International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Reversible Data Hiding in Encrypted Images Based on Chaotic Logistic Map and Median Edge Detector

Bethala Shirisha[1]*, Dr. V. Kamakshi Prasad[2]

Submitted: 26/05/2023 **Revised**: 06/07/2023 **Accepted**: 25/07/2023

Abstract: The approach, called reversible data hiding in the encrypted image(RDH-EI), could be suitable for directly embedding more data in the encrypted domain without interfere on users' right to privacy. In this study, we present a completely new method for RDH-EI that uses the MED (Mean average Edge Detector) and the 2's complement algorithm. The Mean average Edge Detector prediction technique is used to construct the predicted values for the original pixels and quantify the prediction errors. A label is placed in the pixels to indicate the two's complement so that space can be reserved. Creating a label map and embedding it into the image to record the unidentified pixels is necessary. After completing the symmetric encryption, the picture could be included in the data. This paper talks about a way to encrypt and decrypt images using key sequences that are made up of sequences of logistic mappings and sequences of the states of LFSR. The results of the tests indicate not only that the embedding payload has been improved but also that the data hiding can be reliably retrieved and that the cover image can be recovered in its entirety and in good condition. In addition, the test results show that the embedding payload has been improved. We evalauted the suggested method in terms of embedded rate (ER), the MSE (Mean Square Error), and the entropy.

Keywords-*Encrypted images, Huffman coding, Reversible data hiding,Secure compression, Encryption,Chaotic map*

1. Introduction

Digital media has gradually made its way into people's everyday lives thanks to the rapidly growing popularity of smartphone handsets and the Internet and their fast growth. Nevertheless, security errors such as data loss frequently occur in a setting with an open network. The issue of how to make good use of technological tools to protect digital media is now a pressing practical issue that has to be handled as soon as possible. Studies on signal analysis in the encrypted domain based on cloud computing platforms and other robust privacy technologies have gathered growing more attention in present years [1–5]."Specifically, RDH-EI is a technique to securely store and manage digital images on the cloud while securing the images' secrecy. Cryptographic systems, however, need a lot of time and complexity.

In addition to this, they have several performance and security drawbacks [6]. Even though it is possible that current encryption methods will not be able to keep up with the needs of modern communication systems and data storage, many academics want to look into new and better ways to keep data secure [7]. Given this risk, there may be better ways than traditional cryptographic methods. After that, it was found that the essential characteristics of chaotic nonlinear systems, such as unpredictability and hypersensitivity, were well-suited to how cryptographic capabilities are created [8]. This detection came about as a result of the previous finding. Various researchers have created practical cryptographic approaches based on fractal geometry, and the secure features of these methods are discussed at length in [9]. On the other side, the data size can be effectively managed by utilizing several data compression methods during data transmission and data storage. Some examples of these methods include Lempel Zip (LZ), Huffman Coding (HC), and Arithmetic Coding. These methods can reduce the amount of data stored or transmitted (AC) [10]. Compression systems generate separate outputs because they are based on a range of concepts, all of which are suitable for certain types of input data. However, each compression method is based upon the same principle: to reduce the input data size by eliminating all redundant data.

For the last ten years, researchers have created a wide range of efficient RDHEI approaches [11, 12]. The two types of categories that can be found in recent RDHEI methods are (1) methods other than VRAE[13] and (2) methodologies by RRBE [14]. These categories are differentiated based on the different orders before or after encryption. When using VRAE techniques, encryption creates a place on encrypted images. On the other hand, making space might be problematic since encrypted pictures make a chaotic mess. VRAE methods are not high because of this embedding rates (ERs).

[1]*Research Scholar, JNTU Hyderabad,*
E-mail:- shirishasai34@gmail.com
[2]*Professor, JNTU Hyderabad*

On other side, before image encryption using RRBE techniques takes place, the content owner holds the space in reserve. After the data is encrypted, the hidden data can add it to the space the content owner has set aside. This enables the spatial relationship of the main image to be used. Therefore, RRBE strategies have the potential to attain a greater anchoring rate, which is one of the reasons why these strategies have been receiving an increasing amount of attention over the past several months. Even though the solutions suggested in [15] could significantly improve the ER, they need to make better use of the mistakes in the forecast and are therefore not recommended. In addition, to this the above methods the owner of the content is required to reveal extra data to the data hider. This is done in order to prevent the data from being discovered. This highly probable piece of information results in the data hider gaining access to the information included in the original image. This research paper proposes a new RRBE method based on the posterior probability to increase the ER while simultaneously reducing the danger of providing secondary data. As a result, they offer a new RDHEI approach that is both high and error-free. The method relies on predictions and Huffman coding. The earlier methods focuses more on the problems of encryption and compression algorithms so that compaction can be carried out using a secret key, and the outcome can be encrypted and condensed in a single movement. This is accomplished by combining the two problems into a single framework. To do this, the compression process needs to be done in a way that matches the hidden key. The proposed method can produce an ER greater than other comparable methods used in the past.

In the proposed approach, two's complement was utilized for encoding the misclassification rate, and spatial correlation was used to its maximum potential. As a result, more pixels are used to set separately space in the primary image. In the meantime, a label mapping is being built to record the spilled pixels as a substitute for embedding instructions within these pixels. Compressing the labeling map, one further cut in the room the secondary sources require can be obtained. This body of work presents a method that combines the key sequences generated by a logistic map with the value sequences generated by a linear shift register. The method is called a "combined key sequence." Encryption of an 8-bit grayscale image is done using the generated key sequences.

The following is the papers content: Section 2 will discuss about the related work. In Section 3, we cover some early basics of the work that will be done, as well as some basic work concepts. This section lays out the proposed method and its sub-sections in detail. The results of Section 4, in addition to an explanation on the efficiency and security of the proposed method, are presented here. In the final part of this article, Section 5, they will offer some final remarks.

2. Related Work

In [16], this work the author uses both multi MSB (most significant Bit) prediction and hierarchical quad-tree coding and suggests a new RDHEI method. This approach utilizes hierarchical quad-tree coding as its basis. The receiver extracts the data and recovers all the original images by using both data hiding and encryption. While starting image encryption process, the owner of the content performs the pixel predictions to generate a prediction error image. Next, hierarchical quad-tree coding is utilized in order to investigate the maximal hiding capacity of a prediction error image. Here data hider may also include extra data into the image of the vacated room by using the indicated bits of the capacity of the vacated room. Here the data hider don’t have any information regarding the main image content. The results of the studies show that the suggested method can achieve average imbed rates of 3.504 bits per pixel, 3.394 bits per pixel, and 2.746 bits per pixel on BOSS Base, BOWS-2, and UCID databases, respectively. These embedding rates are far higher than those achieved by some techniques to be considered.

In [17], here the authors proposed temporarily hiding data in image encryption (RDHEI). In order to encrypt an image, the embeddable pixels must first be selected from the original picture based on the prediction errors caused by neighborhood pixels that have a high correlation. First, all the pixels include sembedding pixels are reorganized, and then each pixel is encrypted on its own before the process is complete and a cipher text is produced. In order to make a tagged encrypted image, the secret bits must be strongly concentrated in the MSBs of the embedded pixel in the encoded image during the coding process. This is done in order to construct the secret image. It is possible to obtain secret bits from the various MSBs of the embeddable pixels that make up the encrypted image designated during the image's decrypting. In addition, the original embeddable pixels are recreated without any quality loss using the neighborhood pixels' correlation. Therefore, the only time it will be possible to obtain a reconstructed image with good visual quality is when the encryption key is present. Because the proposed approach uses several MSBs of the embedded pixels, it can obtain a higher capacity for embedding.

In [18], the authors revealed a method of RDH for encrypted 3D models are based on integer mapping and the estimation of the most significant bit. This technique

hides the data so it could not be decrypted. Content owner divides all of the triangles into "embedded" and "reference" and converts all the floating-point coordinates into integers. The encryption method is used after calculating the MSB forecast error of the "embedded" sets is done, and the results are added up. An MSB substitution approach could include any additional data needed. Depending on the permissions that have been given, lawful receivers can get either the initial meshes or some other data by employing the separable technique that has been described. Alternatively, they can obtain a combination of the two items. By using the MSB embedding technique, it is possible to get a higher hiding capacity, and by using the ring prediction scheme, it is possible to restore the original meshes completely.

In [19], the authors proposed a high-capacity RDHEI method based on a block-wise multi-predictor. This method also included improved Huffman coding. To begin with, the real image is separated into different blocks that do not overlap. There are sixteen different prediction algorithms used to find out what the current pixel should be based on the values in the bounding box. This is done in order to make the most of the correlation that is present in the image that is being captured. The exact prediction is used for all pixels that comprise a single block. The enhanced Huffman encoding is utilized when compressing the error image prediction and the ideal prediction model, which is calculated by the total of the absolute prediction errors. As a consequence of this, there will be additional room available for the embedding of data. Their results indicated that in terms of the capacity for producing a more expansive hiding room, their method is superior to some of the most recent methods that have recently been developed.

In [20], the authors constructed a reversible new approach to hiding data in encrypted images. Because of correlation, there is a good chance that the pixels located next to one another in the image have some subtle changes from one another. These changes are notably noticeable in the four bits of the pixel that are considered the most significant (the high nibbles). If the high byte of each pixel is thought to have the same value as a 4-bit value, the variation between the high nibbles of neighboring pixels are generally crammed into a small range. As a result of this, the Huffman coding algorithm is utilized in order to encode all of the differences that are present between the high nibbles of the neighboring pixels. This has been done in order to construct a space that has a large capacity and to effectively condense the four planes that make up the most significant bit (MSB). By using cryptographic algorithm the image is then encrypted once the room has been created, and the room is reserved in the encryption algorithm so that data can be hidden without the risk of being lost.

3. Preliminaries

Chaotic logistic map:

The Cryptologic Lattice Model (CLM), which exhibits a complex stochastic nature, satisfies fundamental cryptographic criteria, including volatility and unpredictability. Prominent among 1-*d* chaotic maps, the Chaos Logistic Map (CLM) verifies effective adjustment to starting conditions and parametric control values. CLM is specified by Equation 1 as follows:

$$x_{n+1} = F_\lambda(x_n) = \lambda x_n(1 - x_n) \quad (1)$$

Where $0 < x_n \leq 1$; $n = 0, 1, 2 \ldots$ .. λ is a parametric control variable with a value between 0 and 1, and x_0 is the starting condition. Figures 1 and 2, respectively, illustrate the results of the rigorous bifurcation analysis and the Lyapunov exponent investigation. Both of these studies were conducted in order to obtain accurate results. The numbers show that the CLM has a dynamic behavior that exceeds the nominal control value of 3.57, where orbits $\{x_n\}_{n=0}^{\infty}$ are evenly spread between 0 and 1.

In this current work, the CLM was meticulously constructed in order to provide a permutation key stream to be used in key management. CLM requires two input parameters to iterate and find the next value x_n. These parameters are an initial value referred to as x_0 and a parametric control value referred to as x_1. In addition, both the value x_0 and the value are considered secret keys, even though they are inputs. The x_n value is subsequently used to compute a value for the threshold *c*. The value of the predefined threshold is found to be $t = 0.5$ (in line with the uniform probability model), as indicated by (2):

$$B_n = \begin{cases} 0 & 0 \leq x_n < t \\ 1 & t \leq x_n < 1 \end{cases} \quad (2)$$

In order to create the pseudorandom ciphertext, real values xn of CLM are checked with a threshold value. By multiplying the chaotic sequence of X_i elements by 255, an unregistered integer from 0 to 255 can be derived from the sequence. Y_i is the number obtained by rounding the X_i value to the nearest decimal point.

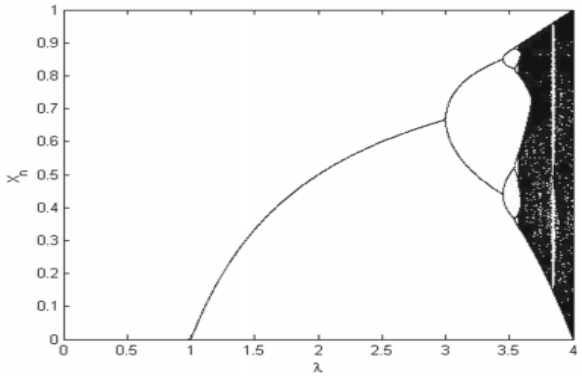

Fig 1: Bifurcation diagram of the CLM [14]

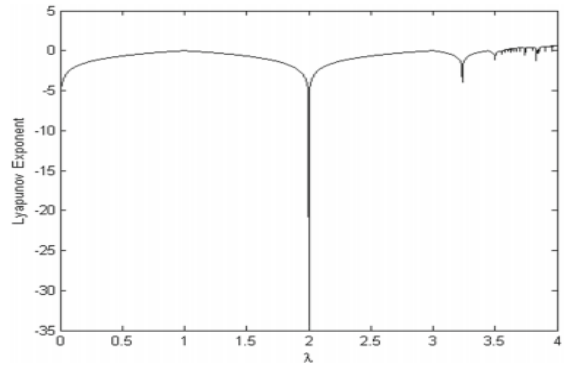

Fig 2: Lyapunov exponent of the CLM [14]

After the sequence Y_i is made, it is turned into a string of 8-bit words, which are then used to make the keyboard shortcut. This configuration is referred to as $K_{1,i}$ by us.

$K_{1,i}$=Round (X_i*255) (3)

8-bit LFBSR Sequence: This type of parity bit is called a linear feedback shift register, or LFSR (taps) for short. It happens when the memory cell's input bit results from a linear model of two or more of the register's previous states. An LFSR with a duration of *m* is made up of m phases, labeled 0, 1,...,*m*-1, each of which can store one bit of information and a clock that regulates data flow. The shift register could be loaded with a vector with elements$s_0,\ldots,s_{m-1}$. The actions listed below are carried out starting at time i.

In this situation, m = 8 is found using the polynomial $x_8+x_6+x_5+x_4+1$. There are 255 unique starting points from which to choose. Each beginning state, other than zero, will spawn a periodic series of succeeding states with a period equal to 2^{8-1} + 255. The period will increase by one for each successive state. The sequences produced by starting from a range of states are shifted versions of each other. For key template matching, the proposed scheme uses statistics of 8 bits. We refer to this pattern as the $\{K_{2,i}\}$sequence. Figure 3 depicts an 8-bit LFBSR using D_0 to D_7 as the starting value, also known as the seed. Taps and xor operations were done on the corners of the diagram labeled D_0, D_4, D_5, and D_6 to get the output bit and the input to the rightmost bit.

For every left shift the right most bit is affected by out bit.

$K_{2,0}$ = (D_0, $D_1\ldots D_7$), $K_{2,1}$= (D_1,$D_2\ldots D_8$), $K_{2,j}$= (D_j, D_{j+1},D_{j+2} D_{j+7}).

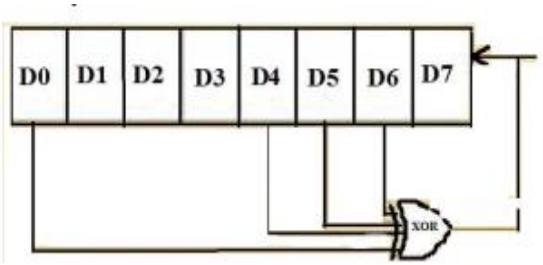

Fig 3: 8 bitLFBSR

4. The Proposed Work

This paper proposes a new method for error-free data extraction and picture decryption based on multi-MSB prediction and Huffman coding. The authors of this paper developed the method. The new method has a higher capacity than the previous method. The suggested procedure may be divided into three phases, as indicated in Figure 4: (i) The content owner is in charge of encrypting images; (ii) the data hider is in charge of embedding data; and (ii) the receiver is in charge of extracting data and decrypting images.

Label Map Generation using Complement Generation:

A prediction procedure is carried out with the real image, which has the dimensions $H \times W$ and pixels that have the values $p_o(i,j)(1 \le i \le H, 1 \le j \le W)$ within the range [0, 255]. This is done in order to generate the prediction errors. In the first step of the process, the MED prediction method is used to figure out which prediction should be used for each of the original pixels. The pixels recorded as references can be found in MED's first row and first column. Figure 5, which can be seen here, shows a simplified diagram of the MED.

Figure 5, p_1, p_2, and p_3 are three actual pixels enveloping the presently analyzed pixel $po(i, j)$. Then, the anticipated value $v(i, j)$ is calculated for each remaining original pixel $p_o(i, j)(2 \le i \le H, 2 \le j \le W)$ by using the following mathematical representation:

$$v(i,j) = \begin{cases} max(p_2,p_3), & p_1 \le min(p_2,p_3) \\ min(p_2,p_3), & p_1 \ge max(p_2,p_3), \\ p_2+p_3-p_1, & o \quad therwise \end{cases} \quad (4)$$

Finally, the actual pixel $p_o(i, j)$ and the expected value $v(i, j)$, as well as each prediction error $e(i, j)$ in the middle, may be obtained by using the following formula:

$$e(i,j) = p_0(i,j) - v(i,j) \quad (5)$$

Due to the correlation of natural pictures, the dispersion of prediction errors, which is given as $e(i, j)(2 \le i \le H, 2 \le j \le W)$, follows a Laplace dispersion with the crucial point set to zero. This is because natural images have a high degree of spatial correlation. Because of this, even the core bins of the distribution of prediction errors can be recorded across two complements. This is because of the way that the distribution is structured. Also, different lengths of two's complements can be used to encode the different "bins" of the distribution of prediction errors.

It is thought that the length of two's complements is symbolized by the sign α ($1 \le \alpha \le 7$), and that the range of prediction errors that a two's complement may hold is $[-2^{\alpha-1}, 2^{\alpha-1} - 1]$. Additionally, the length of two complements is believed to hold the information. This applies to pixels with an 8-bit depth (which is defined as U). This is made possible by the fact that the distribution of prediction errors can be binned.

After an is resolved, each pixel in $p_o(i, j)(2 \le i \le H, 2 \le j \le W)$ can be divided into two groups, one based on its presumed error and the other based on the value of U:

1. pixel labeled
2. pixel unlabeled.

One pixel which is considered to be labeled is the one that is related with the character *U* and has a prediction error. In addition, the pixel is considered unidentified if its validation loss is more than *U*, the threshold level.

Labeled pixel:

The prediction errors of a pixel that have been labeled can be recorded using an a-bit two's complement representation. Based on this, a way of labeling using two's complement is suggested to reserve space. First, we will turn each designated pixel into an 8-bit binary series using the $p_o(i, j)$ syntax. Every part $p_o(i, j)0$, $p_o(i, j)1, \ldots, p_o(i, j)^7$ of the binary series is found to be :

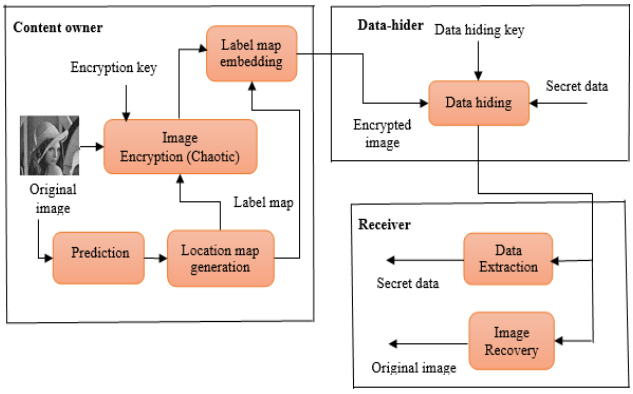

Fig 4: The framework of the proposed method

$$p_0(i,j)^k = \left\lfloor \frac{p_o(i,j)}{2^k} \right\rfloor mod\ 2, k = 0,1,\ldots.,7 \quad (6)$$

Where [ ] there is a floor function in this expression, the associated labeled pixel's prediction error $e(i, j)$ is encrypted by the a-complement, bit's 2's, and the value of each bit $e(i, j)$ k (k = 0, 1,... $\alpha-1$) can be obtained using the following formula:

$$e(i,j)^k = \begin{cases} \left\lfloor \frac{e(i,j)}{2^k} \right\rfloor mod\ 2, & e(i,j) \ge 0 \\ \left\lfloor \frac{2^n + e(i,j)}{2^k} \right\rfloor mod\ 2, & e(i,j) < 0 \end{cases} \quad (7)$$

In the end, $e(i, j)^k(k = 0, 1, \ldots, \alpha - 1)$ is encrypted into $p_o(i, j)^k$ (k = 0, 1, . . . , 7) by LSB to observe that the actual pixel could once again be covered which is seen in Figure 6. Once the prediction errors of the pixels that need to be predicted have been added up, the tagged pixels can be found using the complement of the two bits. As a result, the most significant bits (MSBs) of the first eight $(8 - \alpha)$ bits of labeled pixels can be employed to incorporate data.

Unlabeled pixel:

The prediction errors associated with these pixels cannot be labeled. After the data has been embedded, retrieving the unlabeled pixels is impossible. Therefore, the unmarked pixels cannot have any changes made to them.

Label Map Generation and Embedding:

Because of this, we cannot use the LSB to predict what each pixel is and cannot hide data in pixels that have already been labeled. A labeled map is applied to the labeled image in order to determine the category that each pixel belongs to.

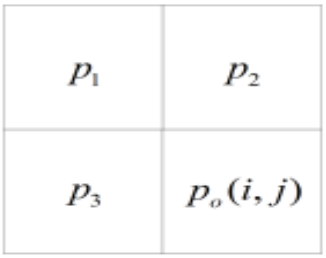

Fig 5: The context of the MED predictor

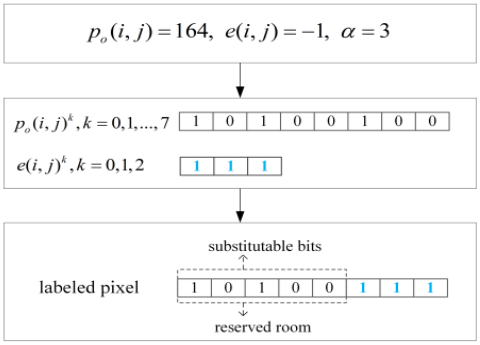

Fig 6: An example of pixel labeling

For the symmetry principle, a bitmap is created with the exact dimensions as the labeled image. This bitmap is referred to as the label map *M*. The coordinates of *M* and I_l. are related to one another in a way that exhibits symmetry. The *M* values can be customized by adjusting the pixel's $I_l$category. In line with the coordinates of the pixels that have been tagged, the values of the associated places in *M* have been reset to zero.

The spatial relationship between the photographs has led to the label map having a disproportionately high number of zeros and a relatively low number of ones. This is because the number of ones is relatively low. The other side, the values that correspond to *M*'s unlabeled pixels have all been set to 1, which is the default value. Based on this, the Huffman coding method [21] is applied to compress *M* without sacrificing any of its data. B_m is the name for the bit stream that is made when this compression is done.

Within the scope of this study, an MSB reorganization approach for embedding the label map is suggested. First, to construct a bit stream denoted by Bu, one must take the unlabeled pixels and extract their MSBs, which are $(8 - \alpha)$ bits long. The eight-bit most significant bits of each pixel in the processed image are reserved space, with the exception of the reference pixels. As a result, a room that may be used regularly for reservations is gained. Then, to guarantee that the unlabeled pixels can be retrieved, the tail of bitstream B_m is spliced onto bitstream Bu to create a new lengthy bit stream. This is done to make sure that the unlabeled pixels can be reclaimed once they have been deleted. Consequently, the pixel in question is unlabeled or labeled, the MSBs of continuous pixels may be utilized to store the lengthy bit stream.

In the final step, image II is swept from left to right and from top to bottom while the lengthy bitstream is simultaneously merged into each pixel's most significant bit (MSB). This completes the process. This is accomplished through the use of bit substitution. It is essential to know that the referenced pixels are not utilized in embedding the extended bitstream. This is a crucial point to keep in mind. After embedding the lengthy bitstream has already been finished, the coordinate of the final pixel in the embedding region is acquired. The coordinate is written down and then used as the variable C_p value to get the extended bitstream from the image. For example, C_p = (123, 45). Before dividing the lengthy bitstream into B_m and B_u, it is necessary to calculate the length of B_m, which is indicated by the parameter Lm. This will allow the bitstream to be divided into B_m and B_u. For instance, L_M = 162,341. It is possible to produce the processed image I_p by going through the label map implanting procedure. Starting at the point C_p in the processed image, the first eight bits of the most significant bit (MSB) of each component pixel show where the embedded data.

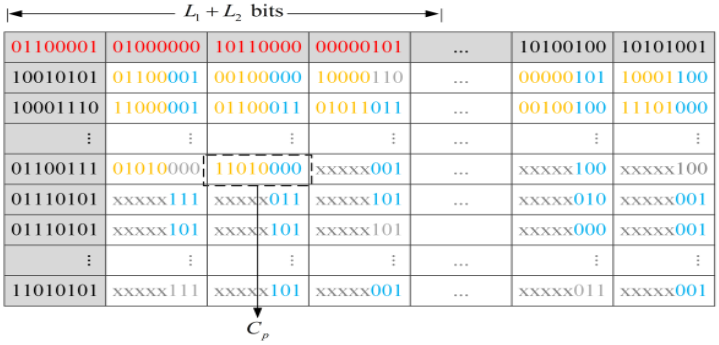

Fig 7: Illustrative example of the processed image

We can see an example of I_p in Figure 7, where the value is α =3. The variables are shown in Figure 7 by the red bits included within reference pixels. The three bits comprising each pixel's least significant bit (LSB) have different meanings. For example, the blue bits represent the two's complement, whereas the grey bits represent the original bits. The entirety of the binary representation is indicated by the yellow bits. Additionally, the pixel's location encompassed by the dashed box is written down and given the value C_p. The bits that start with xxx denote the location of the reserved room.

Proposed Image Encryption Algorithm:

During this phase, the real image being processed is encrypted using the following approach so that the content of the actual image can be secure.

The proposed scheme of the encryption process is discussed below:

The first step involves converting an 8-bit grayscale image with a size of MN pixels into a one-dimensional array of pixels using the formula P_i={P_1, P_2,…..P_n}, where i=1, 2, 3... n=M×N. The next step is to transform each unsigned pixel value, ranging from 0 to 255, into an 8-bit block.

The second step involves A bit-by-bit XOR operation performed between the previously created sequences {$K_{1,i}$} and {$K_{2,i}$} in order to obtain the final key sequence {K_i}.

K_i=$K_{1,i}$③$K_{2,I}$(8)

In the third step, the binary image pixels known as Pi are XORed with the key sequence known as Ki to produce the encrypted pixel known as {C_i}. These 8-bit blocks are then transformed into the decimal digit range of 0 to 255 using the {C'$_i$} notation.

C_i=P_i③K_i (9)

The fourth step involves to encrypt all of the image pixels and repeating the previous step (step 3). In order to acquire the encrypted image, we must first convert all of the encrypted digits, C' ={C' ,C' ,C' …C'}, into an array of the dimensions MN.

Data Hiding in the Encrypted Image

The data about the reserved room is included in the encrypted image so the data hider can embed it. The L_1 bits of that image are transmitted in plaintext. Therefore, after the hider has the encrypted image. The data is encrypted with a data encryption key known as K_d to increase the level of security even further. First, the L_1 bits are recovered from the fixed reference pixels by extracting the bits, and then the parameters and C_p are computed based on the bits using the recovered L_1 bits. This process is repeated until all of the L_1 bits have been recovered. Suppose it is assumed that N_p is the number of un-embedded images andN_r is the number of reference pixels. In that case, the effective payload may be estimated using the parameters. Because of this, the payload can be determined using

$Payload = (H \times W - N_r - N_p) \times (8 - a)$

Data Extraction and Image Recovery

In the 2 cases while the decoding level depends on the receiver with different keys: (1) Data secret hiding key K_d or (2) the key to the image encryption, K_e. Using the different keys, the user who gets the message can get confidential information and the original image.

Data Extraction

After obtaining the parameters and Cp from the L1 bit reference pixels, the receiver then identifies the reserved room using its obtained parameters. The hidden information is then collected from the MSB bits of the matching (8 − α)-bit pixels to complete the process. In the end, the encrypted secret data is deciphered with the help of K_d. If the recipient possesses K_d the confidential data may be recovered from the designated encrypted image I_m.

5. Results and Discussions

The simulation of the photo encryption and decryption method that has been proposed can be carried out with the help of MATLAB, which is the tool that is used. Test images consisting of an 8-bit Lena and a medical greyscale image with a size of M by N pixels (where M = N = 256) have been selected for this research. In this part, we will first evaluate the safety of the suggested approach. After that, we will provide the findings of our experiments and then compare them to the most recent research in this field. Figure 8 displays six frequent test photos, including Tiffany, Lena, a baboon, a jetplane, and a man. Also included are photographs of a jet plane. The most important parameter is the embedding rate, also known as ER. It is measured in bits per pixel or bpp. Also, measurements like PSNR and SSIM are used to determine if the technique can be reversed.

(i) Lena (ii) Jetplane (iii) Baboon

(iv) Tiffany (v) Airplane (vi) Man

Fig 8. The test images.

Security Analysis:

We hide the original picture's feature information by encrypting it with a stream cipher that was used to encrypt the image in the first place. In order to demonstrate that the strategy that was suggested is dependable, we will investigate it from two distinct points of view, namely, statistical features and probability. Let us take Lena's image for reference. Figure 9 shows what happens when our method is applied to the different parts of an experiment.

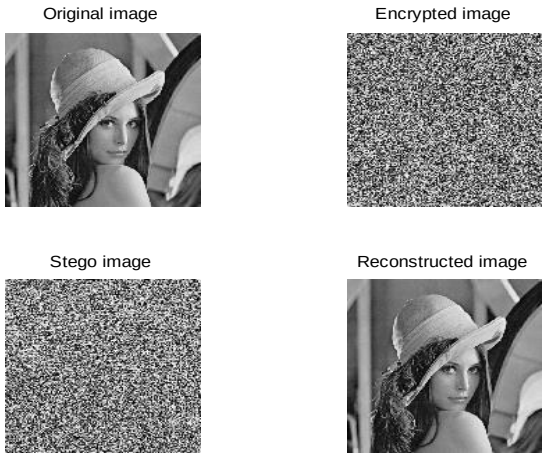

Fig 9: Approach that has been suggested, together with the outcomes of each stage of the experiment

Table 1 displays the auxiliary information for six different test images, the amount of net payload computed, and the tests' outcomes.

Table 1: The EC and the auxiliary information of test images.

Test Images	Extra bits	Payload (bpp)
Lena	52	2.684
Baboon	52	1.187
Jetplane	52	3.142
Man	52	2.541
Airplane	52	3.781
Tiffany	52	2.984

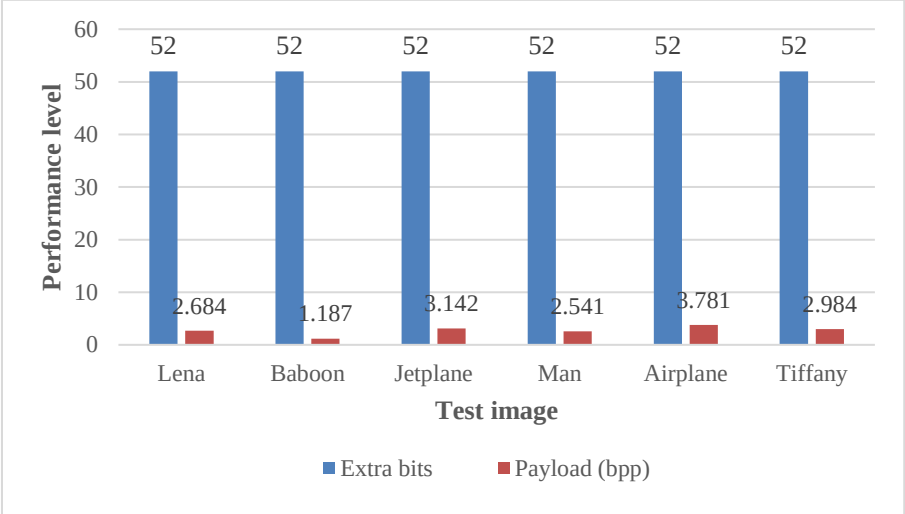

Fig 10. Comparison of ER (bpp) on six test images

Parameter and Capacity Analysis:

In the method shown, prediction errors are encoded using a process called a bit two's complement. It has already been proven that the range of the coded prediction errors changes depending on the value; however, this shift was not anticipated. Three distinct components must be considered to evaluate the parameter's impact: the number of labeled pixels, the compression rate of the label map, and the ER. Until then, α is picked between 1 and 7 and selected to investigate the test photos. The following is an analysis and presentation of the outcomes of the experiment: The number of labeled pixels present in each test image when given a variety of values, as shown in Table 2.

In order to more clearly present the information shown in Table 2, Figure 11 has been crafted to illustrate how the number of labeled pixels fluctuates. Image 11 demonstrates a correlation between the value and an increase in the count of labeled pixels. If the value is less than or equal to seven, the number of labeled pixels in each image is highly close to equal to the total number of pixels. For example, when x equals 7, the total number of pixels in the image of Tiffany, excluding the pixel that serves as the reference point, is 261,121.

Table 2: The labeled pixels' number of each test image under various α.

α -bit complement	Test Images		
	Lena	Jetplane	Tiffany
1	52994	78498	60,236
2	98612	136847	109,227
3	165803	197413	176,759
4	224748	223660	228,156
5	250541	252290	250,614
6	259355	259738	259,381
7	261001	261047	261,080

However, the number of labeled pixels in Tiffany is only 261,080. When it is made smaller, it becomes more difficult for the prediction error of the pixel to fit inside the interval regarded as the α-bit complement. The approach that was proposed. Therefore, as the range increases α, the total count of pixels that have been tagged also increases.

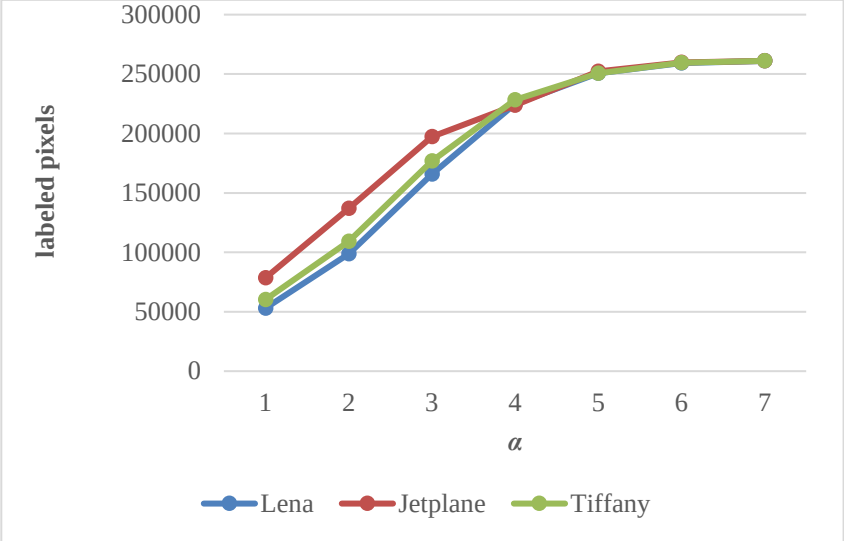

Fig 11: Number of test images under various αin the Labeled pixels.

MSE(Mean Square Error Analysis): This is a parameter used to find the difference between the original image and the encrypted version of the image in which pixels are represented between 0 and 255. This difference is measured using MSE, which stands for Mean Square Error Analysis.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M \times N-1} [F(i,j) - G(i,j)]^2 \quad (10)$$

Where $F(i, j)$ refers to a pixel from the original picture and $G(i, j)$ refers to a pixel from the encrypted image, M and N refer, respectively, to the size of the encrypted image or the image was not encrypted. If an image is encrypted, MSE must be as high as possible. It is more difficult to break if the encrypted image has a mean square error (MSE) lower than the one originally used to encrypt the data.

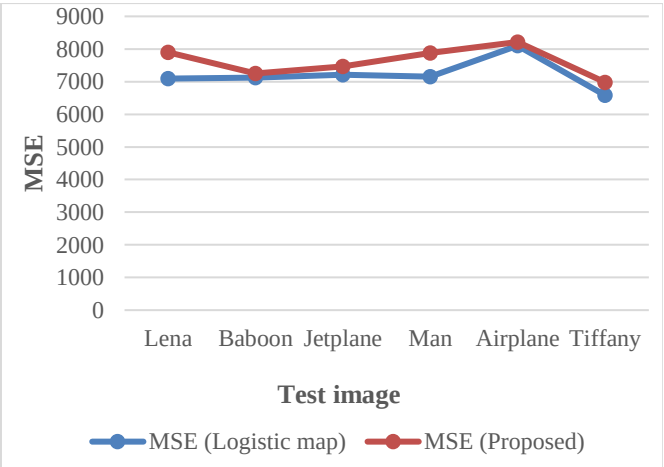

Fig 12: MSE between original and encrypted image

Table 3: MSE between original and encrypted image

Image type	MSE (Logistic map)	MSE (Proposed)
Lena	7097.2	7894.2
Baboon	7125.2	7254.2
Jetplane	7215.4	7471.4
Man	7154.7	7874.7
Airplane	8101.1	8214.1
Tiffany	6580.4	6980.4

Figure 12 demonstrates that the mean squared error (MSE) between the original image and the encrypted image that was generated for the Lena image by using the logistic map is 7097.2, whereas the MSE for the suggested approach is 7894.2. This can be seen by comparing the two values.

Comparatively, the MSE values produced for the airplane image by applying the logistic map approach were 8101.1, whereas the suggested method yielded 8214.1. Based on these observations, it can be deduced that the suggested encryption method is superior to the one that uses the logistic map.

Entropy of the image: With the help of information entropy, one thing that can be looked into is how well the information content can be predicted. $H(c)$ denotes the information entropy for source c, which is defined as

$$E(x) = \sum_{i=0}^{M \times N-1} P(C_i) log_2 \frac{1}{P(C_i)} \quad (11)$$

In the equation (11), C_iM×N is the total number of symbols, and $P(C_i)$ is the chance that the symbol will appear. Theoretically, the highest value of entropy that can be achieved with an 8-bit grayscale picture is eight symbols. This happens when the histogram shows that all pixel values have the same chance of being true or when the histogram is flat. The entropy of a picture reveals the value distribution throughout the grayscale. There is more entropy when the grey levels are spread out more consistently. The entropy of the original picture and the encrypted version of the image is calculated for two different image scenarios, and the findings are summarized in Table 4.

Table 4: Entropy of original and encrypted image

Image type	Entropy (Logistic map)	Entropy (Proposed)
Lena	7.9744	7.9988
Baboon	7.2252	7.3547
Jetplane	7.3474	7.6414
Man	7.1457	7.5468
Airplane	8.1011	8.2147
Tiffany	6.2874	6.9804

There are two situations in which the entropy of the original image is lesser than the entropy of the encrypted image. According to Figure 13, the entropy difference between the encrypted image and the original image that was computed for the Lena image by making use of the logistic map is 7.9744, but the entropy difference between the original image and the technique that was suggested is 7.9988. Similarly, the entropy discrepancy between the encrypted image and the origial image computed for the airplane image utilizing the logistic map is 8.1011, whereas the entropy difference using the suggested approach is 8.2147. When compared to the entropy of an image encrypted using merely the logistic map approach, the entropy of an image encrypted using the suggested technique was significantly higher. In case of suggested method, the pixels distribution is consistently compared to the performance of encryption using a sequence constructed using a logistic map. As a result, the suggested approach is more resistant to statistical assaults.

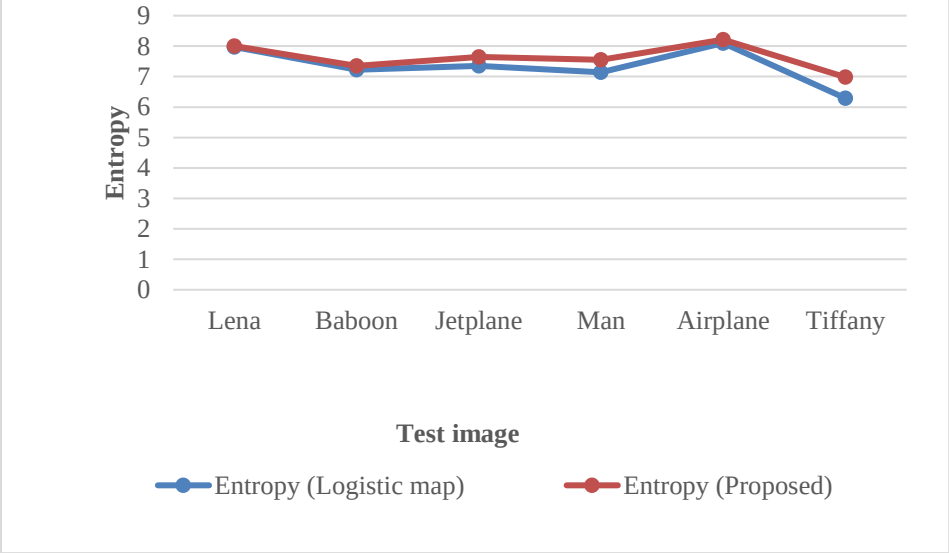

Fig 13: Entropy between original and encrypted image

6. Conclusion

This research paper presents a new method for RDHEI, which is based on MED(Mean average Edge Detector) and 2's complement. In the beginning, the image is given the handling known as the MED prediction approach so that a prediction may be made of the pixels. After that, the errors in the predictions can be computed by comparing the values of the original pixels with the values that were predicted for them. The acceptable interval is encoded by an a-bit two's complement, and the distribution of prediction errors is used to select this encoding. Later, a scheme was made to encrypt and decrypt images using a key sequence formed from the logistic map's sequence and the LFSR states. This key sequence would be formed from the sequence of the logistic map. The suggested technique has a high degree of sensitivity toward the starting value and the initial state of the LFSR. It is further demonstrated that decryption using the incorrect key, which involves only a relatively minor adjustment to the key sequence's beginning value, produces an entirely different image. The approach that has been described can explore a more considerable data embedding space, which results in a higher net payload. One of the benefits of utilizing Huffman coding is that it allows for improved compression of the label map and frees up more space that can be used for embedding information. When the results of several studies are compared, it is also clear that the strategy suggested has the potential to get a higher ER than other methods that have been used in the past.

References

[1] Zheng, Peijia& Huang, Jiwu. (2011). Implementation of the discrete wavelet transform and multiresolution analysis in the encrypted domain. MM'11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops. 413-422. 10.1145/2072298.2072352.

[2] Xiang, S.-J & Luo, X.-R & Shi, S.-X. (2016). A novel reversible image watermarking algorithm in homomorphic encrypted domain. 39. 571-581. 10.11897/SP.J.1016.2016.00571.

[3] Zhou, Jun & Cao, Zhenfu& Dong, Xiaolei& Choo, Kim-Kwang Raymond. (2019). Efficient Privacy-preserving Outsourced Discrete Wavelet Transform in the Encrypted Domain. IEEE Transactions on Cloud Computing. PP. 1-1. 10.1109/TCC.2019.2948012.

[4] Bellafqira, Reda &Coatrieux, Gouenou & Bouslimi, Dalel&Quellec, Gwenole. (2015). Content-based image retrieval in homomorphic encryption domain. Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference. 2015. 2944-2947. 10.1109/EMBC.2015.7319009.

[5] Yi, Shuang & Zhou, Yicong. (2015). An improved reversible data hiding in encrypted images. 225-229. 10.1109/ChinaSIP.2015.7230396.

[6] Xiaolin, Yi & Nanzhong, Chen &Zhigang, Jia & Xiaobo, Chen. (2010). Trusted Communication System Based on RSA Authentication. 329 - 332. 10.1109/ETCS.2010.460.

[7] Puangpronpitag, Somnuk & Kasabai, Piyawad & Pansa, Detchasit. (2012). An enhancement of the SDP Security Description (SDES) for key protection. 1-4. 10.1109/ECTICon.2012.6254320.

[8] LeMay, Michael &Rakshit, Joydeep& Deutsch, Sergej& Durham, David & Ghosh, Santosh & Nori, Anant & Gaur, Jayesh &Weiler, Andrew & Sultana, Salmin& Grewal, Karanvir & Subra money, Sreenivas. (2021). Cryptographic Capability Computing. 253-267. 10.1145/3466752.3480076.

[9] Huang, Jianzhe. (2012). Analytical dynamics of period-m flows and chaos in nonlinear systems. Int J Bifurcat Chaos 22. Article.

[10] Mbewe, Phyela & Asare, Sampson. (2017). Analysis and comparison of adaptive huffman coding and arithmetic coding algorithms. 178-185. 10.1109/FSKD.2017.8393036.

[11] Hong, W.; Chen, T.S.; Wu, H.Y. An improved reversible data hiding in encrypted images using side match. IEEE Signal Process. Lett. 2012, 19, 199–202.

[12] Zhang, W.; Ma, K.; Yu, N. Reversibility improved data hiding in encrypted images. Signal Process. 2014, 953, 118–127.

[13] Hong, W.; Chen, T.S.; Wu, H.Y. An improved reversible data hiding in encrypted images using side match. IEEE Signal Process. Lett. 2012, 19, 199–202.

[14] Mathew, T.; Wilscy, M. Reversible data hiding in encrypted images by active block exchange and room reservation. In Proceedings of the 2014 International Conference on Contemporary Computing and Informatics, Mysore, India, 27–29 November 2014;pp. 839–844.

[15] Wu, Y.; Xiang, Y.; Guo, Y.; Tang, J.; Yin, Z. An improved reversible data hiding in encrypted images using parametric binary tree labeling. IEEE Trans. Multimed. 2020, 22, 1929–1938.

[16] Liu, Ya& Feng, Guangdong & Qin, Chuan& Lu, Haining& Chang, Chin-Chen. (2021). High-Capacity Reversible Data Hiding in Encrypted Images Based on Hierarchical Quad-Tree Coding

and Multi-MSB Prediction. Electronics. 10. 664. 10.3390/electronics10060664.

[17] Wang, Dewang & Zhang, Xianquan& Yu, Chunqiang & Tang, Zhenjun. (2020). Reversible Data Hiding in Encrypted Image Based on Multi-MSB Embedding Strategy. Applied Sciences. 10. 2058. 10.3390/app10062058.

[18] Xu, Na & Tang, Jin& Bin, Luo & Yin, Zhaoxia. (2021). Separable Reversible Data Hiding Based on Integer Mapping and MSB Prediction for Encrypted 3D Mesh Models. Cognitive Computation. 10.1007/s12559-021-09919-5.

[19] Zhang, Huiqi& Li, Lin & Li, Qingyan. (2021). Reversible Data Hiding in Encrypted Images Based on Block-Wise Multi-Predictor. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3072376.

[20] Chen, Chih-Cheng & Chang, Chin-Chen & Chen, Kaimeng. (2021). High-capacity Reversible Data Hiding in Encrypted Image Based on Huffman Coding and Differences of High Nibbles of Pixels. Journal of Visual Communication and Image Representation. 76. 103060. 10.1016/j.jvcir.2021.103060.

[21] Yin, Zhaoxia& Xiang, Youzhi& Zhang, Xinpeng. (2019). Reversible Data Hiding in Encrypted Images Based on Multi-MSB Prediction and Huffman Coding. IEEE Transactions on Multimedia. PP. 1-1. 10.1109/TMM.2019.2936314.

[22] Renato Costa, Deep Reinforcement Learning for Autonomous Robotics , Machine Learning Applications Conference Proceedings, Vol 2 2022.

[23] White, M., Hall, K., López, A., Muñoz, S., & Flores, A. Predictive Maintenance in Manufacturing: A Machine Learning Perspective. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/154

[24] Timande, S., & Dhabliya, D. (2019). Designing multi-cloud server for scalable and secure sharing over web. International Journal of Psychosocial Rehabilitation, 23(5), 835-841. doi:10.37200/IJPR/V23I5/PR190698