Journal of
Sensor and Actuator Networks

MDPI

Article

A Lightweight Hybrid Scheme for Hiding Text Messages in Colour Images Using LSB, Lah Transform and Chaotic Techniques

Iman Qays Abduljaleel [1], Zaid Ameen Abduljabbar [2,3,4], Mustafa A. Al Sibahee [5,6,*], Mudhafar Jalil Jassim Ghrabat [7], Junchao Ma [5,*] and Vincent Omollo Nyangaresi [8]

1 Department of Computer Science, College of Computer Science and Information Technology, University of Basrah, Basrah 61004, Iraq
2 Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq
3 Technical Computer Engineering Department, Al-Kunooze University College, Basrah 61001, Iraq
4 Shenzhen Institute, Huazhong University of Science and Technology, Shenzhen 430074, China
5 College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China
6 Computer Technology Engineering Department, Iraq University College, Basrah 61004, Iraq
7 Design and IoT Lab, Al-Turath University College, Baghdad 10013, Iraq
8 Faculty of Biological & Physical Sciences, Tom Mboya University, Homabay 40300, Kenya
* Correspondence: mustafa.alsibahee@iuc.edu.iq (M.A.A.S.); majunchao@sztu.edu.cn (J.M.)

Abstract: Data security can involve embedding hidden images, text, audio, or video files within other media to prevent hackers from stealing encrypted data. Existing mechanisms suffer from a high risk of security breaches or large computational costs, however. The method proposed in this work incorporates low-complexity encryption and steganography mechanisms to enhance security during transmission while lowering computational complexity. In message encryption, it is recommended that text file data slicing in binary representation, to achieve different lengths of string, be conducted before text file data masking based on the lightweight Lucas series and mod function to ensure the retrieval of text messages is impossible. The steganography algorithm starts by generating a random key stream using a hybrid of two low-complexity chaotic maps, the Tent map and the Ikeda map. By finding a position vector parallel to the input image vector, these keys are used based on the previously generated position vector to randomly select input image data and create four vectors that can be later used as input for the Lah transform. In this paper, we present an approach for hiding encrypted text files using LSB colour image steganography by applying a low-complexity XOR operation to the most significant bits in 24-bit colour cover images. It is necessary to perform inverse Lah transformation to recover the image pixels and ensure that invisible data cannot be retrieved in a particular sequence. Evaluation of the quality of the resulting stego-images and comparison with other ways of performing encryption and message concealment shows that the stego-image has a higher PSNR, a lower MSE, and an SSIM value close to one, illustrating the suitability of the proposed method. It is also considered lightweight in terms of having lower computational overhead.

Keywords: steganography; cryptography; Lah transform; LSB; chaotic map; text embedding

Citation: Abduljaleel, I.Q.; Abduljabbar, Z.A.; Al Sibahee, M.A.; Ghrabat, M.J.J.; Ma, J.; Nyangaresi, V.O. A Lightweight Hybrid Scheme for Hiding Text Messages in Colour Images Using LSB, Lah Transform and Chaotic Techniques. *J. Sens. Actuator Netw.* **2022**, *11*, 66. https://doi.org/10.3390/jsan11040066

Academic Editor: Mohamed Amine Ferrag

Received: 26 August 2022
Accepted: 19 September 2022
Published: 17 October 2022

1. Introduction

The conversion of data to digital representations such as images, text, or audio files increases the possibility of data interchange through the internet. However, there are issues with respect to maintaining data confidentially, as it is possible to guess the secrecy protocols for data when using various electronic hacking techniques. Work on several strong and secure strategies for preserving information is thus ongoing [1,2]. Watermarking, steganography, and cryptography are currently among the most popular security strategies [3,4]. The purpose of watermarking is to safeguard the copyright of digital material owners [4], whereas steganography refers to the ability to transmit unseen secret data by embedding

it in a multimedia carrier, and the goal of steganography is to conceal the existence of embedded data [5]. Cryptography refers to techniques used to hide the data by turning it into a seemingly meaningless or confusing form, generally achieved in a reversable manner by the application of mathematical theories and computational intelligence [6,7].

Steganography conceals both confidential data and the relationship between the sender and the recipient, whereas watermarks merely hide confidential data [8]. In some cryptography techniques, the transmitter and the recipient are also identified, as the primary purpose is to secure the substance of the private data [4]. A combination of cryptography and steganography, however, can result in a more secure platform [9–11].

Technical steganography in images may be categorised based on the domain in which it is used, whether spatial or frequency [3,12,13]. Pixel values are employed to obscure secret information in the spatial domain, whereas in the frequency domain, message encapsulation is performed within the altered image. Several transforms, such as the Fourier and Wavelet transforms [3,14], may also be added. Steganography thus offers a robust and excellent tool for secure communication, particularly when used with encryption [8], and steganography is now frequently utilised with text, image, sound, and video data depending on the cover file.

Image files are the most commonly utilised medium for concealment, due to their large capacity and easy availability via the internet [15,16]. Digital colour images offer more capability than greyscale or black-and-white images. The values associated with RGB colours for each pixel level, such as their hue, brightness, and saturation, may be successful when employed in various image processing applications [17,18]. In steganography, the media carrier is involved in the cover object, so the quantity of concealed information is related to the payload capacity [19]. The cover object is limited by the number of hidden secret messages and the message's imperceptibility and robustness. A stego-image is employed to conceal a hidden message under such a cover image [20] and, usually, the length of hidden messages that may be placed in digital photographs is thus unlimited. Nevertheless, secret messages that are overly lengthy are difficult to incorporate in small images, and when the message length is incompatible with the image size, problems with the dynamics of covert communications may occur [21].

Several factors influence the severity of problems associated with steganography techniques [22]:

1. Invisibility, which denotes that the stego-image resembles the original image;
2. Payload/capacity, which indicates the quantity of hidden data that can be held private in the cover medium;
3. Robustness against statistical assaults, represented by the peak signal-to-noise ratio (PSNR) and mean square error (MSE), fidelity measures used to assess robustness against statistical attacks;
4. Computation complexity, which is a calculation of the cost of embedding and extracting a secret message.

The suggested approach in this work thus takes these factors into account, and a variety of measurements were made with respect to these points, as shown in Section 5.12 Maximum Difference, in order to assess the compatibility between the original image and the stego-image in addition to the suitability of the stego-image's capacity and its robustness against statistical attacks. Compared with the algorithm times for hiding and recovery shown in Section 5.15 Implementation Time, reduced computing complexity was also noted.

As a result of this work, we propose a method for encrypting text files that uses a suggested methodology based on low-complexity Lucas series and mod operation to scramble data in a binary representation. The recommended technique for working on values in binary format is to segment the value vector into tiny vectors of varying random lengths to make it difficult to determine the intended distance of each vector worked on to that point. Each subvector value is shifted using the Lucas series, transferring their locations, before the subvectors are combined into a single vector and conveyed from binary

to decimal representation and then finally into ASCII characters. As a result, the encrypted data are embedded in the three channels of a colour image based on the random locations assigned by the hybrid chaos map (made of the Tent map and Ikeda map). It is important to note that the embedded values are not held directly in the image pixels but, instead, in the values resulting from sending the colour image pixels on the Lah transform. This makes it challenging for any attacker to determine the secret value location, in addition to making retrieval of the value itself complex.

This study contributes to ciphertext security by applying a basic scrambling algorithm based on applying unequal splits to the value vector before sending it to the Lucas series for encoding and relocation of the value vector. Furthermore, the cover image pixels are not used to hide the encrypted data, which are instead passed to the Lah transform and then hidden within the parameters to ensure that the concealed values are difficult to recover while still preserving the cover image. With respect to the applied steganography techniques, reliance on two low-complexity types of chaos allows the proposed hiding algorithm to be lightweight, facilitating the selection of the cover image pixels at random to hide the ciphertext bits using LSB technology. The LSB depends on two bits being concealed for each pixel in the colour image according to the set of conditions, ensuring that the hidden values are difficult to retrieve. As a result, the approach provided in this work is distinguished by its exceptional simplicity arising from dependence on an enhanced mix of three methods (chaos, Lah transform, and LSB), as shown in its low MSE values and the minor differences between cover and stego-images. The integration of encryption and hiding information methods with low computational costs allows the development of a method with as little complexity as possible, whereas the integration process increases the security efficiency of the proposed method. The proposed scheme thus provides secure lightweight cryptography and steganography methods in terms of lowering the overall computational cost, whereas the PSNR and maximum capacity values show that the approach also preserves more confidential data while displaying virtually equal image quality to that of the original.

The rest of this paper is presented according to the following sections: Section 2 presents a survey of existing approaches proposed in steganography, and Section 3 fully outlines the proposed methods. Section 4 then discusses the results and offers a performance evaluation. To end, Section 5 presents the conclusions based on the examination of the proposed method.

2. Related Work

There are various techniques used for implementing steganography methods into cover colour images, and several studies have attempted to investigate the balance between the amount of data, such as embedded text, audio, and image files and the protection offered to a secret message while maintaining image resolution [23].

Researchers in [3] considered a steganography technique for embedding in less than two significant bits of each pixel in JPEG images. A replacement table was used to guide the embedding process, with the pixel bits increased or decreased based on the bit values in the embedded message. The PSNR and maximum capacity of steganography results indicated this method retains more secret data while maintaining nearly identical image quality to the original. This method creates a table based on both the cover image and private data in a manner that consumes time during the embedding or de-embedding process and follows laws that can be anticipated.

The work in [12] combined AES, LSB, and the Fisher–Yates shuffle algorithm to create a new method for colour image steganography that is possibly more efficient. A codeword was initially formed with confidential data and its CRC-32 checksum calculated; following that, it was compressed before being encrypted using AES. Finally, it was added to encrypted header information for further processing before being embedded in the cover image. When embedding encrypted data and header information, the Fisher–Yates shuffle method was used to determine the next pixel location. A different LSB from each colour

channel of the selected pixel was used to hide each separate byte. Based on a colour depth of at least 24 bits per pixel, the algorithm embedded eight bits in each cover image pixel by hiding three bits in the red plane, three bits in the blue plane, and two bits in the green plane. It used a high and sequential modulation rate to embed these bits, and its dependency on compression data hiding enables an increase in the payload capacity inside a colour image; nevertheless, increasing the quantity of embedded data decreases the accuracy of the cover image due to changes in two and three locations for cover image pixels.

Researchers in [24] suggested a method that combines the LSB method and XOR operations to embed secret text messages into 24-bit colour cover images. This resulted in two proposed schemes: an embedding scheme, which uses LSB and triple XOR operation techniques on the binary value of MSB, and an extraction scheme using the same techniques. However, while their statistical analysis showed that the algorithms produced good MSE and PSNR values, the study did not clarify the amount of data hidden in the images and, as a result, the values of statistical measures are unclear. The proposed hiding technique is also relatively simple, relying on the original text and the triple XORed operation interactions between the sequential bits of the cover image pixels, making the hidden data accessible under some circumstances.

The authors in [25] developed a YCbCr colour model based on 2-bit XOR LSB image steganography. The methodology used offers a very safe technique for data concealment in the spatial field for image steganography, in which an image is modified from RGB colour space to YCbCr colour space, with invisible data hidden inside the Cr colour space component using 2-bit XOR. The work applied a simple crypto algorithm to a text file to generate a copy in an unreadable format before disguising the resulting data in an image. The rail fence cipher algorithm was used for text file encryption and decryption, and the PSNR analysis value reached 40 dB, indicating that the stego-image was of the highest quality. However, within this study, the PSNR analysis was the only system efficiency metric evaluated, illustrating both the algorithm's simplicity and its dependency on changing the colour space from RGB to YCbCr.

The ASCII mapping technique was applied in Ref. [22] to create an encoding table. The process began by mapping the text message to match some bits with those of the cover image. Each character in the text file was thus mapped and the two bits of binary stream data were each compared to the same two bits in the image. Locking the image locations then resulted in a distribution of every two bits of the character across the same bits of the image, awarding each character of the text file a new substitution value; in this way, the task of steganography to reconstruct the message was made more difficult. This method was tested and found to have a low computational cost; however, it only works with greyscale cover images.

The purpose of Ref. [26] was to improve cryptography security by simultaneously utilising steganography. The process begins by encrypting the original texts with the RSA algorithm, followed by hiding the encrypted texts in image files using the LSB technique. As an image is then used as a cover file, a large quantity of data can be embedded, whereas resistance to external attacks is increased; however, as this algorithm is a direct application of the RSA algorithm and LSB technology based on hiding a single bit in each cover image pixel, it is a high complexity technique.

In [27], a combination of the RC4 algorithm and LSB cryptography was used. The ciphertext encrypted with RC4 was distributed to an image using PRNG, and the result of combining the ciphertext and the cover image was used as the stego-image. Rendering the stego-image in BITMAP format demonstrates that this method outperforms the use of a stego-image in JPEG format. As the number of ciphertext characters in a stego-image increased, the processing time for encoding and decoding the hidden text grew, however, to such an extent that combining the RC4 algorithm with LSB technology was a failure in terms of processing effort.

To increase the security of their technique, researchers in [28] used his colour model to hide secret messages inside colour images. This novel approach achieved a high average

PSNR, demonstrating the improved efficiency of the proposed method over existing methods. Working with this algorithm necessitates more time spent switching from RGB to HIS colour, however.

In [29], a dual protection method for text message transfer and encrypted communications was proposed using a CRT steganography algorithm, with an RC4 cryptographic algorithm. The MSE, PSNR, and SSIM test results showed that the output quality of the stego CRT image remained perfectly maintained even when the message was set to maximum size. The text file was also successfully extracted. However, this method can only be applied to greyscale images.

The problems experienced by related work in this field include high computation costs and concomitant increases in time that have significantly affected their efficient application. In addition, the effectiveness of such systems declines as both the cover images and the amount of data hidden in them grow. Further, as well as significant dependency on the use of greyscale images as a database rather than colour images during the steganography stage, such work has also suffered from other security problems. This study addresses both of these points by dividing the work into two security phases—cryptography and steganography—to thwart hacking attempts and make data retrieval more difficult. This process helps with the secure encryption of text by applying a basic scrambling model based on the uneven division of the value vector as well as decreasing the computational complexity during the encryption stage by employing the lightweight Lucas series.

In the hiding stage, Lah transformation was used to hide ciphertext in the cover image, ensuring that the hidden values are difficult to retrieve without affecting the cover image. Moreover, utilising LSB technology, lightweight XOR operations, and two low-complexity chaotic keys, the cover image pixels used to hide ciphertext bits were randomly selected, further blocking detection. By lowering the computing complexity of the hiding stage by employing these distinct strategies and numerous safe and low-complexity tools, there was significant enhancement of the recommended system.

3. Basic Concepts

3.1. Chaotic Map

A chaotic map function indicates a type of development exhibiting chaotic and random behaviour that is therefore not predictable while still being characterised by a type of regularity and ergodicity that works to prevent point repetition from occurring in the search ranges caused by the random sequence [30]. Chaos theory is applied not only in steganography but also in many encryption schemes due to its sensitivity to initial conditions: in a chaotic system, even a small change in initial conditions will result in a totally different—and, importantly, unpredictable—output [13].

Various different chaotic maps are used in image cryptography, and more details about those used in this paper are offered in the following subsections.

3.1.1. Tent Map

The chaotic tent map is defined as follows [31,32]:

$$Z_{i+1} = \tau Z_i, \; if \; Z_i < 0.5, \tag{1}$$

$$Z_{i+1} = \tau(1 - Z_i), \; else. \tag{2}$$

where $Z_i \in [0,1], \; for \; i \geq 0$. There is only one control parameter , τ, and $\tau \in [0,2]$. In this case, Z_0 represents the system starting point, whereas $Z_0, Z_1, \ldots, Z_n$ is a set of real values defined as the orbit of the system. For each Z_0, an orbit is attained and, based on the value of τ, Equation (2) exhibits dynamic features that range from predictable to chaotic. Lyapunov's value is greater than 0. A tent system is based on chaos theory. There are no problems with state traversal or certainty in the output signal [31].

3.1.2. Ikeda Map

The Ikeda map was developed as a light model across an optical resonator [33]. Due to its high complexity and other notable chaotic properties, it is, however, particularly beneficial for cryptography. The Ikeda map is a discrete-time nonlinear system with the formula [34]:

$$X_{i+1} = 1 + \gamma(X_i cosp_i - Y_i sinp_i) \quad (3)$$

$$Y_{i+1} = \gamma(X_i sinp_i - Y_i cosp_i) \quad (4)$$

where

$$p_i = 0.4 - \frac{6}{(1 + X_i^2 + Y_i^2)} \quad (5)$$

Here, X_i, Y_i, and p_i represent the state variables. A chaotic sequence occurs when γ is between 0.5 and 0.95, where γ represents an equation parameter. Ikeda map is unpredictable, nonlinear, and sensitive to beginning settings, making it a chaotic system suitable for use in encryption [33,34].

3.1.3. The Proposed Hyperchaotic Map

This section proposes a method for creating a new hybrid chaotic map. In this encryption technique, the novel hybrid chaotic map, which is created from two traditional chaotic maps, the tent map and the Ikeda map, displays exceptional sensitivity to the initial condition and parameters compared with their Lyapunov exponents.

To develop a new compound mapping system, Equations (1) to (5) are thus combined, with some modifications. A brief description of the new hybrid mapping system is thus as follows:

1. Using Equations (1) and (2), generate N keys using the Tent map;
2. Rather than using the Y values in equation 4, apply the Tent map sequences as input to the Ikeda map to create its series;
3. Use a distribution of values in the interval [0, 1];

Using this structural system, more sensitive chaotic maps can be generated than previously, potentially eliminating the threat of attack based on a known technique.

Remark 1. *This work noted that the Tent map produces chaos within a specified range, with the control parameter t having a value range of only [0, 1]. This paper thus employed the Ikeda map with the Tent map as an inner parameter replacing the y series (Equation (4)). A more complex topology and more parameters were found in the resulting 2D Ikeda map, which are likely to improve the encoding quality by expanding the Lyapunov and enhancing its image encoding ability. The algorithm thus incorporates a 2D Ikeda map in the proposed hyperchaotic map.*

3.2. Lah Transform

The transform element set $\{Q_0, Q_1, Q_2, \ldots, Q_m\}$ of the s-pixel group $s_0, s_1, s_2, \ldots, s_n$ of the cover image with encryption was calculated using the association seen in Equation (6) [35]:

$$Q_n = \sum_{u=0}^{n} \frac{n!}{u!} \left(\frac{n-1}{n-u}\right) s_k \quad (6)$$

where $0 < n < s - 1$ for all n. The image is then divided into four blocks, and solutions found for each. The relationship in Equation (7) is then used to measure the inverse Lah transformation [35].

$$s'_k = \sum_{u=0}^{n} \frac{n!}{u!} (-1)^{n-k} \left(\frac{n-1}{n-u}\right) w_n \quad (7)$$

In the condition of no hiding, all recomputed pixel parameters are similar to the pixel values before performing the Lah transform, implying that the inverse Lah transformation of stego-image pixels equals the cover image pixels.

Remark 2. *The main idea behind using the Lah transform is to apply simple calculations to produce an integer polynomial series in coefficient notation from actual pixel values. The Lah transform computation is integer-based, rather than floating-point-based, as well as being quicker in execution. Furthermore, the Lah transform does not produce complex output values. This allows ciphertext data to be hidden in the indirect parameters resulting from the pixels of the cover image.*

3.3. Least Significant Bit (LSB)

The LSB process is a form of image steganography widely applied in the spatial domain. When embedding data in a cover image, the least significant bit (8 bits) is the bit at the far right with the least significant value, and the most significant bit (MSB) is the bit at the far left with the most significant value of the file [11,24,36]. If the MSB value is even slightly modified, this might generate an image different to the original: altering the LSB value, instead, has a far less obvious effect on the MSB outcome [10,24]. Placing a hidden message in an image's 8-pixel LSB can thus minimize any change in colour or visual image loss to avoid arousing suspicion from the observer [21]. However, the more LSB bits utilised, the poorer the stego-image and the more stego-retrieved reserve data remain [37].

Remark 3. *The use of fewer LSB bits in the proposed method makes distinguishing between the cover image and the stego-image extremely difficult. However, one of the problems of LSB is the need to alter a high number of bits in the cover image. This work attempted to preserve the cover image with the minimum distortion possible by employing a test relationship between other pixel bits to ensure the maximum change in a nonsequential manner. Importantly, the cover image was much larger than the size of the hidden ciphertext to facilitate this.*

3.4. Lucas Series

The Lucas number sequence was developed in the 1870s when French mathematician Edouard Lucas was investigating linear recursive sequences [38]. The Lucas sequence is defined as the sum of the two immediately preceding terms. $N_0 = 2$ and $N_1 = 1$ are the first two Lucas numbers, and the Lucas numbers are thus defined as follows [39,40].

$$N_l = \begin{cases} 2 \; if \; l = 0, \\ 1 \; if \; l = 1, \\ N_{l-1} + N_{l-2} \; if \; l > 1. \end{cases} \tag{8}$$

The beginning conditions of the Lucas series are 2 and 1 if l equals 0 or 1. The Lucas sequence is therefore 2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 189, ...

Remark 4. *An unrepeated random Lucas series can be used to scramble the positions of the subvector values into which the text vector in its binary representation is divided. The series' usage of integer numbers is then advantageous for more readily adjusting the locations of the values in each subvector.*

4. Proposed Algorithms

A steganography mechanism enables encrypted secret data to be transmitted. Typically, a digital image comprises disparate visual pieces known as pixels. In this work, the cover image is a colour image. Therefore, each image is represented by three arrays of comparable size, each containing several bytes. This work is dependent on a hybrid technique for concealing encrypted text messages in a colour image that begins with a vector segmentation stage, which provides the binary representation of the text file's values for many vectors of varying lengths. Before these are integrated, the vectors are independently blended based on the Lucas series, and the resulting bits are then converted to ASCII code and stored in a text file. The concealment of the encrypted text file begins with the production of a random integer of values based on a hyperchaotic map (combined Tent map and Ikeda map). This hyperchaotic map value determines the location in which the text data is to be hidden in a nonsequential manner. The encoded text data are then embedded

into the image bits using the Lah transformation and the LSB algorithm. Based on the above, we can see that the proposed strategy relies on the Lah transform, hyperchaotic map, and modified LSB. Text information during network communications is considerably protected when combined with the proposed technique and encryption. The steps given below describe this set of procedures in more detail, and Figure 1 illustrates the proposed steganographic schema diagram.

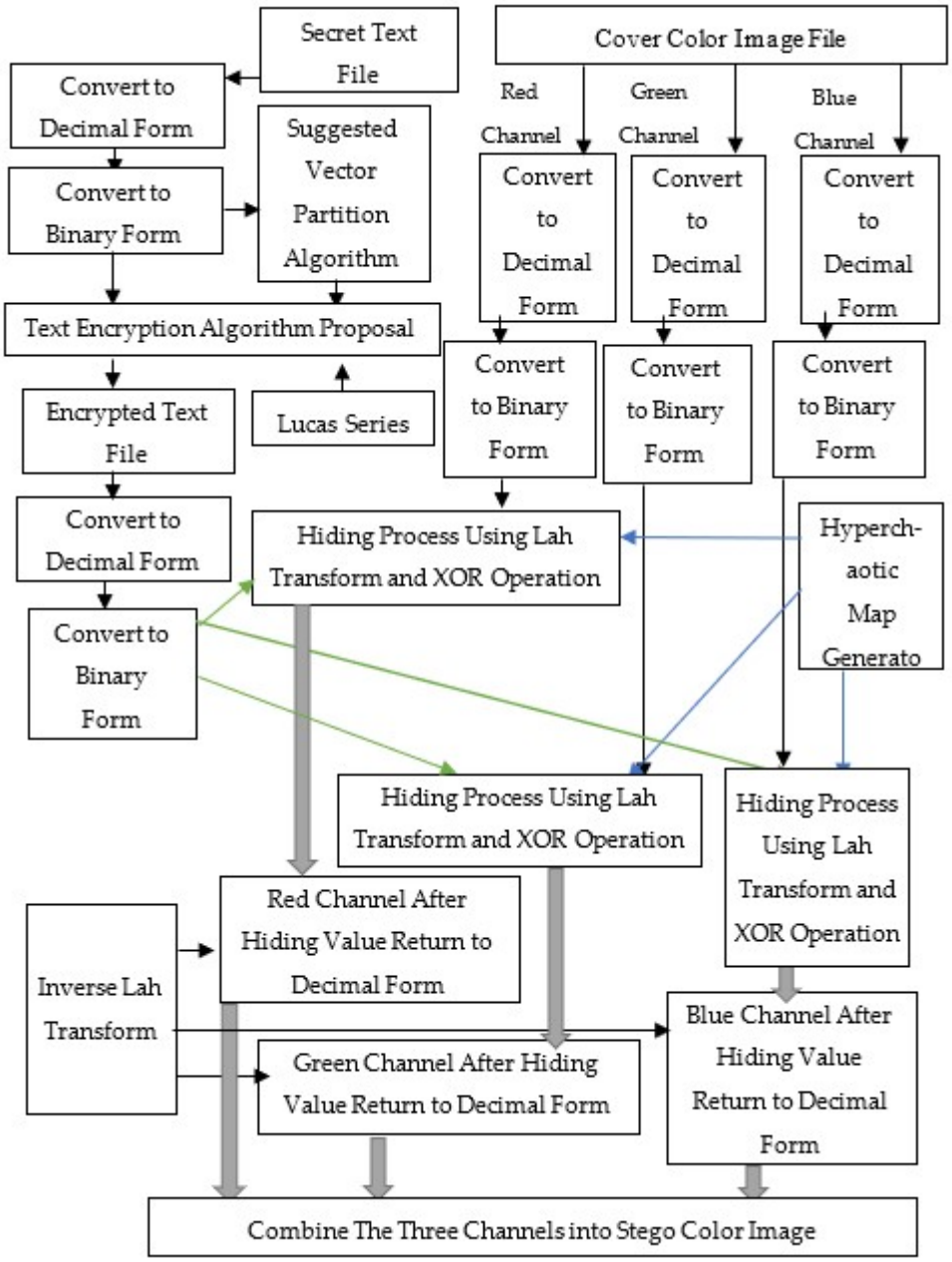


Figure 1. Proposed encrypted and hiding algorithms schema diagram.

4.1. Suggested Vector Partition Algorithm

Begin
Input: Input text file.
Output: Vector Elem, which contains the suggested lengths of the original vector V.

1. Convert the input text file to a decimal representation before converting it to binary form, then store it as vector V.
2. Determine the length of the input vector V; let this be L.

3. Set the lowest length permitted to Min_L = 9 and the maximum length allowed to Max_L = 32 (values determined experimentally, as described in Remark 5).
4. Perform the following steps for each value of vector V:
   a. Calculate the value of c, which is equal to mod (L, Max_L).
   b. If the value of c is greater than nine or equal to zero, use the equation $Max_L = Max_L + 1$. Otherwise, store the initial split length in the vector labelled "Elem," as Elem(i) = c, then remove the amount of the cut-off c from the whole length L such that $L = L - c$.
   c. Amend the maximum allowed length according to the following equation: Max_L = round ((Max_L/3) ×2).
   d. Return to step a.
5. Advance to the next stage with the result vector "Elem", which will contain the suggested lengths of the original vector V.

End

Remark 5. *To divide the binary result text vector of the entered text into the most appropriate number of subvectors, it is necessary to select the shortest possible form for the text file's contents, which is one symbol (i.e., eight bits). It was thus assumed that the highest allowable length would be four symbols (i.e., 32 bits) to ensure the best possible bit shifting and execution speed.*

4.2. Text Encryption Algorithm Proposal

Begin
Input: Input text file, Vector Elem.
Output: Encrypted text file.

1. Read the text from the .txt file offered as input.
2. Convert the read value from the ASCII code to the binary representation for each symbol in the read file.
3. Combine the values in their binary format to form a single vector (V).
4. Call the Suggested vector partition algorithm and save the results in the Elem vector.
5. To shuffle the values for each segment length contained in Elem, perform the following steps:
   a. Determine the length L_i from the vector Elem, and then read the values from the original vector with length L_i.
   b. Change the placements of the values in the vector at L_i. Put the elements in the even places first, followed by the items in the odd positions.
   c. The main transition is performed by utilising the values produced in the Lucas series, which offers position indication for the values in the vector. Every two sequentially created values are thus swapped.
   d. Slip to the right by two locations to rotate the generated values in the final vector.
6. Convert all eight bits from binary to decimal representation, and then convert these to ASCII code.
7. Place the ciphertext generated by the encryption procedure in a new.txt file.

End

4.3. Suggested Hiding Algorithm

Begin
Input: Encrypted text file, length of encrypted text file, cover colour image.
Output: Stego colour image.

1. Read the encrypted text file and transform it from ASCII format to decimal format, then save it in vector format.

2. Convert each decimal value in the vector element to a binary representation and save each binary data point (0 or 1) in a unique location within the encrypted text file's value vector.
3. Read the pixels in the cover colour image, and split them into red, green, and blue channels.
4. Create a series of random values with a length equal to the number of values included in each channel of the colour image by using a hyperchaotic map with a range of 1 to 255.
5. Arrange the produced random values in ascending order and save each value's index in a new result vector for later use.
6. For each of the three channels (red, green, and blue), complete the following steps (Figure 2 illustrates the modified LSB from step 6a to 6j):
    a. Choose two values from the bit vector to hide (hb1, hb2).
    b. Convert these values from a binary matrix to a one-dimensional vector.
    c. Select four values from the full vector of values, depending on the ascending vector of the indexes, and then apply the Lah transform.
    d. Convert each new value of the four output values of the Lah transformation from decimal to binary representation.
    e. Determine the number of zeros (Z1) and ones (Z2) in the binary representation, excluding the last two bits.
    f. If (Z1) is greater than (Z2), apply the following equations: X1 = XOR(hb1,0), X2 = XOR(hb2,0)
    g. Place the first new value, X1, in position 1 of the binary representation of the value, and the second new value, X2, in position 2 of the binary representation.
    h. If (Z2) is greater than or equal to (Z1), apply the following equations: X1 = XOR(hb1,1), X2 = XOR(hb2,1)
    i. Hide the first new value, X1, in position 1 of the binary representation of the value, and the second new value, X2, in position 2 of the binary representation.
    j. Return the updated value after concealment to a decimal representation
    k. To access these values, perform an inverse Lah transformation and store these to the same positions inside the overall image value pixel.
    l. Convert the resultant vector into a matrix.
7. Combine the three resulting matrices into a single image that reflects the final image post steganography technique.

End

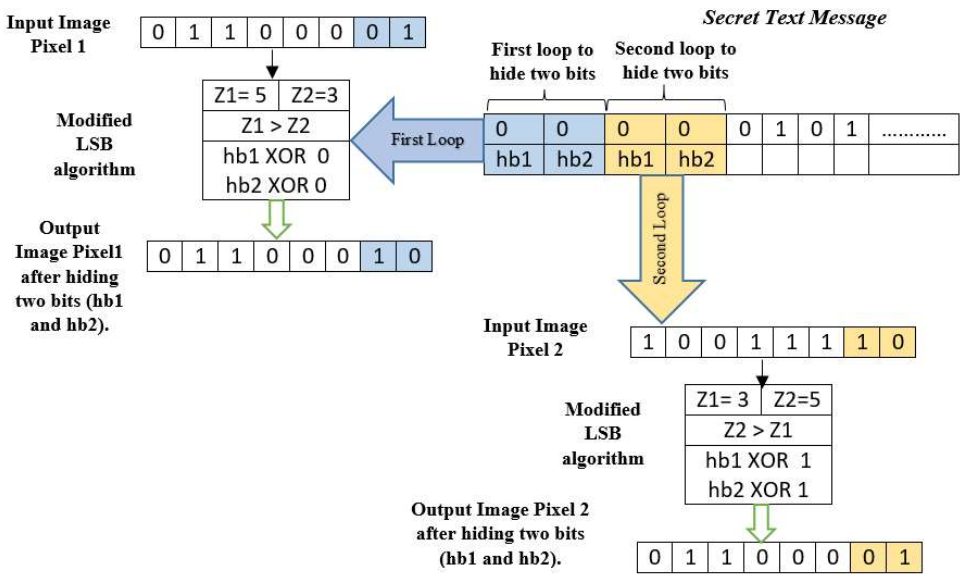


Figure 2. Modified LSB algorithm proposed to hide an encrypted text file.

4.4. Recovery Procedure

To extract the hidden data from the stego-image, the proposed method should be performed as explained in this section.

Begin

Input: Stego colour image, series of random values using a hyperchaotic map (length = Red channel × 3 values), length of an encrypted text file.

Output: Encrypted text file, cover colour image.

1. Read the pixels in the stego colour image, and split them into red, green, and blue channels.
2. Arrange the resulting series of random values in ascending order and save each value's index in a new result vector for later use.
3. For each of the three channels (red, green, and blue), complete the following:
    a. Convert values from a binary matrix to a one-dimensional vector.
    b. Select four values from the full vector of values based on the ascending vector of the indexes, and then apply the Lah transform.
    c. Convert each new value of the four output values of the Lah transformation from decimal to binary representation.
    d. Determine the number of zeros (Z1) and ones (Z2) in the binary representation, except for the last two bits.
        1- If (Z1) is greater than (Z2), perform the following steps: read the first value X1 in position 1 of the binary representation of the value and the second value X2 in position 2 of the binary representation to restore the hidden text bits (hb1, hb2) in the form hb1 = XOR(X1,0), hb2 = XOR(X2,0).
        2- If (Z2) is larger than or equal to (Z1), then restore the first value X1 in position 1 of the binary representation of the value and the second value X2 in position 2 of the binary representation (hb1, hb2) as follows: hb1 = XOR(X1,0), hb2 = XOR(X2,0).
        3- Arrange all restored binary values into the result vector until the text file's requested length is reached.
        4- Return the updated value after restoring it to a decimal representation.
        5- To get the actual values, perform the inverse Lah transformation and store the results to the same positions inside the overall image value pixel.
        6- Convert the resultant vector to a matrix.
    e. Transform the restored result vector from binary format to decimal format, then to ASCII format.
    f. Save the text vector in ASCII codes in the file as encrypted text.
    g. Combine the three resulting matrices into a single image that reflects the final image.

End

4.5. Suggested Decryption Mechanism

Begin

Input: Encrypted text file.

Output: Decrypted text file, vector Elem.

1. Read the text from the .txt file input.
2. Convert the read value from the ASCII code to a decimal representation and then to a binary representation for each symbol in the read file.
3. Combine the values in their binary format to form a single vector (V).
4. Call the suggested vector partition algorithm and save the results in the Elem vector.
5. To shuffle the values for each segment length contained in the Elem, perform the following steps:
    a. Calculate Li (the length) from the vector Elem, and then extract the values from the original vector of length Li.

b. To rotate the produced values in the final vector to their original locations, shift left two locations.
c. The transition is carried out by employing the values generated by the Lucas series, which indicate the locations of the values in the vector. Every two values produced are thus consecutively swapped.
d. Modify the values' places in the vector at Li, placing the elements in the odd places first, followed by the items in the even places.

6. Convert all eight bits from binary to decimal representation, and then to ASCII code.
7. Place the deciphered text generated after the decryption procedure in a new .txt file.

End

5. Experimental Results

The experimental goal was to compare the performance of six cover images in terms of hiding data, as shown in Figure 3. The dataset consisted of a text file (.txt) used to carry a secret message. The sizes of the cover image file used were 256 × 256 and 512 × 512. The experiment was carried out on a computer equipped with an Intel(R) Core(TM) i7-9750H CPU running at 2.60GHz, with 16.0 GB RAM under a 64-bit Windows 10 operating system.

Figure 3. Cover image samples used in experimentation with sizes 512 × 512 and 256 × 256.

Many common measures were then applied to determine the quality of the image and to evaluate the performance of the proposed image steganography method.

5.1. MSE and PSNR Measures

MSE is an estimate of the mean value of the squares of the errors between the stego-image and the original image [41–44]:

$$MSE = \frac{1}{XY} \sum_{a1=1}^{X} \sum_{a2=1}^{Y} (W_{a1,a2} - W'_{a1,a2}) \quad (9)$$

where $W_{a1,a2}$ is the pixel in the cover image in the $a1_{th}$ row and $a2_{th}$ column, $W'_{a1,a2}$ is the pixel in the stego-image in the $a1_{th}$ row and $a2_{th}$ column, and XY is the size of the image, where X is the height and Y is the width.

PSNR is often used as a quality measurement to determine degradation in the embedded image with respect to the cover image, thus highlighting the difference between the original and the stego-image [45–48]:

$$PSNR = 10\, log_{10} \frac{U}{MSE} \quad (10)$$

where U is the pixel range value; so, for an 8-bit red, green, and blue colour, we use the average of these three values, giving U = 255.

5.2. Structural Similarity Index (SSIM)

SSIM is designed to determine the similarity between two images, which in this case are the original and corresponding stego-images. Values close to 1 are indicators of the best possible structural similarity between the compared images. SSIM is defined as [49,50]:

$$SSIM(X,Y) = L(X,Y).M(X,Y).N(X,Y) \tag{11}$$

$$L(X,Y) = \frac{2\alpha_X\alpha_Y + C_1}{\alpha_X^2\alpha_Y^2 + C_1} \tag{12}$$

$$M(X,Y) = \frac{2\gamma_X\gamma_Y + C_2}{\gamma_X^2\gamma_Y^2 + C_2} \tag{13}$$

$$N(X,Y) = \frac{\gamma_{XY} + C_3}{\gamma_X\gamma_Y + C_3} \tag{14}$$

where $L(X,Y)$ represents the luminance function, used for measuring the proximity between mean luminance $\alpha_X\alpha_Y$ of cover and stego-images; $M(X,Y)$ represents the contrast function, used for measuring the proximity between the contrast of the two images; γ_X and γ_Y are used to measure the contrast; $N(X,Y)$ represents the structural function for measuring the correlation coefficient between cover and stego-images; γ_{XY} represents the covariance between the images X and Y; and C_1, C_2, C_3 are positive constants. C_1 equals $(V_1L)^2$, and C_2 equals $(V_2L)^2$. They are two independent variables to balance the division with a low denominator; L equals the dynamic range of the image pixels; by default, $V_1 = 0.01$ and $V_2 = 0.03$; $C_3 = C_2/2$.

5.3. Normalised Cross Correlation (NCC)

NCC is one of the more effective statistical metrics for determining the closeness between two images. Despite the effects of embedding data, the NCC can thus be used to estimate how close the cover and stego-images are. The NCC is computed as in Equation (15) [6]:

$$NCC = \sum_{a1=1}^{X}\sum_{a2=1}^{Y}\left(\frac{W_{a1,a2} \times W'_{a1,a2}}{W_{a1,a2}}\right) \tag{15}$$

where X and Y specify the cover image and stego-image sizes. *NCC* scores range from -1 to 1. A value of NCC approaching 1 shows that the original and stego-images are very close [6]. The distance of NCC from unity represents the tiny differences between the cover and the stego-image pixels.

5.4. Root Mean Square Error (RMSE)

The damage to the stego-image is measured using RMSE, which is calculated using Equation (16) [35]:

$$RMSE = \sqrt{\frac{1}{XY}\sum_{a1=1}^{X}\sum_{a2=1}^{Y}(W_{a1,a2} - W'_{a1,a2})^2} \tag{16}$$

We can determine from Equations (9) and (16) that the value of *RMSE* equals the square root of the value of *MSE*.

5.5. Correlation Coefficient (CC)

In the original image, all pixels are substantially linearly connected and, thus, one of the goals of an excellent data steganography system is to maximise the correlation between adjacent pixels in the encrypted image. The values of the stego-image correlation coefficient and cover image are almost the same, approaching unity. Table 1 shows the results of the experiment based on the correlation values [6].

5.6. Bias

When estimating bias, small values indicate maximum similarity, whereas large values indicate little similarity between images. For this purpose, bias is determined using Equation (17) [35].

$$Bias = 1 - \frac{W'}{W} \tag{17}$$

The value of bias continuously reduces with the size of the cover image and the stability of the quantity of concealed text messages, as shown in Table 1.

5.7. Difference in Variance (DIV)

Equation (18) was used to calculate the *DIV*. A small value of *DIV* indicates significant closeness between images, whereas a large *DIV* indicates greater differences [35].

$$DIV = 1 - \frac{\partial^2 W'}{\partial^2 W} \tag{18}$$

Here, $\partial^2 W'$ and $\partial^2 W$ represent the standard deviations the of cover and stego-images, respectively.

5.8. Structural Contents (SC)

The SC compares the similarity of the cover and stego-images by examining the relevant region between them. Equation (19) is used to calculate the *SC* [35].

$$SC = \frac{\sum_{a1=1}^{X} \sum_{a2=1}^{Y} \left(W'_{a1,a2}\right)^2}{\sum_{a1=1}^{A} \sum_{a2=1}^{B} \left(W_{a1,a2}\right)^2} \tag{19}$$

The standard colour channels are red, green, and blue (in an image). The *SC* value is close to 1 when there is little change.

5.9. Entropy Differential

The entropy differential between two images is calculated as in Equation (20) [51]:

$$Entropy\ Diff = -\sum_{x \in X} \sum_{y \in Y} \Pr(X, Y) ln(\Pr(X, Y)) \tag{20}$$

where Pr represents the probability, X represents the cover image, and Y represents the stego-image.

5.10. Local Quality Index (LQI)

The local quality index of an image, as determined between stego and cover image, is computed as follows [35]:

$$LQI = \frac{4\partial_{WW'} WW'}{\left(\partial_W^2 + \partial_{W'}^2\right)\left(W^2 + W'^2\right)} \tag{21}$$

when attempting to hide a similar scrambled text message, the IQI value approaches one as the size of the cover image increases, as shown in Table 1.

5.11. Average Difference (AD)

To ensure good concealment of the stego-image in the cover image, the AD measure must be zero as determined in Equation (22) [35].

$$AD = \frac{\sum_{a1=1}^{X} \sum_{a2=1}^{Y} \left(W_{a1,a2} - W'_{a1,a2}\right)}{XY} \tag{22}$$

Table 1. Performance of PSNR, MSE, SSIM, NCC, and embedding capacity in Lina images of different sizes.

Static Meters	Between cover Image and Stego Image (256 × 256 Pixels)			Between cover Image and Stego Image (512 × 512 Pixels)			Embedded (Byte)	Percentage Ratio used from Image File (256 × 256 Pixels) for Hiding Information	Percentage Ratio used from Image File (512 × 512 Pixels) for Hiding Information
	Red	Green	Blue	Red	Green	Blue			
PSNR	44.9771	INF	INF	51.0322	INF	INF	2000	10.200%	32.227%
	37.9191	INF	INF	43.9355	INF	INF	10,000	2.040%	6.445%
	35.7730	49.7083	INF	41.6130	INF	INF	17,000	1.200%	3.791%
MSE	2.0671	0	0	0.5127	0	0	2000	10.200%	32.227%
	10.4995	0	0	2.6274	0	0	10,000	2.040%	6.445%
	17.2100	0.6954	0	4.4852	0	0	17,000	1.200%	3.791%
RMSE	1.2242	0	0	0.4356	0	0	2000	10.200%	32.227%
	3.1435	0	0	1.4198	0	0	10,000	2.040%	6.445%
	4.0733	0.5076	0	1.9580	0	0	17,000	1.200%	3.791%
NCC	1	1	1	1	1	1	2000	10.200%	32.227%
	1	1	1	1	1	1	10,000	2.040%	6.445%
	1	1	1	1	1	1	17,000	1.200%	3.791%
SSIM	0.9875	1	1	0.9991	1	1	2000	10.200%	32.227%
	0.9394	1	1	0.9956	1	1	10,000	2.040%	6.445%
	0.9069	0.9959	1	0.9926	1	1	17,000	1.200%	3.791%
Bias	0.0008	0	0	0.0003	0	0	2000	10.200%	32.227%
	0.0019	0	0	0.0009	0	0	10,000	2.040%	6.445%
	0.0026	0.0008	0	0.0013	0	0	17,000	1.200%	3.791%
DIV	0.3650	0	0	0.2209	0	0	2000	10.200%	32.227%
	0.7981	0	0	0.5798	0	0	10,000	2.040%	6.445%
	1.0512	0.1316	0	0.7261	0	0	17,000	1.200%	3.791%
CC	0.9996	1	1	0.9999	1	1	2000	10.200%	32.227%
	0.9978	1	1	0.9995	1	1	10,000	2.040%	6.445%
	0.9965	0.9999	1	0.9991	1	1	17,000	1.200%	3.791%
SC	1	1	1	1	1	1	2000	10.200%	32.227%
	1	1	1	1	1	1	10,000	2.040%	6.445%
	1	0.9999	1	1	1	1	17,000	1.200%	3.791%
Entropy Diff	0.0547	0	0	0.0248	0	0	2000	10.200%	32.227%
	0.1935	0	0	0.0939	0	0	10,000	2.040%	6.445%
	0.2755	0.0209	0	0.1462	0	0	17,000	1.200%	3.791%
Local Quality Index	0.9564	1	1	0.9740	1	1	2000	10.200%	32.227%
	0.8484	1	1	0.8919	1	1	10,000	2.040%	6.445%
	0.8006	0.9856	1	0.8434	1	1	17,000	1.200%	3.791%
AD	0.0041	0	0	−0.0009	0	0	2000	10.200%	32.227%
	0.0322	0	0	0.0057	0	0	10,000	2.040%	6.445%
	0.0570	−0.0034	0	0.0079	0	0	17,000	1.200%	3.791%
MD	25	0	0	26	0	0	2000	10.200%	32.227%
	26	0	0	26	0	0	10,000	2.040%	6.445%
	26	26	0	26	0	0	17,000	1.200%	3.791%

5.12. Maximum Difference (MD)

MD is calculated as shown in Equation (23). To achieve a good data concealing strategy, the MD, as assessed between the cover and the stego-image must be minimal [35].

$$MD = MAX(|W_{a1,a2} - W'_{a1,a2}|) \quad (23)$$

Table 1 displays the performance results for the suggested technique with respect to hiding text in a colour image. It shows the measured results for all the above metrics and uses embedding capacity to evaluate the image quality before and after the text is included.

The graphs shown in Figures 4–9 are PSNR, MSE, SSIM, DIV, bias, and entropy differential measures, respectively, applied to different images.

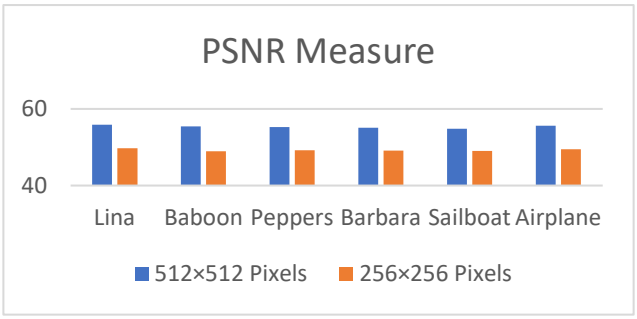


Figure 4. PSNR measure for embedding 2000 bytes in colour images of sizes 512 × 512 and 256 × 256.

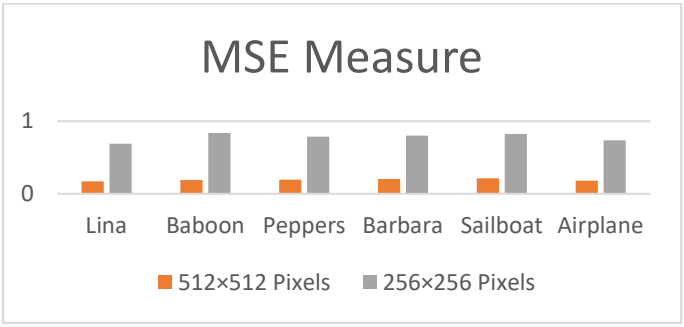


Figure 5. MSE measure for embedding 2000 bytes in colour images of sizes 512 × 512 and 256 × 256.

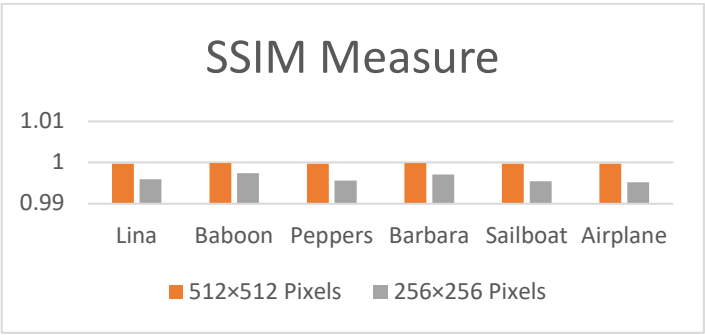


Figure 6. SSIM measure for embedding 2000 bytes in colour images of sizes 512 × 512 and 256 × 256.

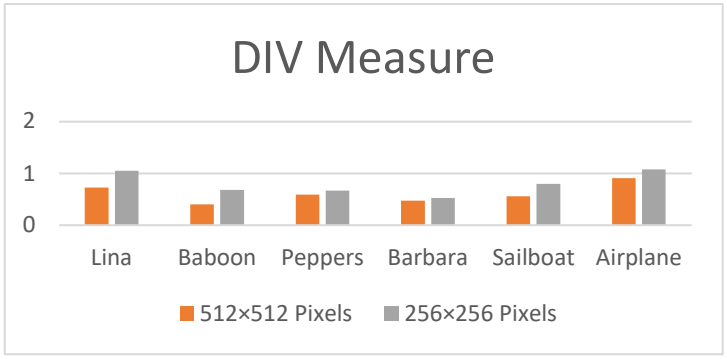


Figure 7. DIV measure for embedding 2000 bytes in colour images of sizes 512 × 512 and 256 × 256.

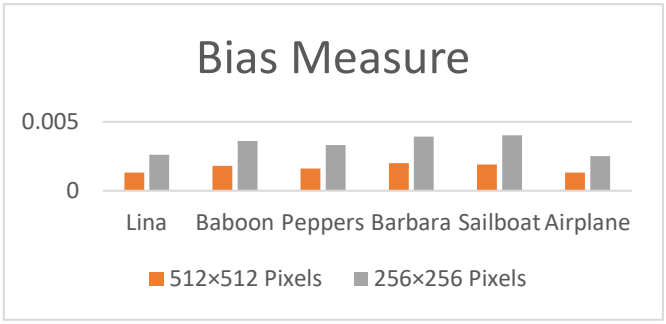


Figure 8. Bias measure for embedding 2000 bytes in colour images of sizes 512 × 512 and 256 × 256.

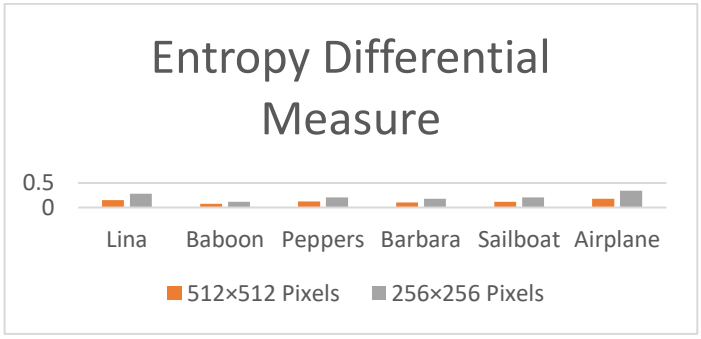


Figure 9. Entropy differential measure for embedding 2000 bytes in colour images of sizes 512 × 512 and 256 × 256.

The length of the text message is the main determinant of performance outcome. According to these findings, the lower the MSE, RMSE, and bias values and the higher the PSNR and SSIM values, the better the stego-image output quality. The NCC, AD, SC, and SSIM values were all close to +1, indicating a strong link between the original and the stego-images.

5.13. Histogram

An image histogram is one of the most critical criteria in statistical image steganography. It displays the pixel density for each brightness value, allowing evaluation of full tonal distribution at a glance based on the histogram for a given image. The histograms for each cover image and stego-image were thus calculated to evaluate the proposed technique using histogram analysis, as shown in Figure 10. Figure 10 illustrates that the histograms of each image cover and its stego-image differ only slightly. As a result, the distortion of the pixel values in the stego-image is reduced, creating no noticeable impacts on the histogram or the visual quality of the resulting colour images.

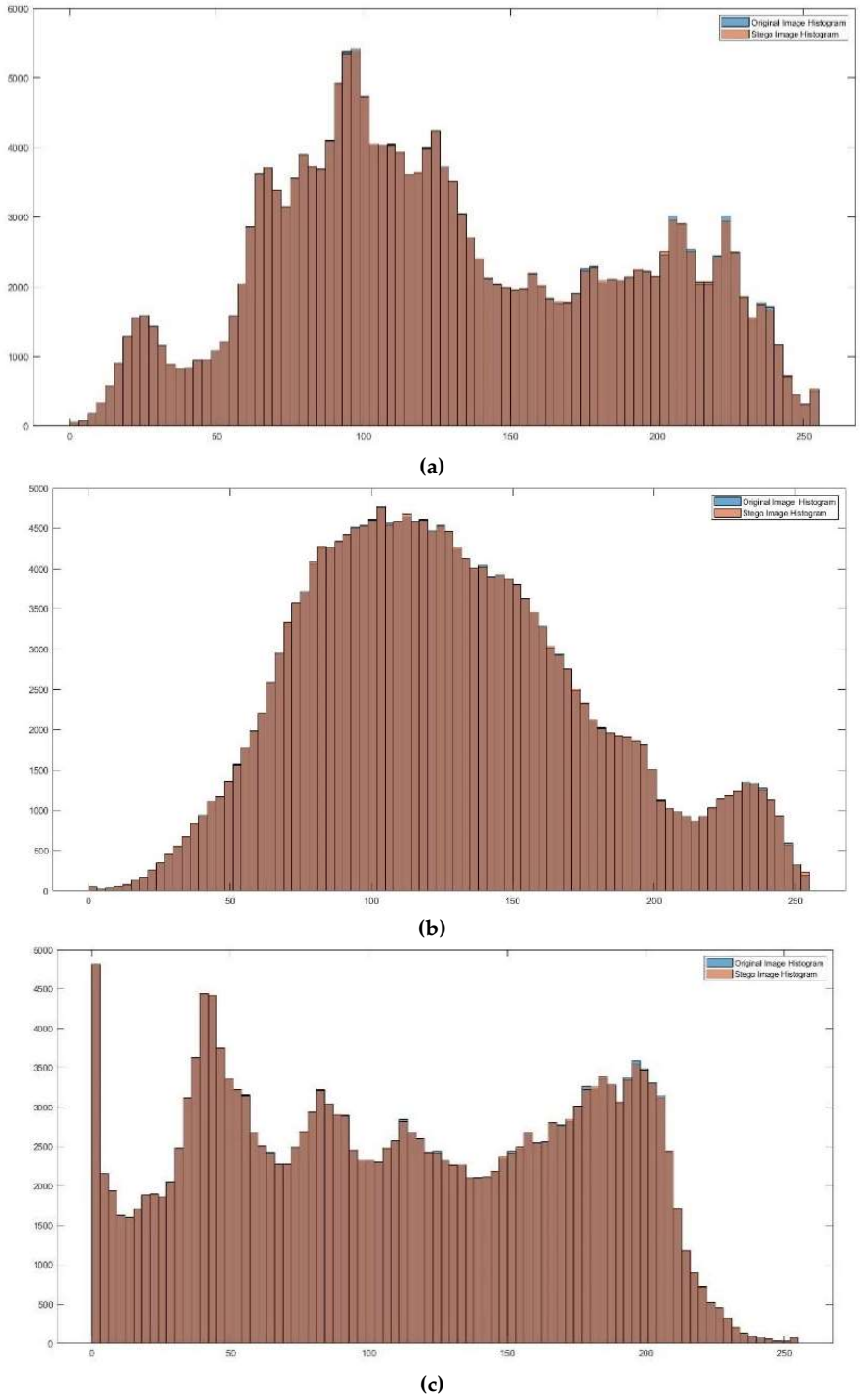
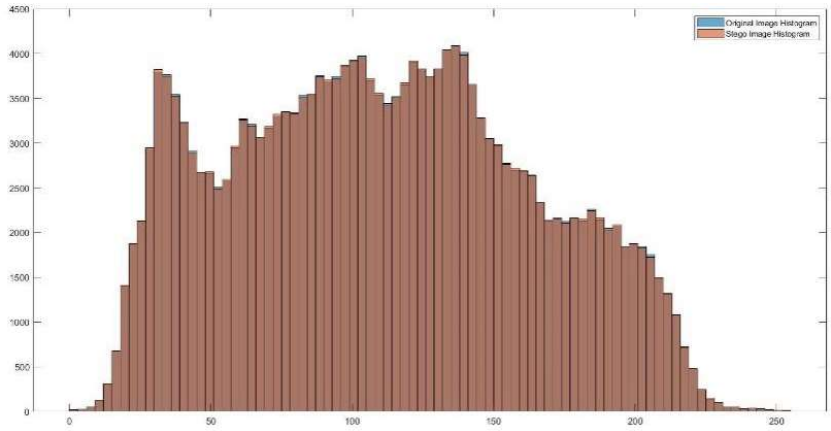
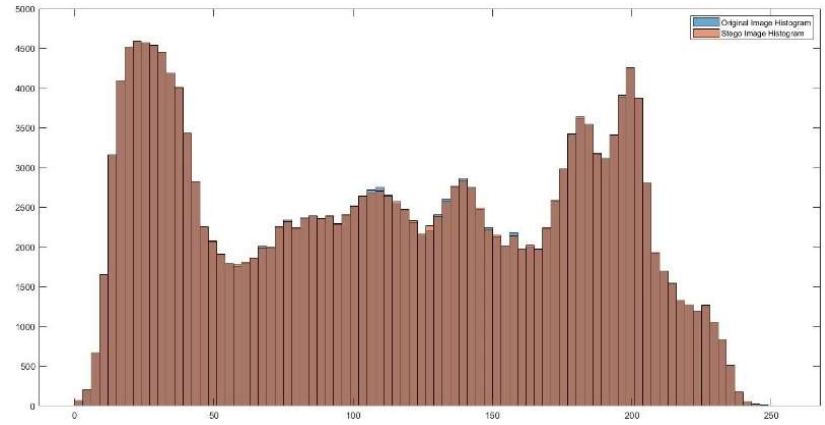


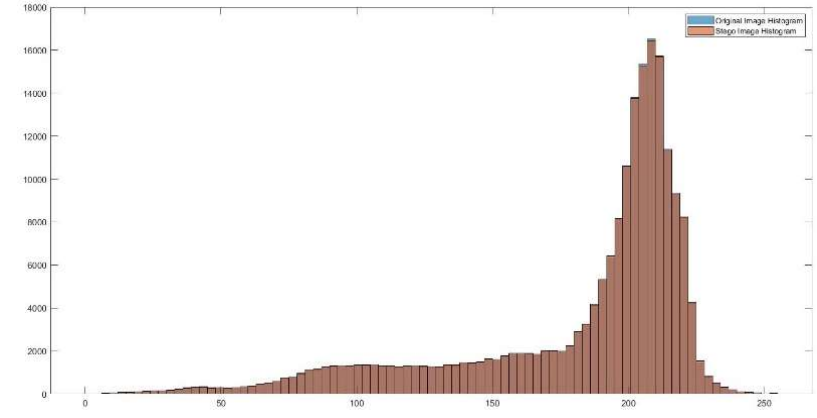
Figure 10. *Cont.*



(d)



(e)



(f)

Figure 10. Original image with stego image histograms. Histograms of original and (**a**) stego Lina image; (**b**) stego baboon image; (**c**) stego pepper image; (**d**) stego Barbara image; (**e**) stego sailboat image; (**f**) stego airplane image.

Text files were tested as a type of secret message contained in the cover colour image. The parameters for measuring the payload were thus the size of the encrypted text file and whether or not this was successfully included inside the cover image. When the proposed methodology was compared with the method in [35], it was seen to significantly reduce MSE while, as shown in Table 2, keeping the PSNR consistently over 40 dB [52], implying that any degradation in quality would be hard to detect with the human eye. The same is true for the bias and SSIM analyses, which bears mentioning in consideration of the concerns raised in [35,53]. As the bias value is low, the SSIM value is also close to 1. It is also important to note that this embedding capacity is based on hiding two bits in each

byte of the cover image, which means that using three colour image channels offers a high storage capacity.

Table 2. Comparison of the proposed algorithm with [35,39] with respect to embedding 2,000 Bytes.

Image Name	Algorithm	Embedded (Bytes)	Image Size in Pixel	PSNR	MSE	Bias	SSIM
Lina	Proposed Algorithm	100	256 × 256	58.7705	0.0863	0.00006	0.9994
Lina	Proposed Algorithm	100	512 × 512	64.4197	0.0235	0.00001	1
Lina	[39]	100	128 × 128	50.95	0.52	-	0.9990
Lina	[35]	Not Determined	256 × 256	-	-	0.0004	0.9651

5.14. Information Entropy

Information entropy metrics consider Shannon entropy, joint entropy (JE), mutual entropy, relative entropy, and conditional entropy. All of these types of entropies may be calculated as shown in Table 3 [35]:

Table 3. Formulae for all entropy types.

Entropy Type	Measurement Equation	Notes
Shannon Entropy	$E = \sum_{x \in X} \Pr(X) \log[\Pr(X)]$	Where Pr represent probability, X represent cover image, Y represent stego image.
Joint Entropy	$JE(X,Y) = -\sum_{x \in X} \sum_{y \in Y} \Pr(x,y) \log[\Pr(x,y)]$	
Mutual Entropy	$ME(X,Y) = E(X,Y) - E(X\|Y)$	
Relative Entropy	$RE(X,Y) = \sum \Pr(X) \log \frac{\Pr(X)}{\Pr(Y)}$	
Conditional Entropy	$CE(X,Y) = -\sum_{x \in X} \sum_{y \in Y} \Pr(X)\Pr(y\|X) \log[\Pr(y\|X)]$	

As shown in Table 4, the computed values are present. There was a close match between the Shannon entropy values of the cover image and the stego-image. Additionally, mutual entropy and conditional entropy grow as the amount of hidden data increases. Moreover, relative entropy is close to zero.

Table 4. All entropy types resulting from comparison between cover and stego images.

Entropy Types	Proposed Algorithm WhenImage Size (256 × 256 Pixels)		Proposed Algorithm When Image Size (512 × 512 Pixels)		Embedded (byte)
Shannon Entropy	7.4467	7.4470	7.4440	7.4442	2000
	7.4467	7.4494	7.4440	7.4448	10,000
	7.4467	7.4526	7.4440	7.4450	17,000
Joint Entropy	7.7963		22.2293		2000
	8.7536		23.6106		10,000
	9.3451		24.5933		17,000
Mutual Entropy	7.0977		21.3240		2000
	6.1453		19.9479		10,000
	5.5600		18.9670		17,000
Relative Entropy	0.0006		0.0003		2000
	0.0009		0.0007		10,000
	0.0037		0.0021		17,000
Conditional Entropy	0.3496		−13.8790		2000
	1.3069		−12.4977		10,000
	1.8984		−11.5150		17,000

As shown in Table 5, based on the results given in [35], the suggested strategy demonstrates better capability than other recent methods in increasing entropy.

Table 5. Comparison of entropy types between the proposed algorithm and [35] for embedding 17,000 Bytes.

Entropy Types	Proposed Algorithm		[35]	
Shannon Entropy	7.4467	7.4526	7.7330	7.7348
Joint Entropy	9.3451		9.4260	
Mutual Entropy	5.5600		6.0147	
Relative Entropy	0.0037		0.0009	
Conditional Entropy	1.8984		1.6912	

5.15. Implementation Time

The implementation time for the proposed steganography method when ciphertexts of various sizes are hidden, as well as the time required to reveal them, must be measured to determine the effectiveness and usability of the proposed technique for steganography applications. Based on the duration of concealment and recovery for two different sized colour images (256 × 256 and 512 × 512 pixels), Table 6 calculates the times required for embedding various sizes of ciphertext in the cover image.

Table 6. A comparison of concealment and recovery times for different colour image sizes.

Time (Second)	Image Size (256 × 256 Pixels)	Image Size (512 × 512 Pixels)	Embedded (byte)
Hiding Time	3.7092	9. 9262	2000
	3.8942	9.8410	10,000
	3.8141	9.9572	17,000
Recovery Time	3.9100	10.0100	2000
	3.9921	10.4015	10,000
	3.9704	10.7510	17,000

Due to the usage of two low-complexity chaotic maps and the resulting dependence on hiding two bits in each byte of the cover image utilising LSB technology, the suggested approach requires little time; further, any increase in the size of the concealed data was minimal.

Remark 6. *The paper aims to provide excellent security with minimum execution time utilising various lightweight strategies. Two strategies are used in the hiding stage to allow its speed to be acceptable: the hyperchaotic map relies on two low-complexity varieties of chaos, and the LSB relies on concealing two bits at a time. As a result, the proposed technique presented in this study stands out for its outstanding simplicity, short execution time, and great secrecy.*

5.16. Analysis of Data Steganography Method by Distance

Several measurements were performed based on distance evaluations between the stego and cover images to confirm their differences, where shorter distances reflect significant connections between the stego and cover images. Table 7 presents the results of several distance analysis types.

Table 7. Analysis of stego and cover images by distance using different colour image Lina sizes.

Distance Metric	Image Size (256 × 256 Pixels)	Image Size (512 × 512 Pixels)	Embedded (byte)
Euclidean Distance	0.0123	0.0126	2000
	0.0051	0.0055	10,000
	0.0040	0.0042	17,000
Mean Squared Euclidean Distance	0	0	2000
	0	0	10,000
	0	0	17,000
Squared Euclidean Distance	0.00015251	0.00015838	2000
	0.00002651	0.00002983	10,000
	0.00001609	0.00001765	17,000
Mean Euclidean Distance	0	0	2000
	0	0	10,000
	0	0	17,000
Manhattan Distance	878	858	2000
	2010	1950	10,000
	2628	2512	17,000
Cosine Distance	0.000117	0.00000706	2000
	0.000686	0.00003798	10,000
	0.0011	0.00006436	17,000
Correlation Distance	0.00039434	0.00002105	2000
	0.0019	0.00010865	10,000
	0.0030	0.00018696	17,000
Mean Pattern Intensity	0.000072	0.00000092	2000
	0.000317	0.00000803	10,000
	0.009320	0.00007749	17,000
Earth Mover Distance	0.0112	0.0044	2000
	0.0432	0.0155	10,000
	0.0519	0.0158	17,000

6. Discussion

The results of various metrics such as the MSE, NCC, AD, NAE, and SC values obtained for the Lina image in two sizes (256 × 256 and 512 × 512 pixels) with varying sizes of hidden texts (2000, 10,000, and 17,000 bytes) are illustrated in Table 1. According to an analysis of the results, the MSE rate is reduced by the suggested technique, and the PSNR rate is higher. That means that the current approach creates no mistakes in the embedding procedure. Table 2 compares the proposed approach to work in [35] and [39] to determine the value of the suggested system, whereas Figures 4–9 provide comparative plots for measures such as PSNR, MSE, SSIM, DIV, bias, and entropy difference. As in Tables 4 and 5, all entropy categories for the Lina image show that the new technique outperforms the present methods. Table 5 also shows that the Shannon entropy difference value between the cover and stego-images is close to zero, which means no change to the cover image is required to disclose the reality of the hidden data. Furthermore, examining the joint entropy with different hiding ratios for the proposed technique (as given in Table 4) shows that joint entropy significantly changes with different embedded ratios and, as a result, joint entropy is a distinguishing feature of this steganographic technique. This applies to the analysis of mutual and conditional entropies, whereas for relative entropy, the value decreases as the size of the cover image increases while the size of the hidden data remains the same; thus, the closer to zero, the better.

7. Conclusions

For covert communication between two authorised organizations, robust information security is necessary. Based on this condition, this work proposed the use of the Lah

transformation. The concealed location is detected using a modified LSB algorithm that includes criteria such as adding randomness to the hidden data once it has reached the final bit, and sometimes even in the preceding bit. The pixels in the image are also randomly selected based on a hyperchaotic map created from two types of classic chaos keys (the Tent map and Ikeda map) to increase the difficulty in predicting the locations in which the data were hidden.

To maintain data privacy and to increase the difficulty of access by attackers, the procedures for hiding textual data were preceded by cryptography. The combination of the Lah transform, hyperchaotic map, and modified LSB offer an effective and lightweight strategy, as seen by the measurements from the value meters in Tables 1, 4, 6 and 7. When used in conjunction with encryption, the recommended approach thus ensures significant privacy of text content during internet conversations.

Author Contributions: I.Q.A. and Z.A.A.; methodology, and writing—original draft preparation, M.A.A.S.; software, and data curation, M.J.J.G.; validation, and writing—review and editing, J.M. and V.O.N.; formal analysis, investigation, supervision, project administration, and funding acquisition, J.M; resources, and visualization. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by university-enterprise cooperative R&D project of SZTU (grant no. 20221061030001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: All individuals included in this section have consented to the acknowledgement.

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

References

1. Ambika; Biradar, R.L.; Burkpalli, V. Encryption-Based Steganography of Images by Multiobjective Whale Optimal Pixel Selection. *Int. J. Comput. Appl.* **2019**, 1–10. [CrossRef]
2. Al Sibahee, M.A.; Lu, S.; Abduljabbar, Z.A.; Liu, X.; Abdalla, H.B.; Hussain, M.A.; Hussien, Z.A.; Jassim Ghrabat, M.J. Lightweight Secure Message Delivery for E2E S2S Communication in the IoT-Cloud System. *IEEE Access* **2020**, *8*, 218331–218347. [CrossRef]
3. Gulve, A.K.; Joshi, M.S. An Image Steganography Method Hiding Secret Data into Coefficients of Integer Wavelet Transform Using Pixel Value Differencing Approach. *Math. Probl. Eng.* **2015**, *2015*, 684824. [CrossRef]
4. Fateh, M.; Rezvani, M.; Irani, Y. A New Method of Coding for Steganography Based on LSB Matching Revisited. *Secur. Commun. Netw.* **2021**, *2021*, 6610678. [CrossRef]
5. Nagaraj, V.; Vijayalakshmi, V.; Zayaraz, G. Color Image Steganography Based on Pixel Value Modification Method Using Modulus Function. *IERI Procedia* **2013**, *4*, 17–24. [CrossRef]
6. Sahu, A.K.; Sahu, M. Digital Image Steganography and Steganalysis: A Journey of the Past Three Decades. *Open Comput. Sci.* **2020**, *10*, 296–342. [CrossRef]
7. Abdullah, S.M.; Abduljaleel, I.Q. Speech Encryption Technique Using S-Box Based on Multi Chaotic Maps. *TEM J.* **2021**, *10*, 1429–1434. [CrossRef]
8. Islam, M.R.; Tanni, T.R.; Parvin, S.; Sultana, M.J.; Siddiqa, A. A Modified LSB Image Steganography Method Using Filtering Algorithm and Stream of Password. *Inf. Secur. J. A Glob. Perspect.* **2021**, *30*, 359–370. [CrossRef]
9. El-Khamy, S.E.; Korany, N.O.; Mohamed, A.G. A New Fuzzy-DNA Image Encryption and Steganography Technique. *IEEE Access* **2020**, *8*, 148935–148951. [CrossRef]
10. Priyadharshini, A.; Umamaheswari, R.; Jayapandian, N.; Priyananci, S. Securing Medical Images Using Encryption and LSB Steganography. In Proceedings of the 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 19 February 2021; pp. 1–5. [CrossRef]
11. Gedkhaw, E.; Soodtoetong, N.; Ketcham, M. The Performance of Cover Image Steganography for Hidden Information within Image File Using Least Significant Bit Algorithm. In Proceedings of the 2018 18th International Symposium on Communications and Information Technologies (ISCIT), Bangkok, Thailand, 26–29 September 2018; pp. 504–508. [CrossRef]
12. Cemkasapbaşi, M.; Elmasry, W. New LSB-Based Colour Image Steganography Method to Enhance the Efficiency in Payload Capacity, Security and Integrity Check. *Sādhanā* **2018**, *43*, 68. [CrossRef]
13. Kasapbasi, M.C. A New Chaotic Image Steganography Technique Based on Huffman Compression of Turkish Texts and Fractal Encryption with Post-Quantum Security. *IEEE Access* **2019**, *7*, 148495–148510. [CrossRef]

14. Darbani, A.; AlyanNezhadi, M.M.; Forghani, M. A New Steganography Method for Embedding Message in JPEG Images. In Proceedings of the 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, 28 February–1 March 2019; pp. 617–621. [CrossRef]
15. Joshi, K.; Gill, S.; Yadav, R. A New Method of Image Steganography Using 7th Bit of a Pixel as Indicator by Introducing the Successive Temporary Pixel in the Gray Scale Image. *J. Comput. Netw. Commun.* **2018**, *2018*, 9475142. [CrossRef]
16. Abduljabbar, Z.A.; Jin, H.; Hussien, Z.A.; Yassin, A.A.; Hussain, M.A.; Abbdal, S.H.; Zou, D. Robust Image Document Authentication Code with Autonomous Biometrie Key Generation, Selection, and Updating in Cloud Environment. In Proceedings of the 2015 11th International Conference on Information Assurance and Security (IAS), Marrakech, Morocco, 14–16 December 2015; pp. 61–66. [CrossRef]
17. Amakdouf, H.; Zouhri, A.; EL Mallahi, M.; Qjidaa, H. Color Image Analysis of Quaternion Discrete Radial Krawtchouk Moments. *Multimed. Tools Appl.* **2020**, *79*, 26571–26586. [CrossRef]
18. Amakdouf, H.; Zouhri, A.; El Mallahi, M.; Tahiri, A.; Chenouni, D.; Qjidaa, H. Artificial Intelligent Classification of Biomedical Color Image Using Quaternion Discrete Radial Tchebichef Moments. *Multimed. Tools Appl.* **2021**, *80*, 3173–3192. [CrossRef]
19. Saad, A.H.S.; Mohamed, M.S.; Hafez, E.H. Coverless Image Steganography Based on Optical Mark Recognition and Machine Learning. *IEEE Access* **2021**, *9*, 16522–16531. [CrossRef]
20. Hashim, M.M.; Rahim, M.S.M. Image Steganography Based on Odd/Even Pixels Distribution Scheme and Two Parameters Random Function. *J. Theor. Appl. Inf. Technol.* **2017**, *95*, 5977–5986.
21. Harahap, M.K.; Khairina, N. Dynamic Steganography Least Significant Bit with Stretch on Pixels Neighborhood. *JISEBI* **2020**, *6*, 151. [CrossRef]
22. Hussein, H.L.; Abbass, A.A.; Naji, S.A.; Al-augby, S.; Lafta, J.H. Hiding Text in Gray Image Using Mapping Technique. *J. Phys. Conf. Ser.* **2018**, *1003*, 012032. [CrossRef]
23. Almayyahi, A.A.; Sulaiman, R.; Qamar, F.; Essa, A. High-Security Image Steganography Technique Using XNOR Operation and Fibonacci Algorithm. *IJACSA* **2020**, *11*. [CrossRef]
24. Ratnasari, D.; Aji, A.S. Text to Color Image Steganography Using LSB Technique and XOR Operations. *Int. J. Appl. Bus. Inf. Syst.* **2019**, *3*, 59–65.
25. Madhu, D.; Vasuhi, S. Image Steganography: 2-Bit XOR Algorithm Used in YCbCr Color Model with Crypto-Algorithm. In Proceedings of the 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, India, 28 September 2020; pp. 1–5. [CrossRef]
26. Nassif Jassim, K.; Khudhur Nsaif, A.; Kuder Nseaf, A.; Hazidar, A.H.; Priambodo, B.; Naf'an, E.; Masril, M.; Handriani, I.; Pratama Putra, Z. Hybrid Cryptography and Steganography Method to Embed Encrypted Text Message within Image. *J. Phys. Conf. Ser.* **2019**, *1339*, 012061. [CrossRef]
27. Sulaiman, R.; Kirana, C.; Sugihartono, T.; Juniawan, F.P. RC4 Algorithm and Steganography to Double Secure Messages in Digital Image. In Proceedings of the 2020 8th International Conference on Cyber and IT Service Management (CITSM), Pangkal Pinang, Indonesia, 23–24 October 2020; pp. 1–4. [CrossRef]
28. Muhammad, K.; Ahmad, J.; Farman, H.; Zubair, M. A Novel Image Steganographic Approach for Hiding Text in Color Images Using HSI Color Model. *arXiv* **2015**, arXiv:1503.00388.
29. Ronaldo Cahyono, H.K.; Atika Sari, C.; Ignatius Moses Setiadi, D.R.; Hari Rachmawanto, E. Dual Protection on Message Transmission Based on Chinese Remainder Theorem and Rivest Cipher 4. In Proceedings of the 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 24–25 July 2019; pp. 74–78. [CrossRef]
30. Jaradat, A.; Taqieddin, E.; Mowafi, M. A High-Capacity Image Steganography Method Using Chaotic Particle Swarm Optimization. *Secur. Commun. Netw.* **2021**, *2021*, 6679284. [CrossRef]
31. Vishwas, C.G.M.; Kunte, R.S. An Image Cryptosystem Based on Tent Map. In Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 August 2020; pp. 1069–1073. [CrossRef]
32. Zhu, C.; Sun, K. Cryptanalyzing and Improving a Novel Color Image Encryption Algorithm Using RT-Enhanced Chaotic Tent Maps. *IEEE Access* **2018**, *6*, 18759–18770. [CrossRef]
33. Tayel, M.; Dawood, G.; Shawky, H. Block Cipher S-Box Modification Based on Fisher-Yates Shuffle and Ikeda Map. In Proceedings of the 2018 IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 8–11 October 2018; pp. 59–64. [CrossRef]
34. Sekertekin, Y.; Atan, O. An Image Encryption Algorithm Using Ikeda and Henon Chaotic Maps. In Proceedings of the 2016 24th Telecommunications Forum (TELFOR), Belgrade, Serbia, 22–23 November 2016; pp. 1–4. [CrossRef]
35. Alanazi, A.S. A Dual Layer Secure Data Encryption and Hiding Scheme for Color Images Using the Three-Dimensional Chaotic Map and Lah Transformation. *IEEE Access* **2021**, *9*, 26583–26592. [CrossRef]
36. Emad, E.; Safey, A.; Refaat, A.; Osama, Z.; Sayed, E.; Mohamed, E. A Secure Image Steganography Algorithm Based on Least Significant Bit and Integer Wavelet Transform. *J. Syst. Eng. Electron.* **2018**, *29*, 639. [CrossRef]
37. Wazirali, R.; Alasmary, W.; Mahmoud, M.M.E.A.; Alhindi, A. An Optimized Steganography Hiding Capacity and Imperceptibly Using Genetic Algorithms. *IEEE Access* **2019**, *7*, 133496–133508. [CrossRef]
38. Pund-Dange, S.; Desai, C.G. Data Hiding Technique Using Catalan-Lucas Number Sequence. *Indian J. Sci. Technol.* **2017**, *10*, 1–6. [CrossRef]

39. Güney Duman, M.; Duman, M. Encryption and Decryption of the Data by Using the Terms of the Lucas Series. *Duzce Univ. J. Sci. Technol.* **2021**, *9*, 1–7. [CrossRef]
40. El Hanouti, I.; El Fadili, H.; Zenkouar, K. Breaking an Image Encryption Scheme Based on Arnold Map and Lucas Series. *Multimed. Tools Appl.* **2021**, *80*, 4975–4997. [CrossRef]
41. Ehsan Ali, U.A.M.; Ali, E.; Sohrawordi, M.; Sultan, M.N. A LSB Based Image Steganography Using Random Pixel and Bit Selection for High Payload. *IJMSC* **2021**, *7*, 24–31. [CrossRef]
42. Kordov, K.; Zhelezov, S. Steganography in Color Images with Random Order of Pixel Selection and Encrypted Text Message Embedding. *PeerJ Comput. Sci.* **2021**, *7*, 1–21. [CrossRef] [PubMed]
43. Zouhri, A.; Amakdouf, H.; El Mallahi, M.; Tahiri, A.; Lakhliai, Z.; Chenouni, D.; Qjidaa, H. Invariant Gaussian–Hermite Moments Based Neural Networks for 3D Object Classification. *Pattern Recognit. Image Anal.* **2020**, *30*, 87–96. [CrossRef]
44. El Mallahi, M.; Zouhri, A.; El Affar, A.; Tahiri, A.; Qjidaa, H. Radial Hahn Moment Invariants for 2D and 3D Image Recognition. *Int. J. Autom. Comput.* **2018**, *15*, 277–289. [CrossRef]
45. Boukili, B.; Mallahi, M.E.; Amrani, A.; Zouhri, A.; Boumhidi, I.; Hmamed, A. A New Approach for H∞ Deconvolution Filtering of 2D Systems Described by the Fornasini–Marchesini and Discrete Moments. *Pattern Anal. Appl.* **2022**, *25*, 63–76. [CrossRef]
46. Boukili, B.; Mallahi, M.E.; El-Amrani, A.; Hmamed, A.; Boumhidi, I. H∞ Deconvolution Filter for Two-dimensional Numerical Systems Using Orthogonal Moments. *Optim. Control. Appl. Meth.* **2021**, *42*, 1337–1348. [CrossRef]
47. El Mallahi, M.; Boukili, B.; Zouhri, A.; Hmamed, A.; Qjidaa, H. Robust H∞ Deconvolution Filtering of 2-D Digital Systems of Orthogonal Local Descriptor. *Multimed. Tools Appl.* **2021**, *80*, 25965–25983. [CrossRef]
48. Abduljabbar, Z.A.; Abduljaleel, I.Q.; Ma, J.; Sibahee, M.A.A.; Nyangaresi, V.O.; Honi, D.G.; Abdulsada, A.I.; Jiao, X. Provably Secure and Fast Color Image Encryption Algorithm Based on S-Boxes and Hyperchaotic Map. *IEEE Access* **2022**, *10*, 26257–26270. [CrossRef]
49. Dhawan, S.; Chakraborty, C.; Frnda, J.; Gupta, R.; Rana, A.K.; Pani, S.K. SSII: Secured and High-Quality Steganography Using Intelligent Hybrid Optimization Algorithms for IoT. *IEEE Access* **2021**, *9*, 87563–87578. [CrossRef]
50. Sugiarto, B.; Sari, C.A.; De Rosal, I.M.S.; Rachmawanto, E.H. Performance Analysis of LSB Color Image Steganography Based on Embedding Pattern of the RGB Channels. In Proceedings of the 2020 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 19 September 2020; pp. 73–78. [CrossRef]
51. Giuclea, M.; Popescu, C.-C. On Geometric Mean and Cumulative Residual Entropy for Two Random Variables with Lindley Type Distribution. *Mathematics* **2022**, *10*, 1499. [CrossRef]
52. Das, P.; Ray, S.; Das, A. An Efficient Embedding Technique in Image Steganography Using Lucas Sequence. *IJIGSP* **2017**, *9*, 51–58. [CrossRef]
53. Chatterjee, A.; Ghosal, S.K.; Sarkar, R. LSB Based Steganography with OCR: An Intelligent Amalgamation. *Multimed. Tools Appl.* **2020**, *79*, 11747–11765. [CrossRef]