple to use and does not require a lot of computing power, which makes it suitable for use in real-time applications. While the image encryption technique proposed in this study has some strengths, it also has some limitations. One potential weakness is that chaotic maps, which are used in the proposed method, can be susceptible to statistical analysis attacks. In the study "A New Image Encryption Algorithm Based on Chaotic Maps and DNA encoding" by [9], the authors presented a new approach for encrypting images that combines chaotic maps with DNA encoding and Huffman coding algorithms. The proposed method was tested on various images and found to be effective in providing security. One advantage of this technique is that it uses multiple encryption techniques, which can enhance the overall security of the method. However, it is possible that the DNA encoding and Huffman coding algorithms may be susceptible to certain attacks, which could potentially compromise the security of the proposed method. The study "Selective image encryption method based on dynamic DNA coding and a new chaotic map" by [10] introduced a new method for encrypting images that combines chaotic maps with DNA sequence operations. The proposed technique was tested on various images and found to be effective in providing security. One benefit of this method is that it uses both chaotic maps and DNA operations, which can enhance the overall security of the technique. However, it is possible that DNA operations may be susceptible to certain attacks, which could potentially compromise the security of the proposed method.

For example, in the study by Xishun et al. [11], the authors propose a novel image encryption method that combines the Kronecker product with DNA computing. The authors demonstrate that their method is able to achieve high levels of security and efficiency, making it a promising approach for image encryption. Several other researchers [12–14] have used the Kronecker product in their image encryption techniques. For instance, in the study by [15], the authors use the Kronecker product to combine multiple encryption keys in order to achieve a higher level of security. Similarly, in [15], the semi-tensor product theory is used to combine a boolean network in order to encrypt images.

The Hill cipher has been widely used in image encryption due to its simplicity and effectiveness in providing security. In a previous study [16], the authors proposed an image encryption technique based on the Hill cipher, which involves dividing the image into blocks and encrypting each block using the Hill cipher algorithm. The proposed method

was tested on various images and was found to be effective in providing security for the images. However, the Hill cipher has some weaknesses that may make it vulnerable to attacks. One weakness is its reliance on a fixed key, which can be discovered through a brute force attack. Additionally, the Hill cipher is vulnerable to known-plaintext attacks, where an attacker with access to both the plaintext and the corresponding ciphertext can easily recover the key. This can be mitigated by using a large key size and regularly changing the key, but it is still potentially vulnerable.

There is a lack of studies that have successfully integrated Hill cipher, chaotic map, and the Kronecker xor product for image encryption. As such, this study represents a significant innovation in the field by presenting a unique combination of these encryption techniques. The combination of the Hill cipher, chaotic map, and Kronecker xor product is expected to provide robust security for images, as each technique brings its own strengths to the table; it should also be able to overcome the problem of plain text attacks due to having more than one secret key. Hill cipher is a well-known and widely-used technique for symmetric key encryption, the chaotic map is a powerful tool for generating randomness and unpredictability, and the Kronecker product is a mathematical operation that can be used to create complex, nonlinear transformations. By leveraging the strengths of these techniques, this study's proposed image encryption method is expected to be highly effective in protecting the confidentiality and integrity of images.

Contribution of the Study

- This paper presents a novel image encryption method that combines the Hill Cipher, sigmoid logistic map, and Kronecker xor product matrix techniques. The sigmoid logistic map was modified in this method to generate complex pseudo-random numbers that are effective in encrypting images. By integrating these three techniques, the proposed image encryption method offers a new approach to digital encryption that has not been explored previously. The combination of the Hill Cipher, sigmoid logistic map, and Kronecker xor product matrix brings together a range of strengths to provide robust security for images. This contribution is expected to be a valuable addition to the field of image encryption.
- One contribution of this paper is the use of the Kronecker bitwise exclusive OR operation to make the image pixels more obscure and increase the size of the image. The Kronecker product of matrices is a well-known technique that is often used for digital watermarking, encryption, and compression. In this study, we adapted the Kronecker product approach by replacing the product operator with the xor operator, which generates extremely large and complex keys. These keys are then combined with the complex keys generated by the modified sigmoid logistic map through the xor operation. This contribution represents a new application of the Kronecker product and xor operator for image encryption and adds to the growing body of knowledge in this field.
- It is important to note that the Kronecker product used in this paper is the Kronecker xor product, not the Kronecker tensor product. The Kronecker xor product can significantly increase the size of the data being stored when used for encryption. While this additional storage space may be a consideration, it is essential to weigh this against the importance of protecting the sensitive information being stored. The Kronecker xor product is an effective method for ensuring that data is secure and not accessible to unauthorized parties.
- Another contribution of this paper is the thorough testing and evaluation of the encrypted image, which was performed and compared to other similar studies. The results of these tests suggest that the proposed image encryption method meets all the criteria for good encryption. The effectiveness of the proposed method was demonstrated through comparison with other techniques, providing confidence in its ability to secure images against potential threats.

The structure of this study can be summarized as follows. Section 2 provides an overview of the basic concepts and background information necessary for understanding the proposed image encryption method. In Section 3, the details of the encryption process and its implementation are described and illustrated. Section 4 presents the simulation results, analysis, and evaluation of the proposed encryption scheme, as well as a comparison with other studies. Finally, the conclusions of the study are presented in the last section.

2. Preliminaries

2.1. Logistic Map

The logistic map results from the logistic equation, sometimes called the Verhulst model [17], which is often used to predict the growth of the population of living beings or the occurrence of natural phenomena. The logistic equation is described by the following differential Equation (1).

$$\frac{dN}{dt} = \frac{rN(K-N)}{K} \tag{1}$$

where r is the Malthusian parameter [18] that determines the growth of a living organism at a given time, and K is called a carry. If we let $x = N/K$, then the described Equation (1) after dividing both sides with K becomes the following Equation (2).

$$\frac{dx}{dt} = rx(1-x) \tag{2}$$

which can be written as Equation (3).

$$x_{n+1} = rx_n(1-x_n) \tag{3}$$

In the logistic map, "r" is a positive value known as the "biotic potential". When the value of "r" is greater than 3.5 and an initial parameter, "$x_0 \in [0,1]$," is used, the map produces a chaotic system that can be used for encryption. This has been demonstrated in reference [19], where the logistic map is used to encode the selected content of an image. The combination of the logistic map and sine map results in a complex, chaotic map that offers good security performance. The use of these maps in image encryption is rapidly increasing, as demonstrated in references [19–22]. This trend is likely to continue as researchers continue to explore the potential of chaotic maps for image encryption.

2.2. Sigmoid Function

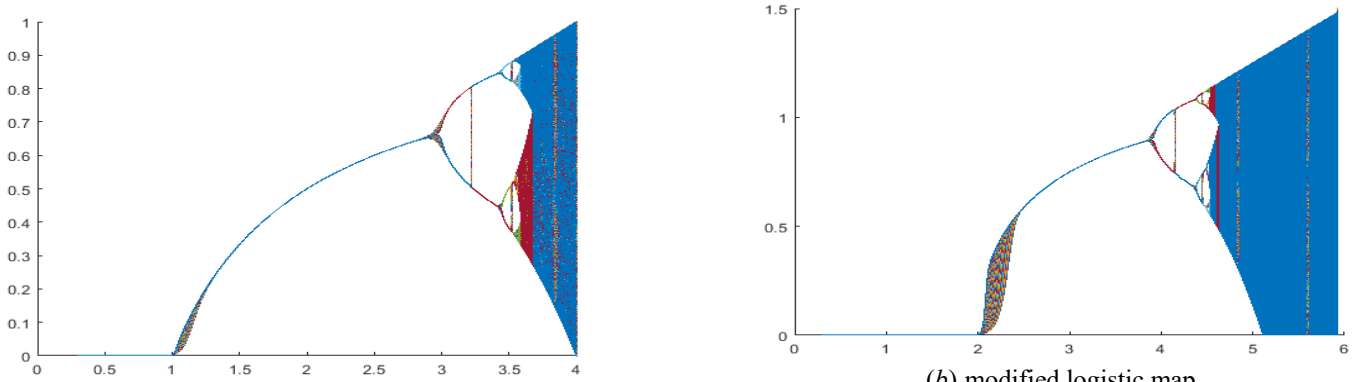
This is a mathematical function that maps any input value between 0 and 1, mostly as the probability of a binary number. The sigmoid function is expressed in different ways, but the most used is defined as in Equation 4, known as the logistic function.

$$f(x) = \frac{1}{1+e^{-x}} \tag{4}$$

where x is the input value, e is the mathematical constant e, and $f(x)$ is the output value. The sigmoid function has been used in many applications, including machine learning, neural networks, and logistic regression. As explained in depth in [23], the shape of the sigmoid function is characterized by the steep slope around the origin and the shallow slope away from the origin. Any changes in the input value near the origin result in larger changes in the output values. The disadvantage of the sigmoid function is that it is prone to saturation, which means the slope of the function becomes small as input values become very large or very small. To overcome this limitation, this study integrates the sigmoid function and the logistic map, which means the logistic map is the alpha, and the sigmoid function is the omega, as seen in Equation (7).

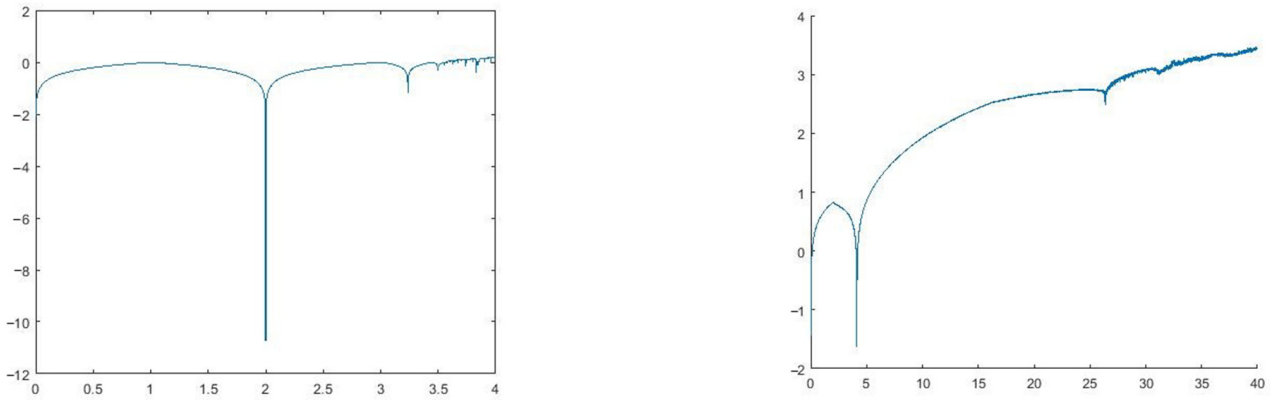
In the proposed system, the chaotic behavior begins when the control parameter "r" is set to 7.0158 and persists throughout the process. In order to achieve a good encryption result, we used a value of "r" equal to 20.1245 for K_2 and 20.1254 for K_3. Its corresponding

bifurcation diagram is shown in Figure 1. To obtain good and complex chaotic results, it is important that the proposed map has a Lyapunov exponent with a positive value greater than one. This is because the Lyapunov exponent reflects the rate of separation between initially close trajectories in the phase space of the system, and a positive value indicates that the system is chaotic. In this study, we have analyzed the Lyapunov exponent of both the basic logistic map and the modified sigmoid logistic map, as shown in Figure 2. The results demonstrate that both maps have Lyapunov exponents with positive values, indicating that they are suitable for use in image encryption.



(*a*) logistic map (*b*) modified logistic map (r = 20.1245)

Figure 1. Bifurcation diagram.



(a) logistic lyapunov exponent (b) modified sigmoid logistic lyapunov exponent

Figure 2. Lyapunov exponent.

2.3. Kronecker Product

The Kronecker product, also known as the tensor product, is a fundamental operation in mathematics and engineering. It is a way to combine two matrices or tensors into a single matrix or tensor, and has a wide range of applications, including signal processing, image processing, machine learning, error correction, repeated replication of equilibrium, quantum gates, and statistics in variance estimation. The concept of the Kronecker product has a long history dating back to the 19th century, and has been extensively studied and applied in many different fields, as explained in detail in [24].

The Kronecker product, denoted by the symbol "⊗", is a binary operation that takes two matrices A and B and produces a new matrix C as the result. The size of the resulting matrix C is determined by the sizes of the matrices A and B. Specifically, if A is an m-by-n

matrix and B is a p-by-q matrix, then the resulting matrix C is an mp-by-nq matrix. The elements of matrix C are computed as follows in Equation (5).

$$C(\mathrm{i},\mathrm{j}) = A(\mathrm{i} \bmod \mathrm{m}, \mathrm{j} \bmod \mathrm{n}) \times ((\mathrm{i} \operatorname{div} \mathrm{m}), (\mathrm{j} \operatorname{div} \mathrm{n})) \quad (5)$$

where "mod" denotes the modulo operation and "div" is the integer division operator. The Kronecker product has many useful properties, including distributivity over addition and scalar multiplication and bilinearity. In this paper, we propose a new technique known as the Kronecker xor product that uses the same approach as the Hadamard matrix and Kronecker product to increase the size of the image from $n \times n$ to $n^2 \times n^2$. The operation is performed between two paired columns and adjacent rows, using the results of the pseudo-code random number of sigmoid logistic maps as the input source leading to the encrypted image.

2.4. Hill Cipher

The Hill cipher is a classic polyalphabetic substitution cipher technique that uses linear algebra to encrypt and decrypt messages. It was invented by Lester S. Hill in 1929 [25,26] and has since been widely used as a method of secure communication. In the Hill cipher, the plaintext message is represented as a matrix of numbers, and a key matrix is used to perform a series of matrix multiplications and additions to encode the message. The key matrix must be invertible so that it can be used to reverse the encryption process and decrypt the message. Some of the weaknesses when the hill cipher is used alone include having a fixed block size in which the size of plaintext and cipher text is always the same; this makes the cipher technique vulnerable to block analysis. Another weakness includes known plaintext attacks, in which the adversaries can decrypt the future image if it contains both plaintext and ciphertext.

To overcome these vulnerabilities, we propose an encryption method that integrates the Hill cipher with the sigmoid logistic map and the Kronecker bitwise exclusive OR operation. The resulting method is simple and effective, making it a promising candidate for secure image communication. However, as with any cryptographic technique, it is important to use the Hill cipher in combination with other methods to ensure the maximum level of security.

3. Design and Implementation of the Proposed Scheme

A cryptosystem is a system designed for secure communication that employs a set of algorithms and protocols to encode and decode messages. This study presents a technique that combines multiple encryption methods to enhance the security of digital images. The proposed approach first scrambles the pixels of an image and then applies confusion and diffusion techniques. The following steps outline the process used in this study.

3.1. Image Scrambling

Step 1: The scramble process is the same as that of AES encryption [27]. The shift rows operation works by shifting the values in each row of the state matrix to the left by a certain number of positions. The number of positions shifted by each row in the matrix is determined by its index. Specifically, the first row is not shifted at all, the second row is shifted by one position, the third row by two positions, and the fourth row by three positions; the process continues for the whole image. Figure 3 shows the example in a matrix P of size 4×4.

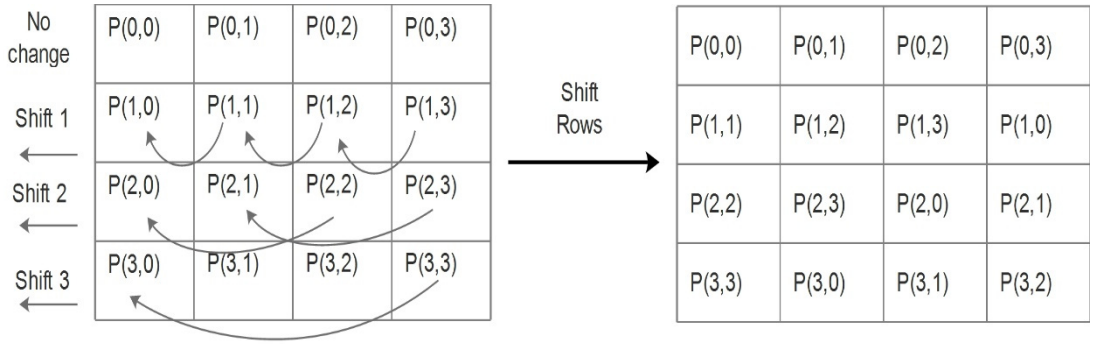


Figure 3. Shift rows transformation.

3.2. Diffusion of Image Pixels

Step 2. The top values of matrix P are preserved, while the top value of each odd column is used to perform a bitwise exclusive OR operation with all values in the corresponding even column, excluding the top value. Similarly, the top value of each even column is used to perform a bitwise exclusive OR operation with all values in the corresponding odd column, also excluding the top value to obtain matrix H. Figure 4 and Equation (6) illustrate the process of performing a bitwise exclusive OR operation on the matrix, where x and y represent odd and even columns, respectively, and i, j represent the different image pixel positions within the matrix at the nth position.

$$\begin{cases} x(i, j+n) = y(i+1, 0) \oplus x(i, j+n) \\ y(i, j+n) = x(i-1, 0) \oplus y(i, j+n) \end{cases} \tag{6}$$

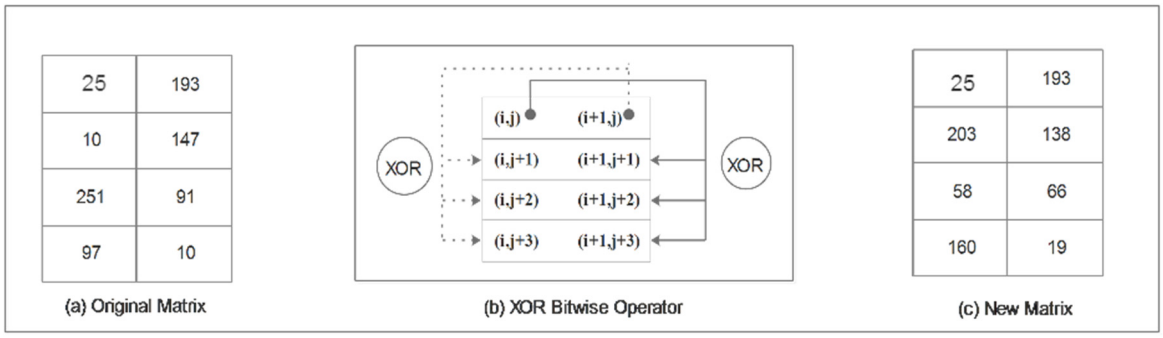


Figure 4. Diffusion process 1.

Step 3. In this process, all elements in a matrix H' are encrypted using the Hill Cipher technique, in which a predefined 2×2 secret key K is used to encrypt the diffusion matrix H'. The process involves iterating over the 4×4 block matrix until the entire image is encrypted. To ensure that the encryption is secure, a modulo operation of 256 is performed on each element of the matrix during the process, as shown in Equation (7), to have intermediary Ciphertext K_1.

$$K_1 = \text{mod}\big((H' \times K), 256\big) \tag{7}$$

Step 4. In this process, a sigmoid logistic map is used to randomly generate secret keys K_2&K_3 according to Equation (8). Secret key K_2 is used in the exclusive operation with secret key elements K_1 to obtain intermediary Ciphertext K_4, as shown in Equation (9). This helps to ensure that the secret keys are unique and more secure.

$$K_2 \& K_3 = \left| r \times \left(\frac{x}{1 + exp(x)} \right) \times \left(1 - \left(\frac{x}{1 + exp(x)} \right) \right) \right| \tag{8}$$

$$K_4 = K_1 \oplus K_2 \tag{9}$$

Step 5. In this stage of the diffusion process, the Kronecker xor transformation process is applied to matrix K_4, expanding the image from size $n \times n$ to size $n^2 \times n^2$. To further obscure the original message, the columns or rows of the matrix are shifted by the i^{th} positions, where i acts as the secret key in the process. In the proposed technique, the rows are moved down, and the value of i is chosen as $i = 3$. Each element of the matrix K_4 undergoes a bitwise exclusive OR operation with the other elements of K_4, while its value remains unchanged to form the image (secret keys) K_5. The Kronecker xor product, as shown in Equation (11), illustrates the operation process on a 2×2 block matrix M, which is a part of K_4, as shown in Equation (10). This helps to ensure that the diffusion process is secure and resistant to attacks. Figure 5 shows the expansion of the 2×2 encrypted image that results in the 4×4 matrix.

$$M = \begin{bmatrix} i,j & i+1,j \\ i,j+1 & i+1,j+1 \end{bmatrix} \tag{10}$$

$$M \oplus M \Leftarrow K_5 = \begin{bmatrix} (i,j) & (i,j) \oplus (i+1,j) & (i,j) \oplus (i+1,j) & (i+1,j) \\ (i,j) \oplus (i,j+1) & (i,j) \oplus (i+1,j+1) & (i,j+1) \oplus (i+1,j) & (i+1,j) \oplus (i+1,j+1) \\ (i,j) \oplus (i,j+1) & (i,j+1) \oplus (i+1,j) & (i,j) \oplus (i+1,j+1) & (i+1,j) \oplus (i+1,j+1) \\ (i,j+1) & (i,j+1) \oplus (i+1,j+1) & (i,j+1) \oplus (i+1,j+1) & (i+1) \oplus (j+1) \end{bmatrix} \tag{11}$$

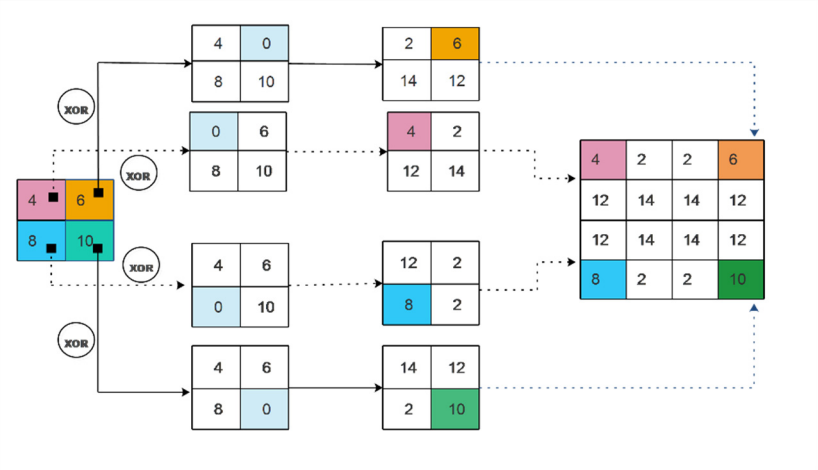


Figure 5. Kronecker xor product concept.

The use of the Kronecker xor product aims to obscure the original message and prevent unauthorized access to the information being transmitted. By expanding the image and shifting the rows or columns, the resulting encrypted image can be made to appear significantly different from the original, making it more difficult for unauthorized users to infer the contents of the message. Additionally, the use of a secret key in the transformation process adds an additional layer of protection, as the original message can only be decrypted with knowledge of the key.

Step 6. The exclusive operation is performed between intermediary ciphertext K_5 and secret keys K_3 to obtain the last encrypted image C, as shown in Equation (12).

$$C = K_3 \oplus K_5 \tag{12}$$

3.3. Flowchart of the Encrypted System

The following Figure 6 shows the encryption process of the whole operation of this proposed encryption mechanism.

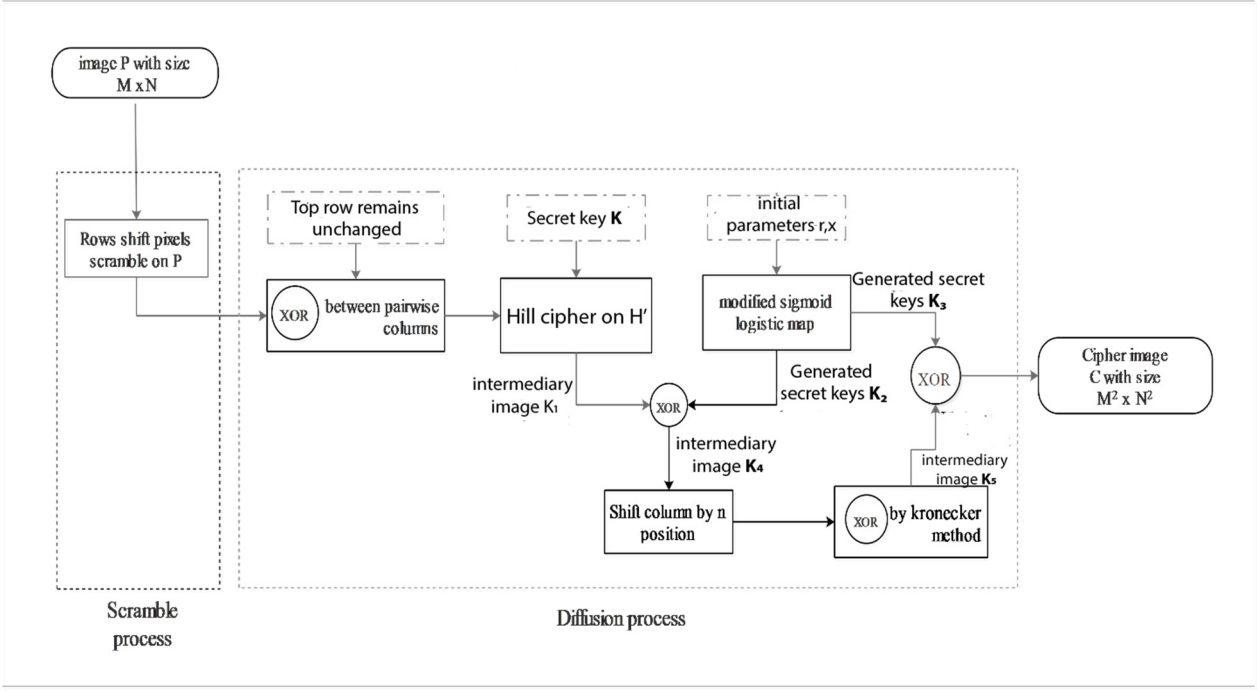


Figure 6. Flowchart of the encryption process.

3.4. Decryption Step for Proposed Algorithm

The decryption process is described in the pseudo-code at the end of this paper in Appendix A. The steps for decrypting an encrypted image are provided below.

Step 1. Input an encrypted image (C).

Step 2: Perform a bitwise exclusive OR operation between image (C) and secret key (K_3) from the sigmoid logistic map to obtain an intermediate image (K_5).

Step 3: Compress the intermediate image (K_5) using the Kronecker xor operation to obtain a new image (K_4).

Step 4: Perform a bitwise exclusive OR operation between the new image (K_4) and secret keys (K_2) from the sigmoid logistic map to obtain another new image (K_1).

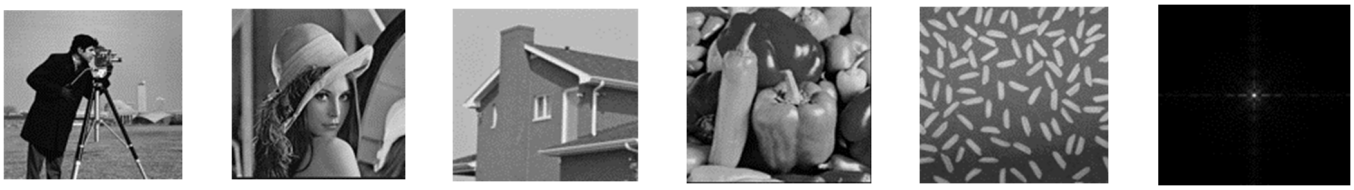
Step 5: Decrypt the new image (K_1) using the inverse of the hill cipher and a key (K inverse) to obtain an intermediate image (H).

Step 6: For each element in an even column of the intermediate image (H), perform a bitwise exclusive OR operation with the top value of the corresponding odd column. For each element in an odd column, perform a bitwise exclusive OR operation with the top value of the corresponding even column, while preserving the top values of matrix (H).

Step 7: Determine the number of rows and columns in the intermediate image (H). Iterate over each row in the image (H), starting from the second row (the first row is not shifted). For the current row, shift it to the right by its index within the matrix by concatenating the portion of the row that was shifted off the end with the portion that was not shifted until you obtain the plain image (P).

4. Simulation Results and Analysis

In this study, the image sizes $16 \times 16, 32 \times 32, 64 \times 64, 128 \times 128$ and 256×256 were used as the dataset for encryption. Insufficient resources, such as computers with high processing speed and good resolution, have hindered this study's ability to conduct experiments on data sets exceeding 256×256 image pixels, including those up to 12 megapixels. The images used for testing the proposed encryption are presented in Figure 7. The secret key of the Hill cipher was comprised of $K_h = [8\,7; 9\,6]$, and $r = [20.1245, 20.1254]$, $x_0 = 0.71234$ which serve as the control parameter for the logistic map and the initial conditional parameter, respectively.



(*a*) Cameraman (*b*) Lena (*c*) House (*d*) Pepper (*e*) Rice (*f*) Blacks

Figure 7. Image used for testing.

4.1. Key Space

The proposed encryption algorithm utilizes a total of three different keys, which significantly increases the key space and makes it more difficult for unauthorized parties to decipher the encrypted information. The use of three keys in the proposed algorithm not only satisfies the requirements but also surpasses the standards set by IEEE [28], making it an ideal and secure option for encryption. These keys include K, which is the secret key for hill cipher, as well as K_2 and K_3, generated from the modified sigmoid logistic function. According to the guidelines set forth by the IEEE [28], a key space of at least 2^{100} is necessary for strong encryption. The Hill cipher (K) comprises four different positions due to the 4×4 square matrix, each of which has 254 possible values (ranging from 0 to 255). This results in a total of 2^{254} combinations or key spaces. For the case of the Sigmoid logistic map (K_2 and K_3), the value of the key length varies depending on the size of the image. When using an image of 256×256, the resulting key length is 65,536. The keys consist of only two positions; each position can have two possible values (0 and 1 when converted to binary numbers), leading to a total combination of 2^{65536}. The same approach can be used for the image of sizes 128×128, 512×512, and 32×32 to obtain the key length and number of key spaces. Additionally, as shown in Table 1, the proposed algorithm compares favorably to other studies of a similar nature in terms of averages, further demonstrating its effectiveness as an encryption method. Overall, the use of multiple keys in the proposed algorithm effectively strengthens the security of the encryption and makes it a reliable choice for protecting sensitive information. Equation (13) is used to find the number of key spaces for three keys in general.

$$keys = \left(10^{15\times 3}\right) \approx 2^{149} \tag{13}$$

Table 1. Key space comparison.

	Proposed	**Ref [29]**	**Ref [30]**	**Ref [31]**	**Ref [32]**
Average	2^{149}	2^{555}	2^{196}	2^{187}	2^{199}

4.2. Key Sensitivity

It is crucial that encryption algorithms are sensitive to even small changes in the secret key. This is because even a slight change in the key can significantly affect the output. As shown in Figure 8, we tested the initial value $x_0 = 0.0117$ of our proposed algorithm during the encryption process and slightly changed it to $x_0 = 0.0117 \pm 10^{\pm 14}$ during decryption. The results demonstrate that our algorithm is highly sensitive to changes in the key, making it an effective and secure method for protecting data. This is an important consideration when choosing an encryption algorithm, as it ensures that the output will be protected even if the key is slightly altered during the encryption or decryption process.

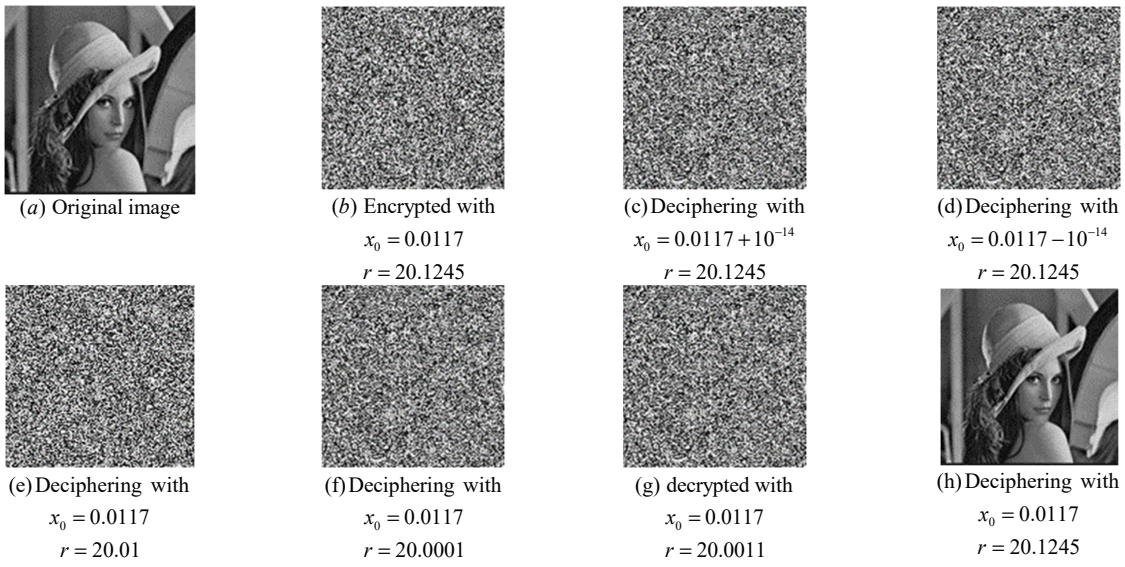


Figure 8. Key sensitivity.

4.3. Noise and Data Loss Analysis

When transmitting information online, it is important to consider the potential for it to become lost or overwhelmed by unrelated and unimportant data, known as noise. If digital images are not properly managed, this can negatively impact their usefulness when attempting to analyze, interpret, or retrieve the information they contain. It is important to handle digital information with care to ensure that it remains clear and meaningful. As shown in Figure 9, we evaluated the decryption outcomes of encrypted images that underwent varying levels of data loss. The recovery test was conducted across a range of data loss from 0.01% to 0.9%, and the results revealed that data loss up to 0.5% can be effectively recovered. Figure 10 shows the effectiveness of the proposed mechanism in handling various forms of noise, as demonstrated through the noise effect test. The test covered a range of gap sizes from 0.01% to 0.2% when applied to encrypted images, and the recovery results indicated that data up to 0.19% could be successfully recovered. However, beyond that point, the output quality deteriorates, as evidenced at the 0.2% level. The results demonstrate the capability of the mechanism to deal with various degrees of noise as well as data loss to a considerable extent. Our approach proves to be highly efficient in recovering lost or corrupted data during transmission. The design of the method aims to be highly effective in retrieving information, even in the presence of significant distortion during transmission.

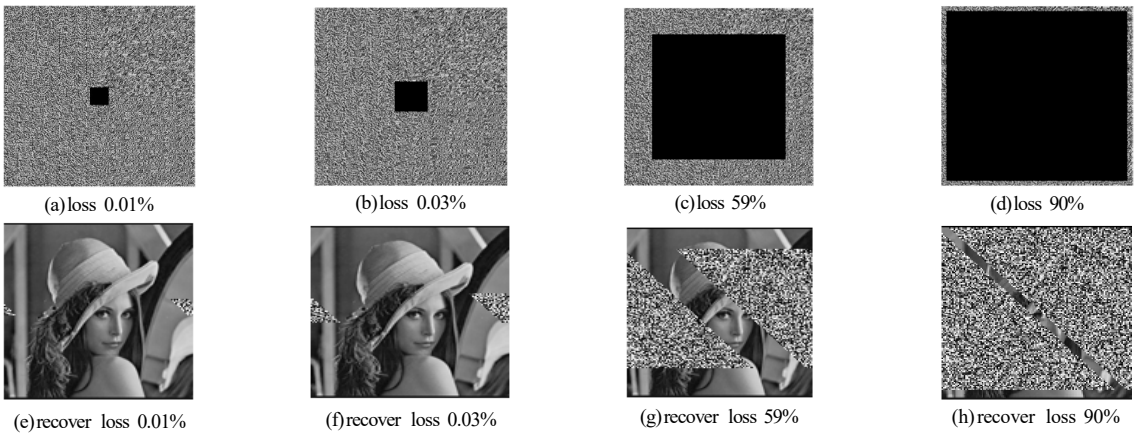


Figure 9. Data loss and recovery in different degrees.

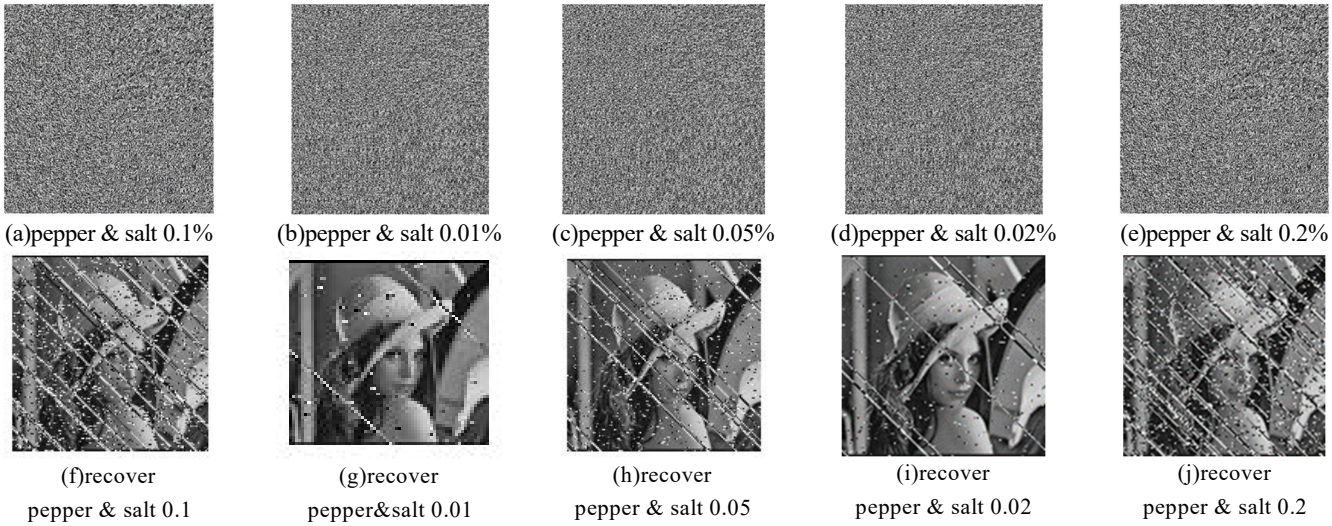


Figure 10. Noise effect on ciphertext and recovery image in different degrees.

4.4. Differential Attack

Two methods, namely, the number of pixel change rate (NPCR) and the unified average change intensity (UACI), are used to measure the resistance of differential attacks. The number of pixel change rate is a measure of the difference between two images. It is calculated by comparing the number of pixels that have changed between the two images and expressing the result as a percentage. An ideal NPCR value for an encrypted image is above 99%, indicating that all pixels have been significantly altered from their original positions. The UACI is calculated by comparing the pixel values of two images and quantifying the average intensity of the changes between the two images. It is expressed as a percentage and can range from 0 to 100, with higher values indicating a greater difference between the two images. To calculate the UACI, the differences between the pixel values of the two images are first determined. These differences are then averaged and expressed as a percentage by dividing the average difference by the maximum possible difference between the pixel values and multiplying by 100. The two methods are calculated using the following mathematical Equations (14) and (15).

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \tag{14}$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{C_1(i,j) - C_2(i,j)}{255} \times 100\% \tag{15}$$

$$D(i,j) = \begin{cases} 0 iff\, C_1(i,j) = C_2(i,j) \\ 1 iff\, C_1(i,j) \neq C_2(i,j) \end{cases}$$

where $M \times N$ represent the size of the image in terms of width and height, and c_1 and c_2 are two cipher images after changing one pixel position of their original plain image. Table 2 presents the NPCR and UACI values for the encrypted images. To find the average NPCR or UACI, we add up the results for all the image sizes and then divide by the total number of images. Table 2 shows the average NPCR and UACI values for about 24 images of different sizes. When comparing the NPCR and UACI values of two images of the same type and the change in pixel values, Table 3 compares these values with those from other similar studies using the Lena image of size 256 × 256. These results demonstrate the effectiveness of the proposed encryption in resisting differential attacks.

Table 2. NPCR and UACI values of Cipher text images.

Algorithm	Average	Lena (128×128)	Lena (64×64)	Lena (32×32)	Cameraman (64×64)	Pepper (64×64)	Pepper (32×32)	Baboon (32×32)
NPCR	99.62314	99.6073	99.6213	99.6912	99.5861	99.6639	99.6541	99.5781
UACI	33.3955	33.4477	33.4213	33.4102	33.3896	33.3581	33.4123	33.3514

Table 3. The proposed method with other encryption methods.

Algorithm	Proposed	Ref [33]	Ref [32]	Ref [34]	Ref [35]	Ref [36]
NPCR	99.62314	99.6094	99.62	99.6273	99.6101	99.60
UACI	33.3955	33.4635	33.50	33.4691	33.45.93	33.45

4.5. Histogram

The shape of the histogram of an image is influenced by the distribution of its pixel values. A histogram can be normal, skewed to the right, or skewed to the left. In the case of encrypted images, a flat histogram indicates a successful encryption process. Figure 11 shows encrypted and decrypted images with their corresponding histograms, while Figure 12 presents the spatiotemporal histogram of plaintext and ciphertext images of various images. These results suggest that the proposed encryption algorithm is effective in protecting against statistical attacks.

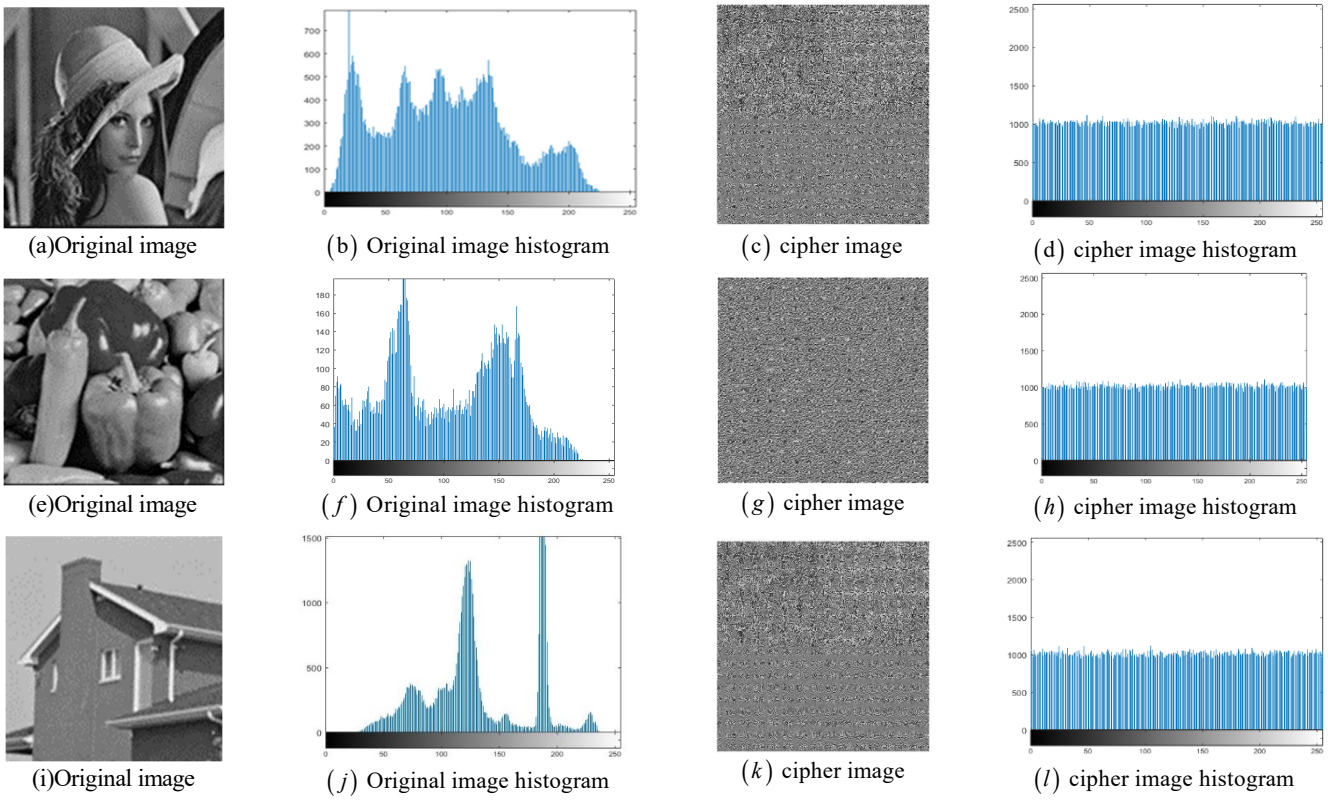
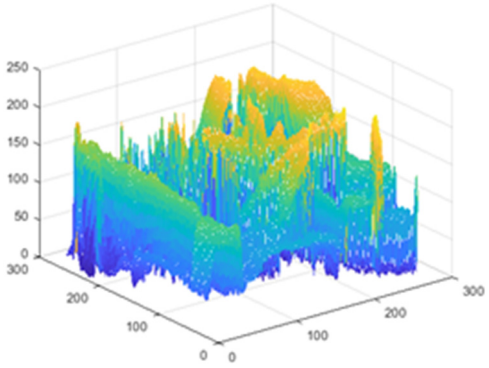
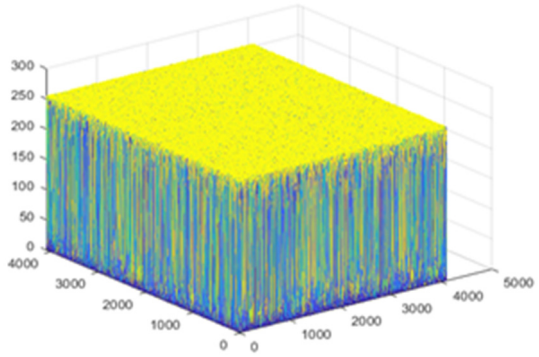


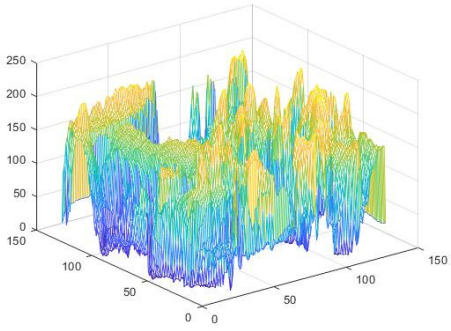
Figure 11. Histogram of 128 × 128 plaintext in 2D.



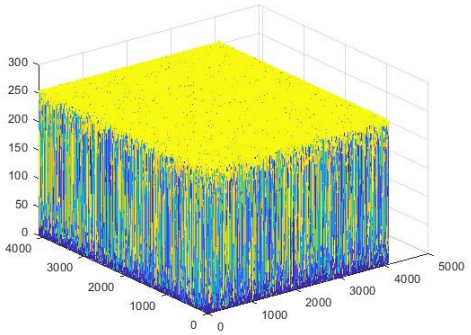
(a)Spatial distribution of pixels in plaintext Lena (256×256)



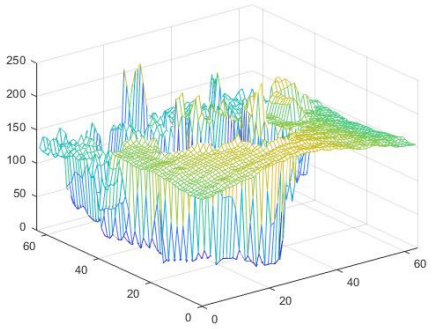
(b)Spatial distribution of pixels in ciphertext Lena (256×256)



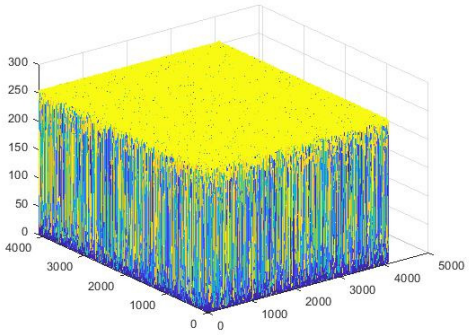
(c)Spatial distribution of pixels in plaintext pepper (128×128)



(d)Spatial distribution of pixels in ciphertext pepper (128×128)



(e)Spatial distribution of pixels in plaintext cameraman (64 × 64)



(f)Spatial distribution of pixels in ciphertext cameraman (64×64)

Figure 12. Spatiotemporal histogram of the plaintext image and the ciphertext image.

4.6. Information Entropy

Information entropy is a technique used to assess the randomness of variables in an image. When applied to both encrypted and original images, it can help to determine the variables present in the image. A high value of information entropy, close to 8, indicates a high level of randomness in the variables and a strong image encryption mechanism. Information entropy can be mathematically calculated using Equation (16).

$$e = \sum_{i=1}^{256} p(i) \log\left(\frac{1}{p(i)}\right) \tag{16}$$

In this study, we have analyzed the values of information entropy for both plain and encrypted images, and the results are presented in Table 4. In addition, Table 5 presents the average results of the Lena image encrypted with all image sizes used in this study and

compares them to those of other similar studies. These tables provide valuable insights into the security and performance of the proposed encryption method.

Table 4. Information entropy values of Ciphertext images.

Algorithm	Lena (128×128)	Lena (64×64)	Cameraman (64×64)	Pepper (64×64)	Pepper (32×32)	Baboon (32×32)	Rice (256×256)
Plain Image	6.4971	6.4512	6.7457	7.3801	7.5264	6.5335	7.5468
Encrypted image	7.9981	7.9978	7.9991	7.9954	7.9991	7.8546	7.9989

Table 5. Information entropy comparison.

	Proposed (Average)	Ref [19]	Ref [29]	Ref [33]	Ref [35]	Ref [37]	Ref [38]
Entropy	7.9992	7.999284	7.9022	7.996513	7.9977	7.9970	7.9974

4.7. Correlation

Correlation analysis is a statistical method used to evaluate the strength and direction of the relationship between two variables. The correlation coefficient, a measure of association, quantifies the degree to which two variables are linearly related. The Pearson correlation coefficient is the most commonly used measure of association and can range from -1 to 1, with -1 indicating a perfect negative correlation, 1 indicating a perfect positive correlation, and 0 indicating no correlation. When the correlation coefficient value is significantly different from 0 on the encrypted image, it becomes difficult or impossible to interpret the data without using the appropriate decryption key or method. An effective image encryption scheme should result in a low correlation between adjacent pixels, which are randomly distributed in order to ensure secure protection of the image, as seen in Figure 13d–f. In the case of a plain image, the correlation among variables is concentrated, making it easier to predict the original image, as seen in Figure 13a–c. Equation (17) shows how to calculate the correlation coefficient of two variables in an image. Table 6 presents the correlation coefficient for plain and encrypted images of Lena, House, and Cameraman in various sizes and the average value of the horizontal, vertical, and diagonal correlation and compares the results to those of previous studies on the encrypted Lena image. This demonstrates the effectiveness of our proposed encryption techniques and their potential for use in image encryption.

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{(x)}\sqrt{(y)}} \tag{17}$$

where $\operatorname{var}(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)^2\right)$, $\operatorname{cov}(\mathrm{x},\mathrm{y}) = \mathrm{E}([x - E(X)][y - E(Y)]), E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$, $E(y) = \frac{1}{N}\sum_{i=1}^{N} y_i$.

In our proposed scheme, the variables x_i and y_i represent the grayscale values of two adjacent pixels, with $E(x)$ denoting the average of x_i and $E(y)$ denoting the average of y_i. By applying Equation (16) to the proposed system, we observed a significant difference between the original and encrypted images. This indicates that the proposed system is resistant to statistical attacks.

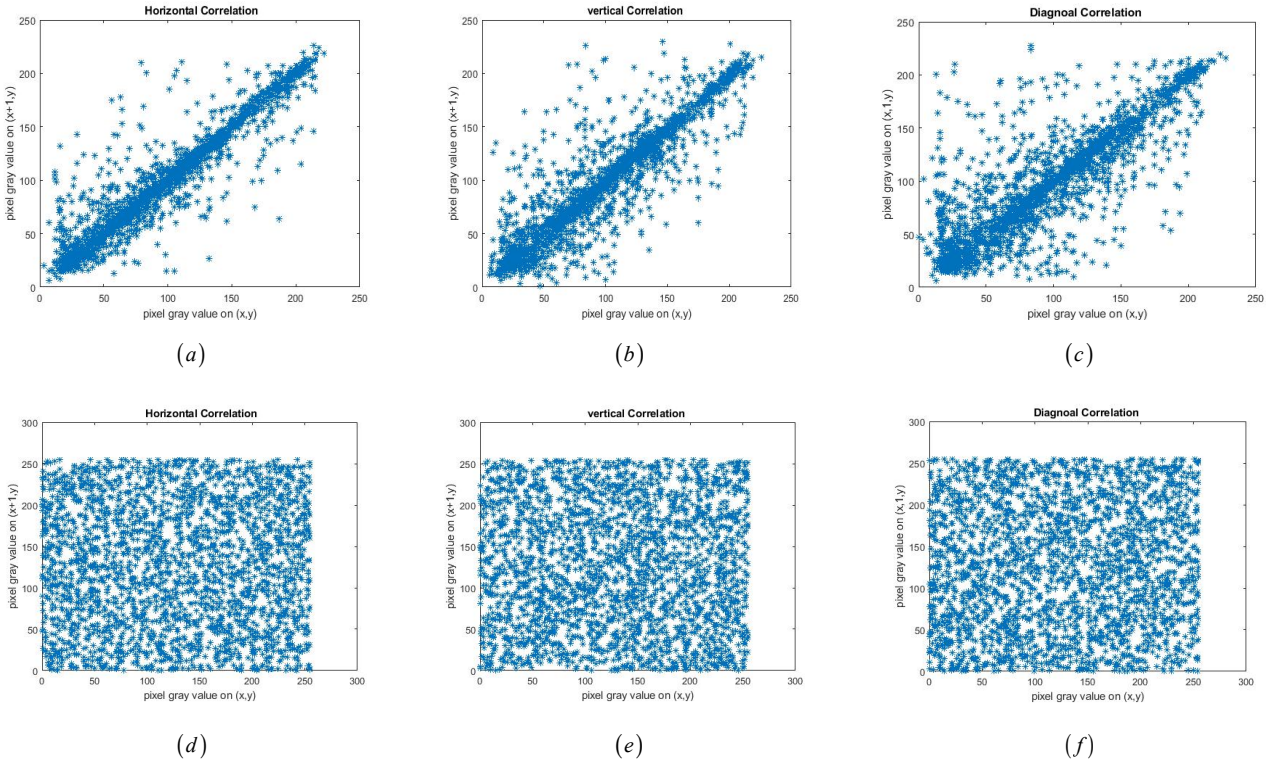


Figure 13. Correlation (128 × 128): (**a**–**c**) correlation of horizontal, vertical, and diagonal directions of the Lena graph, respectively; (**d**–**f**) correlation of horizontal, vertical, and diagonal directions of Lena encrypted graph, respectively.

Table 6. Comparison of average correlation of ciphertext.

Image	Plaintext			Ciphertext			Average
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal	
House(128 × 128)	0.9507	0.9396	0.9058	−0.0001	−0.0085	0.0009	−0.0025
Cameraman(64 × 64)	0.8908	0.9315	0.8443	0.0056	−0.0075	0.0159	0.0047
Lena (128 × 128)	0.8988	0.8573	0.9481	0.0049	−0.0022	−0.0042	−0.00053
Ref [29]	-	-	-	0.001136	0.00080	0.00147	0.00113
Ref [30]	-	-	-	0.002225	0.003075	0.001625	0.00230
Ref [39]	-	-	-	0.0021	0.0029	0.0023	0.00243

4.8. Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error Analysis (MSE)

Evaluating the quality of encrypted images is an important task in ensuring that the original image is accurately represented after being processed through an encryption algorithm. Two commonly used metrics for this purpose are the mean square error (MSE) and the peak signal-to-noise ratio (PSNR) [40]. The MSE is a measure of the similarity between the encrypted image and the original image. A low MSE value indicates that the two images are similar and that the encrypted image is of high quality, while a high MSE value indicates that the encrypted image is of poor quality. The MSE is calculated by subtracting the encrypted image from the original image, as shown in Equation (18). The PSNR is another useful metric for evaluating the quality of encrypted images. The higher the PSNR, the better the image quality; a lower PSNR value is an indicator of poor image quality. The PSNR is calculated mathematically, as shown in Equation (19). $P(i,j)$ and $E(i,j)$ represent the original and encrypted image pixels consecutively in this context. The mean square error (MSE) is always zero when the pixel values of the original and encrypted images are exactly the same (when $P(i,j)$ and $E(i,j)$ are equal). Table 7 demonstrates the effectiveness of the proposed image encryption method with the high values of the mean

squared error (MSE) and low values of the peak signal-to-noise ratio (PSNR), which was conducted using images that were 128 pixels by 128 pixels in size. These results suggest that the proposed method is capable of effectively protecting the confidentiality of images, and thus, has the potential to make a significant contribution to the field of image encryption.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (P(i,j) - E(i,j))^2 \tag{18}$$

$$PSNR = 10 \times \log_{10} \frac{M \times N}{\sqrt{MSE}} \tag{19}$$

Table 7. Mean squared error (MSE) and peak signal-to-noise ratio (PSNR).

	Cameraman	Lena	Pepper	Rice	House
MSE	107.856	89.2446	94.24519	101.1521	130.143
PSNR	8.3741	8.5655	8.4310	9.3546	9.2214

5. Conclusions

In this paper, we propose a new and innovative method for encrypting images that combines a variety of techniques to provide robust security. The proposed mechanism includes image scrambling, Hill cipher encryption, and the use of a sigmoid logistic map and Kronecker xor product techniques. The modified sigmoid logistic map, in particular, generates complex and secure pseudo-random numbers that are ideal for use in encryption. This enhances our understanding of digital encryption and helps to ensure the security of the proposed system.

The use of the Kronecker xor product in encryption can significantly increase the size of the data being stored. While the increased storage space required by the Kronecker xor product may be a consideration, it is important to balance this against the value of the data being protected. In today's digital age, data breaches and cyber attacks are becoming increasingly common, and the consequences of such events can be severe. By integrating techniques such as the Kronecker xor product to encrypt data, we can ensure that the proposed encryption algorithm is secure and can protect information and minimize the risk of data breaches and cyber attacks.

To verify the effectiveness and reliability of the proposed mechanism, we conducted a series of experiments and analyzed the results using various methods such as statistical attack analysis, differential attack analysis, brute force attack analysis, and information entropy analysis. The results of these tests demonstrate that the proposed system is highly secure and efficient and meets all the requirements necessary for protecting information stored using images. Overall, our proposed image encryption mechanism represents a significant advancement in the field and offers a reliable and effective way to secure sensitive information.

Author Contributions: Conceptualization, D.E.M. and Y.X.; data curation, D.E.M., X.F., and X.W.; formal analysis, D.E.M. and X.F.; funding acquisition, X.F.; investigation, D.E.M. and X.W.; methodology, D.E.M.; project administration, X.W.; resources, D.E.M.; software, D.E.M.; supervision, X.F. and X.W.; validation, D.E.M. and X.W.; visualization, D.E.M. and X.F.; writing—original draft, D.E.M.; writing—review and editing, D.E.M. and Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (No: 61672124), Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund (No: MMJJ20170203), Liaoning Province Science and Technology Innovation Leading Talents Program Project (No: XLYC1802013), Key R&D Projects of Liaoning Province (No: 2019020105-JH2/103), and Jinan City 20 Universities Funding Projects Introducing Innovation Team Program (No: 2019GXRC031).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available on request.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Algorithm A1. Encryption Pseudo-code.

```
% Read an image from file
P = imread('my_image.png');
% Perform the shift rows operation on the image
shifted_image = shift_rows(P);
function shifted_image = shift_rows(P)
% Shift rows operation for an image
% Determine the dimensions of the input image
[height, width, channels] = size(P);
% Check that the image can be divided into 4 × 4 blocks
if mod(height, 4) ~= 0 || mod(width, 4) ~= 0
    error('Image dimensions are not divisible by 4.');
end
% Initialize the shifted image
P = zeros(size(P));
% Iterate through each 4 × 4 block in the image
for i = 1:4:height
    for j = 1:4:width
        % Extract the current 4 × 4 block
        block = P(i:i + 3, j:j + 3, :)
        % Shift each row to the left by its index minus 1
        for k = 1:4
            block(k,:) = circshift(block(k,:), [0, − (k − 1)]);
        end
        % Store the shifted block in the output image
        P(i:i + 3, j:j + 3, :) = block;
    end
end
For each odd column i:
    Let top_value_i = P [1][i]
    For each even column j corresponding to column i:
        If j > 1:
            For each row k:
                P[k][j] = P[k][j] XOR top_value_i
For each even column i:
    Let top_value_i = P [1][i]
    For each odd column j corresponding to column i:
        If j > 1:
            For each row k:
                P[k][j] = P[k][j] XOR top_value_i
% Divide the image into 4 × 4 blocks
H' = mat2cell(P, 4*ones(1,size(P,1)/4), 4*ones(1,size(P,2)/4));
% Encrypt each block using the Hill Cipher algorithm
for i = 1:size(H',1)
```

Algorithm A1. *Cont.*

```
    for j = 1:size(H',2)
        % Convert the block into a column vector
        v = reshape(H'{i,j}, [], 1);
        % Compute the product of the key matrix and the column vector
        c = mod(K*v, 256);
        % Convert the resulting column vector back into a 4 × 4 block
        K1{i,j} = reshape(c, size(H'{i,j}));
    end
end
% Concatenate the encrypted blocks to form the encrypted image
E = cell2mat(K2);
for i = 1:n
        % Generate new secret key value from sigmoid-logistic map
        x = r*x/(1 + e^x)*(1 − x/(1 + e6^x));
        k2(i) = uint8(255*x); % Scale to 8-bit unsigned integer (0–255)
end
K4 = bitxor(K1, k2);
// Apply Kronecker xor transformation to expand matrix K4 to size N^2 x N^2
K5= kron(K4, ones(N, N));
// Shift rows of M by NTH positions
K5= circshift(K5, [N, 0]);
// Apply bitwise exclusive OR operation to each element of M with other elements of M, keeping its value unchanged
for i = 1:N^2
    for j = 1:N^2
        K5(i,j) = bitxor(K5(i,j), K5(mod(i + NTH-1, N^2) + 1, mod(j + NTH-1, N^2) + 1));
    end
end
K3 = zeros(size(K5));
for i = 1:numel(K5)
    x = r*x/(1 + exp(x))*(1 − x/(1 + exp(e^x)));
K3(i) = mod(round(x*256), 256); % modulo 256 to match the image data
end
% Load the K5 matrix
load('K5.mat'); % assuming K5 is saved in a .mat file
% XOR K5 with the sigmoid logistic keys k3
ciphertext = bitxor(K5, K3);
```

Algorithm A2. Decryption pseudo-code.

```
Input image C
M->xor(C,key2)
function M = inverseKroneckerxor(matrix M, int I)
    M = shiftRowsUp(M, I)
    M = contractMatrix(M)
    for i = 1 to M.numRows
        for j = 1 to M.numColumns
            M[i][j] = M[i][j] XOR M[i][j − 1] xor M[i − 1][j] xor M[i − 1][j − 1]
    return K4
end function
numkeys = K4
function numKeys = inverseGenerateSecretKeys(secretKeys, numKeys)
  % Undo bitwise xor with secretKeyElements
  secretKeyElements = [1, 2, 3, 4];
  keys = xor(secretKeys, secretKeyElements);
  % Use inverse logisticMap function to generate numKeys
  numKeys = inverseLogisticMap(keys);
```

Algorithm A2. *Cont.*

```
end
state = numkeys;
for i = 1:size(state, 1)
      for j = 1:size(state, 2)
        if i == 1 && j == 1 % Skip top-left value
          continue
        end
        if i == 1 % Handle top row
          if mod(j, 2) == 0 % Even column
          state(i, j) = xor(state(i, j), state(i, j − 1));
        else % Odd column
           state(i, j) = xor(state(i, j), state(i + 1, j));
        end
      else % Handle all other values
        if mod(j, 2) == 0 % Even column
          state(i, j) = xor(state(i, j), state(i, j − 1));
        else % Odd column
          state(i, j) = xor(state(i, j), state(i−1, j));
        end
      end
    end
  end
end
function state = inverseShiftRows(state)
  for i = 2:n-size % Shift rows 2-n-size
P(i,:) = circshift(state(i,:), [0, −(i − 1)]);
  end
end
```

Table A1. Sample vector.

IMAGE	SIZE	NPCR	UACI	ENTROPY	PSNR	MSE
LENA	32 × 32	99.69120	33.41020	7.99650	8.93950	94.56743
	64 × 64	99.62130	33.44530	7.99780	8.43100	94.24519
	128 × 128	99.60730	33.44770	7.90040	8.79068	96.54386
	256 × 256	99.6891	33.32720	7.89930	8.08182	98.56431
CAMERAMAN	32 × 32		33.34790	7.90010	8.17186	105.90900
	64 × 64	99.58610	33.35810	7.99936	8.85879	105.99800
	128 × 128	99.69970	33.47560	7.90560	8.37410	107.85600
	256 × 256	99.99970	33.49106	7.90060	8.20933	107.98750
PEPPER	32 × 32	99.58810	33.41230	7.99910	8.14837	94.67875
	64 × 64	99.62570	33.34992	7.99540	8.55323	94.89750
	128 × 128	99.61949	33.49137	7.99610	8.96133	94.21672
	256 × 256	99.60714	33.45404	7.99730	8.43100	94.24519
BABOON	32 × 32	99.57810	33.35140	7.89970	9.37974	99.90874
	64 × 64	99.62429	33.48444	7.86610	9.43248	100.19087
	128 × 128	99.63179	33.31653	7.84320	9.27922	98.98765
	256 × 256	99.61445	33.46576	7.85460	9.12840	99.99876
RICE	32 × 32	99.62432	33.30754	7.99930	9.02900	107.87650
	64 × 64	99.63109	33.42864	7.99540	9.39561	105.98760
	128 × 128	99.60689	33.40145	7.90870	9.35469	101.15210
	256 × 256	99.62870	33.40266	7.99890	9.69764	105.90800
HOUSE	32 × 32	99.63182	33.32305	7.91681	9.66016	133.87900
	64 × 64	99.61221	33.48850	7.92955	9.88513	128.02300
	128 × 128	99.62649	33.90039	7.96809	9.22140	130.14300
	256 × 256	99.60129	33.59067	7.90526	9.32130	129.85400

References

1. Ramasamy, P.; Ranganathan, V.; Kadry, S.; Damaševičius, R.; Blažauskas, T. An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using Enhanced Logistic-Tent Map. *Entropy* **2019**, *21*, 656. [CrossRef] [PubMed]
2. Hua, Z.; Jin, F.; Xu, B.; Huang, H. 2D Logistic-Sine-coupling map for image encryption. *Signal Process.* **2018**, *149*, 148–161. [CrossRef]
3. Hénon, M. Numerical study of quadratic area-preserving mappings. *Q. Appl. Math.* **1969**, *27*, 291–312. [CrossRef]
4. Phatak, S.C.; Rao, S.S. Logistic map: A possible random-number generator. *Phys. Rev. E* **1995**, *51*, 3670–3678. [CrossRef] [PubMed]
5. Ravichandran, D.; Banu, S.A.; Murthy, B.; Balasubramanian, V.; Fathima, S.; Amirtharajan, R. An efficient medical image encryption using hybrid DNA computing and chaos in transform domain. *Med Biol. Eng. Comput.* **2021**, *59*, 589–605. [CrossRef] [PubMed]
6. Pathak, B.; Pondkule, D.; Shaha, R.; Surve, A. Visual Cryptography and Image Processing Based Approach for Bank Security Applications. In Proceedings of the Second International Conference on Computer Networks and Communication Technologies, Coimbatore, India, 23–24 May 2019; Smys, S., Senjyu, T., Lafata, P., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 292–298. [CrossRef]
7. Valandar, M.Y.; Ayubi, P.; Barani, M.J. A new transform domain steganography based on modified logistic chaotic map for color images. *J. Inf. Secur. Appl.* **2017**, *34*, 142–151. [CrossRef]
8. Arab, A.A.; Rostami, M.J.B.; Ghavami, B. An image encryption algorithm using the combination of chaotic maps. *Optik* **2022**, *261*, 169122. [CrossRef]
9. Babu, B.M. Image Encryption Using Chaotic Maps and DNA Encoding. *J. Xidian Univ.* **2020**, *14*. [CrossRef]
10. Cun, Q.; Tong, X.; Wang, Z.; Zhang, M. Selective image encryption method based on dynamic DNA coding and new chaotic map. *Optik* **2021**, *243*, 167286. [CrossRef]
11. Zhu, X.; Liu, H.; Liang, Y.; Wu, J. Image encryption based on Kronecker product over finite fields and DNA operation. *Optik* **2020**, *224*, 164725. [CrossRef]
12. Wang, X.; Su, Y. Image encryption based on compressed sensing and DNA encoding. *Signal Process. Image Commun.* **2021**, *95*, 116246. [CrossRef]
13. Zhang, D.; Liao, X.; Yang, B.; Zhang, Y. A fast and efficient approach to color-image encryption based on compressive sensing and fractional Fourier transform. *Multimed. Tools Appl.* **2017**, *77*, 2191–2208. [CrossRef]
14. Sun, X.; Shao, Z.; Shang, Y.; Liang, M.; Yang, F. Multiple-image encryption based on cascaded gyrator transforms and high-dimensional chaotic system. *Multimed. Tools Appl.* **2021**, *80*, 15825–15848. [CrossRef]
15. Wang, X.; Gao, S. Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. *Inf. Sci.* **2020**, *507*, 16–36. [CrossRef]
16. Essaid, M.; Akharraz, I.; Saaidi, A.; Mouhib, E.A. Image encryption scheme based on a new secure variant of Hill cipher and 1D chaotic maps. *J. Inf. Secur. Appl.* **2019**, *47*, 173–187. [CrossRef]
17. Duan, H.; Pang, X. A multivariate grey prediction model based on energy logistic equation and its application in energy prediction in China. *Energy* **2021**, *229*, 120716. [CrossRef]
18. Ridout, M.S.; Cole, D.J.; Morgan, B.J.T.; Byrne, L.J.; Tuite, M.F. New Approximations to the Malthusian Parameter. *Biometrics* **2006**, *62*, 1216–1223. [CrossRef] [PubMed]
19. Wang, J.; Liu, L.; Xu, M.; Li, X. A novel content-selected image encryption algorithm based on the LS chaotic model. *J. King Saud Univ. - Comput. Inf. Sci.* **2022**, *34*, 8245–8259. [CrossRef]
20. Wang, X.; Liu, C.; Jiang, D. Visually meaningful image encryption scheme based on new-designed chaotic map and random scrambling diffusion strategy. *Chaos, Solitons Fractals* **2022**, *164*, 112625. [CrossRef]
21. Wang, X.; Wang, X.; Teng, L.; Jiang, D. A novel meaningful image encryption algorithm based on newly-designed coupled map lattice and adaptive embedding. *Optik* **2022**, *270*, 170073. [CrossRef]
22. Teng, L.; Wang, X.; Xian, Y. Image encryption algorithm based on a 2D-CLSS hyperchaotic map using simultaneous permutation and diffusion. *Inf. Sci.* **2022**, *605*, 71–85. [CrossRef]
23. Datta, L. A Survey on Activation Functions and their relation with Xavier and He Normal Initialization. *arXiv* **2020**, arXiv:2004.06632. [CrossRef]
24. Henderson, H.; Pukelsheim, F.; Searle, S.R. On the history of the kronecker product. *Linear Multilinear Algebra* **1983**, *14*, 113–120. [CrossRef]
25. Christensen, C. Lester Hill Revisited. *Cryptologia* **2014**, *38*, 293–332. [CrossRef]
26. Hill, L.S. Cryptography in an Algebraic Alphabet. *Am. Math. Mon.* **1929**, *36*, 306. [CrossRef]
27. Hidayat, T.; Mahardiko, R. A Systematic Literature Review Method On AES Algorithm for Data Sharing Encryption On Cloud Computing. *Int. J. Artif. Intell. Res.* **2020**, *4*, 49–57. [CrossRef]
28. Carlson, A.; Gang, G.; Gang, T.; Ghosh, B.; Dutta, I.K. Evaluating True Cryptographic Key Space Size. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 1–4 December 2021; IEEE: New York, NY, USA, 2021; pp. 0243–0249. [CrossRef]
29. Wang, X.; Liu, H. Cross-plane multi-image encryption using chaos and blurred pixels. *Chaos, Solitons Fractals* **2022**, *164*, 112586. [CrossRef]

30. Zhang, X.; Wang, X. Multiple-image encryption algorithm based on DNA encoding and chaotic system. *Multimed. Tools Appl.* **2018**, *78*, 7841–7869. [CrossRef]
31. Zhou, N.; Yan, X.; Liang, H.; Tao, X.; Li, G. Multi-image encryption scheme based on quantum 3D Arnold transform and scaled Zhongtang chaotic system. *Quantum Inf. Process.* **2018**, *17*, 338. [CrossRef]
32. Zhang, X.; Wang, X. Multiple-image encryption algorithm based on mixed image element and chaos. *Comput. Electr. Eng.* **2017**, *62*, 401–413. [CrossRef]
33. Zhao, H.; Wang, S.; Wang, X. Fast image encryption algorithm based on multi-parameter fractal matrix and MPMCML system. *Chaos, Solitons Fractals* **2022**, *164*, 112742. [CrossRef]
34. Patro, K.A.K.; Acharya, B. A novel multi-dimensional multiple image encryption technique. *Multimed. Tools Appl.* **2020**, *79*, 12959–12994. [CrossRef]
35. Zhou, S.; Wang, X.; Zhang, Y.; Ge, B.; Wang, M.; Gao, S. A novel image encryption cryptosystem based on true random numbers and chaotic systems. *Multimed. Syst.* **2021**, *28*, 95–112. [CrossRef]
36. Zheng, J.; Zeng, Q. The unified image encryption algorithm based on composite chaotic system. *Multimed. Tools Appl.* **2022**, 1–20. [CrossRef]
37. Wang, X.; Zhang, M. An image encryption algorithm based on new chaos and diffusion values of a truth table. *Inf. Sci.* **2021**, *579*, 128–149. [CrossRef]
38. Zhang, Y.-Q.; Hao, J.-L.; Wang, X.-Y. An Efficient Image Encryption Scheme Based on S-Boxes and Fractional-Order Differential Logistic Map. *IEEE Access* **2020**, *8*, 54175–54188. [CrossRef]
39. Gao, X. Image encryption algorithm based on 2D hyperchaotic map. *Opt. Laser Technol.* **2021**, *142*, 107252. [CrossRef]
40. Srivastava, R.; Singh, O.P. Performance Analysis of Image Encryption Using Block Based Technique. *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.* **2015**, *4*, 4266–4271.