A Review on Security Techniques in Image Steganography

Sami Ghoul[1], Rossilawati Sulaiman[2], Zarina Shukur[3]
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia UKM Bangi 43600, Malaysia[1, 2, 3]
Department of Computer Systems Engineering-Faculty of Engineering, University of Zawia, Zawia, Libya[1]

***Abstract*—Given the increased popularity of the internet, the exchange of sensitive information leads to concerns about privacy and security. Techniques such as steganography and cryptography have been employed to protect sensitive information. Steganography is one of the promising tools for securely exchanging sensitive information through an unsecured medium. It is a powerful tool for protecting a user's data, wherein the user can hide messages inside other media, such as images, videos, and audios (cover media). Image steganography is the science of concealing secret information inside an image using various techniques. The nature of the embedding process makes the hidden information undetectable to human eyes. The challenges faced by image steganography techniques include achieving high embedding capacity, good imperceptibility, and high security. These criteria are inter-related since enhancing one factor undermines one or more others. This paper provides an overview of existing research related to various techniques and security in image steganography. First, basic information in this domain is presented. Next, various kinds of security techniques used in steganography are explained, such as randomization, encryption, and region-based techniques. This paper covers research published from 2017 to 2022. This review is not exhaustive and aims to explore state-of-the-art techniques applied to enhance security, crucial issues in the domain, and future directions to assist new and current researchers.**

***Keywords*—Image steganography; data hiding; steganographic security; randomization; encryption**

I. Introduction

With the evolution of the internet and social networking as well as a rapid increase in communication facilities, a huge amount of information is being exchanged every moment. Security and privacy of exchanged data should be guaranteed, especially against malicious threats. Cryptography and steganography are techniques that are being used to achieve high security [1]. Cryptography is the process of converting raw information or plaintext to unreadable form called ciphertext, using an encryption algorithm with a key. The algorithm and the key utilized by the sender are, in most cases, used by the receiver for the decryption purpose. Hence, the original data is unreadable, and confidentiality is maintained. However, the availability of the ciphertext raises suspicions and draws the attention of opponents. On the other hand, steganography seems to be more appropriate and has recently received much attention, since it does not give rise to any signs detectable by the human eyes [2]. Steganography is the science of hiding relatively smaller information in a larger multimedia cover. Cover media could take the form of text [3], image [4], audio [5], or video [6]. Image steganography is the process of concealing secret data within an image that appears normal to the human eye.

Several reviews on image steganography have been published in recent years. Study [7] reviews and compares various deep learning methods in the domain of image steganography. It categorizes these methods into three types: traditional, Convolutional Neural Network based, and General Adversarial Network based methods. It also discusses the datasets, experiments, and metrics used in the field. However, traditional steganographic methods are not discussed in this article. The authors of research [8] provide a comprehensive overview and evaluation of some of the latest steganographic methods. This article addresses the challenges of recent deep learning-based steganographic methods. It also discusses how to measure the performance of a steganographic technique using various criteria. Although it evaluates the security of the techniques vis-à-vis varied advanced attacks and tools, randomization-related security concepts, such as the chaotic map function, have not been mentioned or discussed.

The research [9] presents a review and a critical assessment of recently proposed steganography methods. It describes various schemes, along with their technical terms, main logics, as well as strengths and weaknesses in terms of important measures. This critical assessment is based on the type of cover object used, the algorithmic domain, and important properties used as evaluative measures for the steganographic system. However, this article might be outdated, because many of contributions have been published since then.

This research article provides an extensive and comprehensive review of the security principles used in spatial-domain image steganography. It presents and discusses various steganographic techniques according to the security concepts adopted by them, such as randomization, encryption, and adaptive embedding. Further, it provides an overview of image steganography, its goals, and traditional steganography methods along with their features, pros and cons, and performance evaluation metrics. It also provides analysis, discussion, and suggestions based on its study of image steganography security principles.

The remainder of this paper is organized as follows: Section II presents an overview of image steganography. Section III demonstrates the basic goals of image steganography. In Section IV, the search process is defined, followed by detailing of security techniques in image steganography in Section V. Section VI contains observations,

discussion, and recommendations. Finally, Section VII concludes the paper.

II. An Overview of Image Steganography

The word steganography is of Greek origin and is constructed from two words: "steganos" and "graphie". "Stegano" means covered and "graphie" means writing. Steganography is not a new art; it was used in ancient times to send secret messages by hiding them in certain ways. Fig. 1 summarizes the two ideas used to secure secret information [10].

Generally, cryptography is divided into symmetric and asymmetric keys, based on whether it provides authentication or confidentiality. Likewise, steganography techniques can be classified into spatial and frequency domain techniques. In the spatial domain, the secret information is directly embedded in image pixels; hence, pixel values are modified. On the other hand, in the frequency or transform domain, the image pixels are transformed into another domain, and the secret information is embedded by modifying the coefficient values. Image steganography is the process of embedding secret information within a cover image, wherein the embedded information is not visible to the human eye. The output image, which contains the secret information, is called a "stego image" [11]. Fig. 2 depicts a general steganographic system, wherein the sender optionally compresses or/and encrypts the secret information and then a certain steganographic algorithm is utilized to embed the information and produce a stego image. At the receiver's end, the same sequence is followed in reverse to obtain the embedded information.

A. Traditional Steganography Methods

As mentioned above, steganography techniques are classified into spatial domain and frequency domain techniques. In spatial domain techniques, pixel values are modified directly to incorporate the secret information. These techniques are famous for attaining high capacity, but they are not immune to statistical attacks and image manipulations [12]. Examples of spatial domain methods include Least Significant Bit (LSB) Replacement and its successors, Pixel Value Differencing (PVD), Pixel Indicator Techniques (PIT), and Exploiting Modification Direction (EMD) techniques, among others. In the transform or frequency domain techniques, the image pixels are first transformed into the frequency domain. Subsequently, the secret information is hidden by modifying the coefficient values. Finally, the inverse transform is applied to obtain the stego image. These techniques resist statistical attacks but achieve low capacity. The most well-known transform domain methods are the Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Discrete Contourlet Transform (CT), and Discrete Wavelet Transform (DWT) [12],[13].

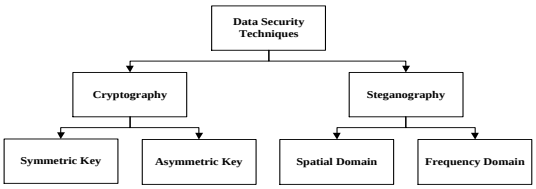

Fig. 1. Data security classes [7].

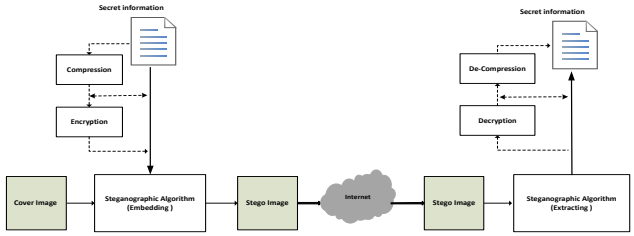

Fig. 2. General image steganographic system.

LSB Replacement (or simply LSB) is a well-known spatial domain technique, widely utilized due to its simplicity and potential to achieve high capacity. In this method, the secret information is embedded within the least significant bit of the cover image pixels, in order to minimize distortion. The resultant stego image is indistinguishable from the original image. Further, to increase payload capacity, more than one LSB of a pixel may be used to hide information; however, this might degrade the visual quality of the stego image [14]. It is worth mentioning that several enhancements of the original LSB have already been introduced. LSB Matching (LSBM) is an enhanced version of the LSB method [15]. Here, if the embedded bit does not match the cover pixel's LSB, then +1 or -1 values are added randomly to that pixel, so as to avoid the asymmetry artefacts that are usually introduced by the standard LSB technique and can be detected by steganalysis techniques [16]. To improve upon the previous methods, the LSB Matching Revisited (LSBMR) method was introduced by [17]. It embeds secret information within the LSB, with minimum changes to the carrier image. It hides two secret bits into two pixels simultaneously, wherein the first bit is embedded directly, while the second secret bit value is produced based on the relationship between those two bits. The objective is to make the detection of the secret information more difficult, compared to standard techniques [14], [17].

Another approach towards embedding secret information in the cover image consists of hiding bits in the edge area where pixels' intensity values change abruptly. Such techniques are referred to as Edges Based Embedding (EBE) Steganography, which allows the hiding of large payloads in those particular edge pixels. Several research studies have been published in this context, such as [16],[18],[19],[20]. PVD is another method of hiding binary data, wherein the cover image is considered to be in the form of non-overlapped blocks of two pixels each. The difference between the two pixels is calculated and quantized into several regions that determine the number of bits of the payload [21]. In the Cyclic Steganographic Technique (CST), the embedding process is cycled through color channels of consecutive pixels [22]. The color channel selected for the current pixel is not the same color channel used in the previous pixel, or the next pixel. For example, if the LSB of the red channel is selected for the current pixel, then the green channel is selected for the next pixel, and the blue channel for the following pixel. Hence, the same pattern is preserved for three consecutive pixels. The concept of randomization along with CST is proposed in [23] wherein secret information is randomly hidden in the pixels' LSB.

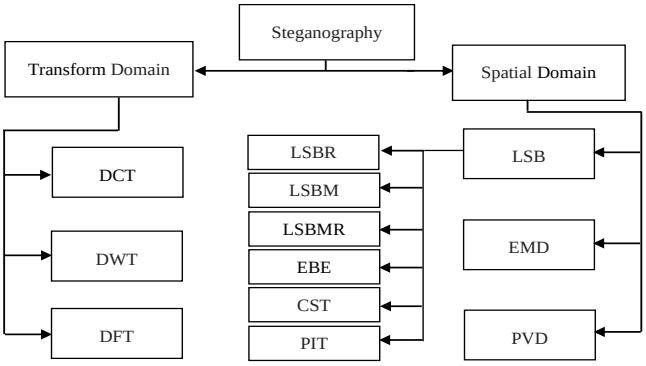

Fig. 3. Traditional steganographic techniques.

PIT is another variation of LSB techniques and is used to enhance the security and robustness of classical systems. In this method, one of the color channels of the pixel is selected as an indicator for the other two channels that are used for the embedding process [12]. An improvement is introduced in [24] takes into consideration the length of the secret message and uses two LSBs of a particular channel to indicate the presence of data in other channels. Another variation is introduced in [25]; it uses three LSBs of one of the pixel's channels as an indicator. EMD is another technique used to enhance security. In this method, the cover image is partitioned into blocks of n pixels. These n pixels of the cover image involve secret digits in a 2n+1-ary notational system. One pixel is eventually altered by ±1. For a group of n, there are 2n+1 possible digits to be secretly hidden, since there are 2n possible pixels' modifications and one case with no changes [26]. Article [27]presents a scheme to improve the EMD method called improved EMD (IEMD). This scheme uses an 8-ary notation system, wherein the secret digit is embedded into a group of two pixels. Fig. 3 summarizes the steganographic techniques explained above.

III. Basic Goals of Image Steganography

Several considerations must be taken into account while designing a steganography algorithm: capacity, imperceptibility, and security. The ultimate goal is to attain high capacity, better imperceptibility, and high security. However, these considerations are conversely related, and a trade-off needs to be made among the criteria mentioned above. More details are given in the following section [13],[28],[29].

A. Capacity

It refers to the amount of data in bits that can be embedded in the cover image. The objective is to hide as many bits as possible in the cover image, without affecting image imperceptibility and security. Embedding rate or Bits Per Pixel (BPP) is another widely-used term, which refers to the total amount of information relative to the total number of the cover image pixels.

B. Imperceptibility

When the secret information has been hidden, the resultant stego image should not create any signs that might be suspected by human eyes. Hence, the aim is to make the level of deterioration as low as possible. High imperceptibility means low capacity and vice versa. Hence, these two concepts need to be balanced.

C. Security

Security is the main key to secure information sent over an unsecured network such as the internet. It refers to the ability to withstand the detectability of hidden secret information in the cover image as well as the capability of extracting it. Accordingly, the steganographic algorithm should resist the attacker's attempts to detect and extract the secret information.

As stated before, the aim is to achieve a high embedding rate, high imperceptibility, and high security. However, this is a challenging task since these metrics compete against each other. In other words, focusing on one of them will sacrifice one or more other metrics. For example, embedding high payloads will definitely lead to low imperceptibility and the possibility of low security.

IV. Search Process

A. Materials and Methods

This review study examines the existing methods and techniques of image steganography security researched between 2017 and 2022. Only studies that involved spatial image steganography are considered. This section includes subsections on data sources, search processes, data selection, and data extraction.

B. Data Sources

The search and downloading phase has been implemented intermittently over the period from November 2021 to February 2022. The primary sources for the research have been selected from the following libraries:

- Institute of Electrical and Electronics Engineers Xplore Digital Library (https://ieeexplore.ieee.org/Xplore/home.jsp.
- Science Direct (https://www.sciencedirect.com/).
- Springer Link (https://link.springer.com/).
- Google Scholar (https://scholar.google.com/)
- Association for Computing Machinery (ACM) Digital Library (https://dl.acm.org/).

In addition, some other resources have been considered as well, such as the International Journal of Advanced Computer Science and Applications (IJACSA).

C. Search Process

The search has focused on image steganography security and various keyword patterns have been used in this process. Boolean operators have helped refine the data on keywords for each research publication. Symbols and Boolean operators such as "OR" and "AND" have been used to look for the following keywords:

- ((digital image steganography) OR (security in image steganography) AND (randomization OR chaotic) AND (spatial domain).

- ((chaotic map) AND (image steganography) AND security)) OR (randomization AND (image steganography)) AND (spatial domain).
- (((edge detection) OR (region based)) AND ((LSB image steganography) OR (LSB image steganography))) AND (spatial domain).
- (cryptography AND (image steganography)) OR (encryption AND (image steganography)) AND (spatial domain).

D. Data Selection

We have applied three filtering steps to select the relevant studies from the search results based on our keywords. The first step involves specifying the criteria for selecting the studies. Next, in the second step, the titles and abstracts of the studies are reviewed according to the research question in the second step. The third step involves reading the full texts of the selected studies and extracting the data. The following criteria have been mostly used for data selection:

- Has the paper been published in the last five years or so?
- Does the research article mention or discuss any of the security concepts in the image steganography field?
- Has the research article been included in any of the reference data sources?

E. Data Extraction

We have examined each preliminary study to identify if it was related to the security of image steganography. We have found about 220 papers upon concluding the search in February 2022. We have selected the relevant ones based on the criteria mentioned earlier. Finally, 100 related studies have been identified.

V. Security Techniques in Image Steganography

One of the most challenging steganography aspects is the security attribute. Here, security refers to the process of making the hidden secret information undetectable or the embedded size and locations unguessable. Much research has been conducted in this domain, which employs different approaches to improve steganographic security. Fig. 4 depicts a summary of the findings related to securing image steganography, and Fig. 5 classifies research articles accordingly.

Encryption is a concept widely utilized to achieve security; it encrypts the secret information before embedding it. Also, the entire image may be encrypted as an intermediate step in an algorithm. Traditional encryption algorithms such as Rivest, Shamir, and Adleman (RSA), Advanced Encryption Standard AES, and Triple Data Encryption Standard (3DES) are utilized to achieve this goal. User-defined techniques are employed as well. Randomization is another approach towards attaining security, which embeds secret information in a randomized way by scattering it over the original image. As part of this concept, chaotic function, pseudorandom number generator, user-defined keys, and other techniques can be exploited. In addition, randomization can be used alone or combined with encryption techniques to add an extra layer of security. Chaotic functions are famous for their random behaviour, wherein the outputs are unpredictable for certain input parameter values. The output is then exploited in the process of pixel location selection. Another technique to hide secret information and enhance security is transforming the cover image into another bit plane, embedding, and then retransforming the cover image to the original form. The region-based concept also improves the overall security, since it breaks the adjacent pixel correlations. In addition, some other techniques using different approaches are implemented. In the following section, more details about the above-mentioned categories have been presented.

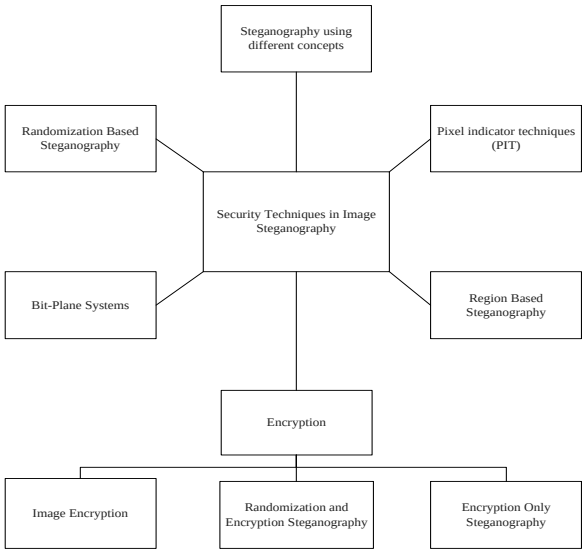

Fig. 4. Summary of findings.

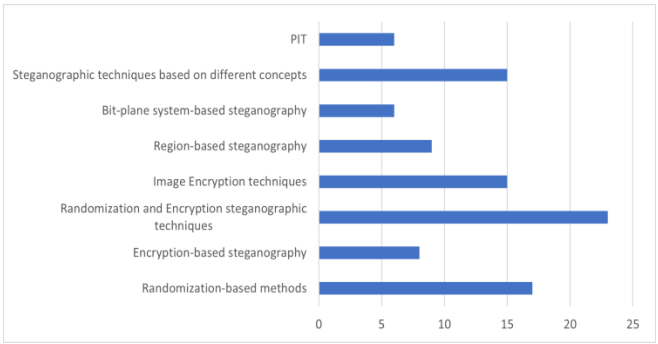

Fig. 5. Classification of the existing methods based on security technique.

A. Randomization based Techniques

These techniques rely merely on randomization to achieve security. In [30], random pixel positions are picked using Linear Congruential Generator (LCG). Eight bits from the secret message are embedded using the LSB technique with a sequence of 3-3-2, which means embedding three bits in the red channel, three bits in the green channel, and two bits in the blue channel. Authors in [31] suggest hiding three binary images within a grayscale image. Binary pixel values are rearranged using the mean of three random series and are hidden in a grayscale image employing the concept of Ultra Unique Numbers (UUN). Random numbers are used for pixel selection by generating multiple series in [32]. Similarly, [33]

uses Pseudo-Random Number Generator (PRNG) to choose a pixel for the embedding process. Three PRNGs are generated with the help of the Skew Tent Map (SKTM) in [34]. They are employed to scramble secret messages before insertion, choose an embedding color channel, and select a pixel location with a particular Red-Green-Blue (RGB) channel.

Authors in [35] proposed an improved 1D chaotic system model for choosing an embedding pixel location. The chaotic system utilizes a chaotic Logistic Map (LM) and a sine map function. A secret message is embedded at random positions using the Beta Chaotic Map (CM) in [36]. The algorithm in [37] selects random bits for embedding based on the use of PRNG. The key seed value and number of embedded bits are user-defined. Randomization of pixels is done via random chain codes of the key determined by the user in [38]. These chains contain a random sequence of bytes based on the hexadecimal representation of the bytes in the current key block. This sequence is used to embed the bits of secret message within the LSB of the pixels in the sub cover image. The authors of [39] suggest using two secret keys to randomize one bit of secret message: K1 chooses a channel, whereas K2 places the secret bit within a pixel. The Knight's Tour (KT) algorithm using the LSB technique is suggested by [40]. Here, the cover image is considered to be the surface of a chessboard, wherein the knight travels once to each square. In [41], the secret message is embedded in un-patterned fashion based on the outcomes of quadratic equations. In addition, combinations of RGB channels and image partitions are produced, wherein the Hungarian algorithm is employed to choose the least noisy combination, that is, the partitions of the cover image and the secret message.

In [42], standard deviation is utilized to select a richly-textured block of pixels to hold the secret message. Next, four Most Significant Bits (MSBs) of three diagonal pixels in the treated block are selected using SKTM to generate three correcting bits [Hamming code H (7,4)]. Two of these bits are XORed with the two secret bits and embedded in the neighboring pixels. In [43], the embedding position is located by a key generated utilizing a chaotic LM. Next, these located positions are further optimized by using two approaches. In the first approach, Arnold's Cat Map (ACM) is applied to add more randomness by shuffling the pixels. In the second approach, LM parameters are adjusted using the Genetic and Bat algorithms to find the best key value for the LM, leading to minimal changes. In the scheme proposed in [44], two chaotic maps are considered for pixel selection purposes: 1-D (Tent map) and 2-D (Baker's map). The embedding is done within the red channel. To minimize distortions, pixels on the edges are avoided, while the embedding process employs one bit or two bits of LSB. In the method proposed in [45], the cover image is scrambled using Power Modulus Scrambling (PMS) with the aid of the Brownian motion concept. Lighter pixels or pixels with less intensity exhibit more randomness than the heavier, darker pixels. The embedding process is carried out considering certain conditional factors. Finally, the reverse Brownian-based scrambling is applied to generate the stego image.

A keyed PRNG is used to scramble pixels, in order to achieve random permutation, in [46]. The permutation order is first based on an 8×8 Sudoku puzzle. Next, the embedding is done using LSB Matching, wherein two bits are embedded in the red channel, one bit in the green channel, and three bits in the blue channel. The Cross-coupled chaotic system (using two chaotic LM) is utilized in [47] to generate a DNA sequence, which is then added to the secret message's DNA sequence using the ASCII format. The result is then embedded using the typical LSB method. Table I summarizes the references related to randomization-based methods.

TABLE I. SUMMARY OF THE MOST OF THE MENTIONED RANDOMIZATION-BASED TECHNIQUES

Ref	Features and Pros	Cons	PSNR (dB)	Evaluation metrics
[30]	• Cover image is 90 angular transformed • Security is achieved through pixel randomization using linear congruential generator • LSB embedding following 3R-3G-2B pattern	• Few experiments • Embedding does not take into account the image content • Not compared to similar techniques • No steganalysis experiments • Not robust against statistical and geometrical attacks	512 x 512 PSNR: R: 64.0484 G: 64.5621 B: 67.362	PSNR, MSE
[31]	• Able to hide three binary images in the cover image • Shuffling based security using three random number series with the help of mathematical relations called UUN • Simple implementations	• Embedding does not take into account the image content • No steganalysis experiments • Not robust against structural and geometrical attacks • Low PSNR	PSNR: 37.71: 40.46	PSNR, MSE, SSIM, histogram analysis
[32]	• Binary image is embedded in a grayscale image. • Randomization embedding using a random number series. • Multiple series and reoccurring numbers are eliminated.	• All image regions are considered • No steganalysis experiments • Not robust against structural and geometrical attacks	Different resolutions images Payloads: (10:100%) 7680: 76800 bits PSNR: 83.97: 63.45	PSNR, MSE, Bit error, histogram
[33]	• Embeds 8bits/pixel 3R-3G-2B • XORing a bit of 5 MSBs with a secret bit and embedded in the particular LSB. • Randomization based on PRNG to choose a pixel and one of 5 MSB • Simple yet efficient encryption using XOR operation • Simple implementations	• Security is about inability to retrieve the message • All image regions are involved • No steganalysis experiments • Not robust against structural and geometrical attacks •	512x512 RGB bmp images Payloads 100: 262144 bits. PSNR: 39.263: 73.798	PSNR, MSE

[34]	• SKTM to generate three PRNGs to: scramble secret messages, choose an embedding color channel, and select a pixel location. • 1 & 4 LSBs versions • Chaotic map exhibits good statistical features • To make the algorithm more robust against statistical attacks	• StegExpose detects approximately half of the embedded information truly. • Security is about the secret bits' locations. • Embedding does not take into account the image content • Computational complexity • Many parameter to generate pseudorandom sequences	RGB 256x256 1-LSB, C=809:9041 Bytes. PSNR=65.97 :55.49 4-LSB, C= 809:36,167 Bytes PSNR= 52.621:36.101	COR, HOM, CON, ENR and CoC. MSE, PSNR, MaxErr, L2RAT, ENT SSIM, MSSIM, FSIM steganalysis: StegoExpose tool
[35]	• Improved 1D chaotic behaviour (an improved LM and sine map) • Improved robustness against statistical attacks.	• Moderate PSNR • Histogram has many spikes • Not enough experiments against chi square test • Embedding does not take into account the image content • Parameters exchange overhead	Color images 128x192, 192x256 256x288, 256x384 Payload: 24,576: 98,304 B Avg PSNR= 38.209	PSNR, MSE, Image Fidelity, Entropy
[36]	• Using Beta chaotic map • Good Visual quality	• Clear Histogram deformity • Complexity of the Beta map • Embedding does not take into account the image content • Key exchange overhead	USC-SIPI Image Database. PSNR: 57.5 – 56.79	PSNR, MSE
[37]	• PRN generator to define embedding locations. • user-defined Key seed value & number of embedded bits. • Not complex • Variable embedding capacity	• Embedding does not take into account the image content • Key exchange overhead • Few tests • Non-standard image • No steganalysis experiments	Message length: 343: 8866 characters Avg PSNR= 68.49	PSRN
[38]	• Uses chains of a random sequence of indices (codes) of the bytes in the carrier image. • Use of the full capacity of the cover image. • Robustness, and undetectability have been improved through extracting chains of randomly selected pixels from the cover image based on a user key	• Not compared to rival techniques. • Uses non-standard images. • Uses relatively large Stego secret key • No steganalysis experiments	Image size= 147456 Payload=18432 Bytes Image size=111156 Payload= 13894 Bytes PSNR avg = 51.31	PSRN
[40]	• Security achieved through encryption of secret message and randomization using KT algorithm (self-developed algorithm) • Performance against Chi-square is improved when using KT	• Not compared to rival techniques • Image content not considered • No numerical results • Stego-key value exchange overhead	Greyscale 512x512 from USC-SIPI	
[41]	• Finds message & cover image combinations with minimum changes • Randomization through using an un-patterned quadratic embedding sequence with unbounded i/p parameters. • An artificially created assignment problem with an optimized solution of the Hungarian algorithm	• Security is related to the embedding locations • Image content not considered • Stego-key value exchange overhead • No steganalysis experiments • Complexity high	Color image: 256x384 Payload= 23KB. Avg PSNR =52.739	PSNR
[43]	• Randomization of embedding location with optimizations using Chaotic LM to achieve highest PSNR possible • Uses LSB replacement and LSB matching • Optimization using Arnold's Cat map and adjustment of LM parameters using Genetic and Bat algorithm	• High complexity • No steganalysis tests • Key exchange overhead • Image structure not considered	256x256 grayscale. Embedding rate: 20:100% 2LSB: PSNR Org= 47.52 Optimized=48.44 SM2LSB PSNR, Org= 44.53 Optimized=45.22	PSNR
[45]	• Randomization key generated using Brownian motion • Nonlinear Brownian motion adds more security • High capacity	• Image contents not considered • Complex • Payload size not mentioned in the steganalysis • Key exchange overhead	128 × 128, 256 × 256, 512 × 512 Avg. PSNR= 48.467 without Brownian Avg. PSNR with Brownian = 48.45417	MSE, PSNR, SSIM entropy, Laplacian MSE Error (LMSE), Mean, S. deviation, Kurtosis, Skewness, KL Divergence, and NCC
[47]	• Randomization is achieved by combining two chaotic LMs to generate PRNG. • LM is utilized to generate a DNA sequence • The sequence is added to the secret message's DNA sequence using the ASCII format. • The result is LSB embedded.	• No tabulated experiments • Image regions not considered • Key exchange overhead	Payload: 37-character (296 bits) PSNR 99 dB	PSNR, Coefficient Correlation Test, Entropy

B. Encryption

In the following sections, steganographic techniques will be presented based on cryptography only or combined with randomization. In addition, some image encryption techniques will be presented, which can be utilized to encrypt the secret image and be used during the steganographic process.

1) Encryption based steganography: The technique in [48] proposes to encrypt the secret message using an XOR operation with a user-defined key. Next, a 4-bit shifting operation is applied to the encrypted text message in order to form the secret message to be hidden. The secret data is eventually embedded within the cover image using the standard LSB method. As per the technique suggested in [49], the letters of the secret message are initially transposed as an encryption process. Next, the secret message bit is XORed with the MSB of the image pixel based on a particular key hidden in the LSB of the cover image. In the method presented by [50], data hiding is performed using the typical LSB concept. The secret message is encrypted using the AES encryption algorithm with a 128-bit key in Cipher-Block Chaining (CBC) mode. Next, the order of pixel selection for embedding is obtained by utilizing a combination of the image attributes such as type and image resolution. The secret message in [51] is encrypted using XOR operation through a key generated via a circular bit shift operation having varying block sizes. Next, the encrypted message is embedded in RGB channels using the LSB algorithm in various ways for each channel. This process uses a 2-3-4 paradigm, wherein the insertion is done sequentially for the red channel, using raster scan pattern from right-left for the green channel, and using top-bottom raster scan for the blue channel.

The input RGB image in [52] is scrambled using a hyper-chaotic map to produce a permuted encrypted version of the cover image. The encrypted image is converted to YCBCR color space, and the luminance channel (Y) is then divided into 8×8 non-overlapping blocks to apply DCT and quantization on each block. Finally, Huffman coding is applied to the secret message and embedded in the left MSB. In the technique suggested by [53], the secret message is encrypted using a symmetric key cipher. The encrypted message is XORed with selected bits from the cover image for obtaining a higher-order pixel to add more confusion to the stego image. A block-wise inversion technique is applied to minimize changes during embedding by inserting them to an LSB or inverting them. The stego key consists of a symmetric encryption key and the encoding key, which contains parameter settings such as the number of blocks, starting block, start pixel offset, and block selection rule. In [54], the author introduces a secure steganography scheme, which encrypts the secret information using the permutations concept. Two chaotic map functions are utilized to obtain a sequence that is XORed with secret bits. The cover image is JPEG compressed, and DCT coefficients are modified and adaptively selected to hide the secret bits using a histogram modification-based data hiding scheme. In [55], the cover image is flipped by 180°. Next, the blue channel is divided into four sub-blocks, each of which is shuffled. The difference between the red pixels and the secret message is calculated and encrypted using a Multi-Level Encryption Algorithm (MLEA), which includes XOR, permutation, and shifting operations. The LSB embedding is done within the blue shuffled pixels. Subsequently, the sub-images are reshuffled and eventually combined using the red and green channels and re-flipped. Table II summarizes the encryption-based method in image steganography.

2) Randomization and encryption based steganographic techniques: The study [56] presents a scheme to improve the security of LSB steganography. In this method, the secret message is encrypted using a One-Time Pad (OTP) encryption. Subsequently, randomization is achieved by columnar transposition and RGB color plane scattering technique. Also, the technique in [13] uses OTP to encrypt the secret message, but it employs PRNG to select a pixel location for LSB embedding. In [57], the AES algorithm is applied to encrypt the secret message, which is then embedded using the LSB technique at a location randomly determined by the LCG method. The AES algorithm is also used in [58] to encrypt the secret message before dividing it into blocks. The cover image is segmented using a technique called a Non-Uniform Block Adaptive Segmentation on Image (NUBASI) algorithm. Finally, PRNG is used to randomly choose a message block to be embedded into an image segment. In fact, there are 32 predefined pattern orders of segments that can be selected at random. In [59], the secret message is encrypted and compressed by employing Vigenère Cipher and Huffman Coding, respectively. Next, the image is segmented into blocks in order to apply the KT algorithm to make groups of blocks. An arbitrary function is employed to select which blocks and groups can be used to conceal a specific pixel in the group randomly. Authors of [60] suggest encoding the secret message using bitwise XOR operations between adjacent pixels. Next, a local user selection is applied to find a particular position among the four least significant bits to hide a secret bit. The technique presented in [61] is based on Modified Least Significant Bit (MDLSB) to embed data in the cover image. It uses two layers of randomization: segment selection and pixel selection based on a user key. Further, a lossless compression algorithm, the DEFLATE algorithm, is applied to overcome the issue of embedding size in the subsequent layer. In addition, the AES encryption algorithm is utilized to encrypt the secret message in the next layer. The Artificial Bee Colony (ABC) algorithm is employed to reduce the noise caused by embedded information.

A randomized key based technique is proposed in [62]. First, the cover image is compressed, and then a secret key with the same length as the secret message is generated. Next, the secret message bits are XORed using the generated key. The resultant bits are embedded in the locations specified by the key. Finally, the image is encrypted using the 3DES algorithm. Authors in [63] proposed a secure digital image steganography scheme. In this approach, the secret message is initially compressed using the Run Length Encoding (RLE) algorithm, which is then encrypted using the Bernoulli map and the inverting bit map technique. During the embedding stage, the embedding location is selected by using KT algorithm and Henon Map (HM) function. The cover image is

divided into 8x8 blocks in order to apply the KT for selecting a block. HM function is employed to choose a pixel location among the 64 pixels in a particular block. The selected pixel is decomposed through Fibonacci, so as to embed the secret bit using a New Stego Key Adaptive LSB (NSKA-LSB), which maintains good imperceptibility.

In the technique proposed by [64], the location and the order of the image pixels are randomly chosen for embedding using a chaotic PRNG. The Pseudo-Random Number (PRN) is generated by exploiting the Duffing map and Circle map functions. In addition, the secret message is encrypted by XORing it with PRN before initiating the embedding process. In [65], the authors present a secure method that starts with encryption of the secret message using the RSA algorithm in combination with the Diffie-Hellman (DH) key exchange algorithm. Next, the encrypted message is compressed using RLE. Finally, via the Direct Sequence Spread Spectrum (DSSS) technique, PRN is used to select random pixels and to embed the secret message through 1-bit LSB and 2-bit LSB. The research [66] suggests a new method to enhance steganographic security. In this technique, the secret message written in Turkish text is encoded and compressed using the Huffman coding. The secret message is encrypted using the XOR operation and an OTP key with an equal-length payload. The key is generated using the super Mandelbrot sets and with the help of the LM. Subsequently, the LSB plane is analyzed morphologically to avoid low entropy pixel locations. Finally, LSB hiding is applied with the help of the chaotic LM in order to randomly choose a pixel location.

TABLE II. SUMMARY OF ENCRYPTION-BASED STEGANOGRAPHY TECHNIQUES

Ref	Features and Pros	Cons	PSNR (dB)	Evaluation metrics
[48]	• XOR Encryption with bit-shifting of the secret message. • Three RGB pixels used to hide an 8-bit data • LSB based • Simple encryption operation	• Image structure not considered • Steganalysis evaluation is missing • Not robust against statistical attacks • Same channel sequence followed	RGB 512x512 87373 bytes Avg PSNR= 51.637	PSNR, SSIM
[49]	• Transposition of the secret message and XOR with MSB of the pixels then embedded in LSB • Less computation and time complexities comparing to standard encryption techniques	• Steganalysis evaluation is missing • Secret key exchange • Not robust against statistical attacks • Limited capacity	256x256 grayscale image Payload: 1: 4 KB. PSNR= 63.236 : 57.132	MSE, PSNR, Histogram
[50]	• AES-CBC Encryption of the secret message • Embedding order of RGB depend on: file type & resolution • LSB based	• Steganalysis evaluation is missing • Not robust against statistical attacks • Limited capacity • Secret key exchange overhead	512x512 color images bpp=1 Avg. PSNR=51.196	RMSE, PSNR, MSE
[51]	• XOR encryption of the secret message with a circular bit shift generated key • varying block size • LSB embedding in the 2-2-4 LSB's of the RGB channels • Less complexity due to use of XOR encryption	• All image regions considered • Steganalysis evaluation is missing • Secret key exchange • Not robust against statistical and geometrical attacks	128×128, 512×512, 800×600 Payload: 16 KB: 480 KB bpp = 8 Avg. PSNR =64.85	MSE, PSNR, Entropy
[52]	• Encryption by shuffling image rows & columns with a help of hyper chaotic map • The Y channel of YCbCr version is DCT quantized • Embedding the Huffman of secret message into the MSB • Huffman coding increases capacity & security. • Robust against geometric attacks such as cropping attack, rotation attack, scaling attack	• Steganalysis evaluation is missing • Parameters exchange overhead • Tested against low payloads • Computation complexity	512×512 & 256×256 standard color 256x256 medical images Payloads: 100: 2050 Bytes PSNR=77.16: 53.68	PSNR, MSE, BER, SSIM, correlation, Quality Score prediction, Mean value, Standard Deviation, Entropy, Gradient
[53]	• Encryption followed by XOR encoding of the message with randomly selected higher-order pixel. • Resultant bit alterations minimized using a block-wise inversion. • Robust against chi-square, RS analysis, and sample pair test for the majority of stego images	• Use public key to exchange stego-key • Many parameters • All image contents considered • Clear Histogram spikes • Low payloads	256×256 Greyscale bpp=0.25: 0.9 PSNR: 57.475:51.629	MSE, PSNR, NCC, SSIM, Histogram based analysis (PDH), chi-square, RS groups analysis, sample pair test
[54]	• Hyper-chaotic system to permute the secret message and XORing it with the chaotic sequence • DCT coefficients modified using histogram modification-based data hiding scheme for high capacity • Low distortion	• Steganalysis evaluation is missing • Parameters exchange overhead • Computation complexity • Moderate embedding rate	USC-SIPI&UCID database 512x512 bpp = 0.2671 PSNR= 36.05: 38.93 bpp=0.08:0.18 PSNR=38.322: 41.695	PSNR, MSE, SSIM, average embedding rate ER, Information entropy, Correlation coefficients
[55]	• Flipping of the cover image and blue channel is divided and shuffled. • Multi-level encryption algorithm is applied to encrypt intermediate values • The result is embedded in LSB of the blue, reshuffle, and combined with red & green.	• Image structure not considered • Steganalysis evaluation is missing • Parameters exchange overhead • Computation complexity •	512x512 color images from USC-SIPI Payloads: 2:8 KB Avg PSNR= 65.9225 For 256×256 Payloads: 2:8 KB Avg PSNR= 71.0515	MSE, RMSE, PSNR, MAE, NCC and SSIM

In the scheme proposed by [67], the Bernoulli chaotic map function is utilized in three stages. It begins with encryption of the secret message by XORing it using a random sequence generated by the chaotic function. Then, one of the RGB channels is selected randomly using the random sequence. In addition, the secret message bits are randomly embedded in the rows and columns of the cover image. Embedding is done in the fourth least significant bit of the pixel. In [68], the secret data is transformed into binary form and then its CRC-32 checksum is generated. The data and its CRC are Gzip-compressed, followed by the AES encryption that is eventually appended to the header information. The last phase consists of information insertion using the Fisher-Yates Shuffle algorithm. This insertion enables selection of a pixel location for the embedding process, wherein all LSB channels are employed.

A method is suggested in [69] which starts by finding the best place to hide secret data using the LSB technique. Then, the secret information is encrypted using the 3DES algorithm; here, the RSA algorithm is used for the secret key exchange. The last stage is randomization, which is realized using a built-in randomization function based on the modulo Operation (mod) function with a secret seed. The secret seed is exchanged using the RSA. In the three-layered security suggested by [70], a small size fragile watermark is appended to the secret information to make it tamper-proof. The data is scrambled before the embedding process and partitioned into three vectors of varying length. The largest vector is embedded in the lowest order bit plane to minimize distortion that may arise after embedding. Before insertion, the host image is encrypted using the scrambling notion based on a Pseudo-Random Address Vector generated for image encryption, which is called PAVE. The concept of Pseudo Noise (PN) sequence generator is employed to generate the PAVE. The encrypted image is then partitioned. The embedding location is selected randomly by generating a random vector called Pseudo-Random Address Vector for Data Hiding (PAVH). Once the embedding is accomplished, the host image is decrypted to obtain the final stego image.

In [71], the first step towards achieving security comprises encryption of the secret image using RSA, which produces a 16-bit pixel cipher. The resultant cipher is rearranged to obtain a binary image, which is subsequently scrambled using a randomization concept relying on 2-D ACM transformation. The secret image is then embedded within the cover image using an inverted 2-bit LSB steganography wherein two bits are embedded. Bit inversion is based on the 3rd and 4th bits that are utilized as quantifiers. In [72], a chaotic LM and a support image are combined to generate a random binary sequence that is XORed with the secret message to get an encrypted version of the message. LSB embedding is done at positions determined randomly through random sequencing. The support image is pre-processed and utilized during the embedding stages to help resist steganalysis. In the method proposed by [73], the sequential color cycle is combined with pattern-based image steganography. The data to be hidden is encrypted using the AES algorithm. The cover image is divided into blocks and sub-blocks. Finally, a sub-block is chosen based on a predefined bow-tie shape pattern. The embedding operation is accomplished sequentially for the LSB of the RGB channels.

An algorithm to enhance security using randomization and multiple encryptions of secret images is presented in [74]. This technique embeds three secret binary images inside an RGB cover image. First, the three-color planes are separated. The red channel matrix is further separated into even and odd matrices. Next, the bits of the first secret image are embedded in the LSB of the odd matrix and the second LSB of the even matrix containing red pixels. Further, the bits of the second and third secret images are encrypted by XORing them with the first secret image. Finally, the two encrypted images are inserted in the LSB of the green and the blue channels, respectively. A pixel locator sequence-based technique was suggested by [75] to enhance LSB steganography security. This scheme starts by enciphering the secret information using the AES algorithm. The encrypted information is randomly embedded within the LSB bits of the image pixels with the help of a pixel locator sequence. Modern Fisher-Yates shuffle is utilized to generate the random sequence. The pixel locator sequence is also encrypted and appended to the image to form the final version of the stego image.

To secure communication, [76] incorporates randomization in combination with encryption. Confidential data is first encrypted using the Blowfish algorithm. Next, the LCG algorithm is used to generate random numbers that are in turn used to select pixels for LSB embedding. The number of bits to be inserted varies depending upon the pixel's intensity. Chaos based steganography is presented in [77]. In the first step, the secret information is encrypted using the AES algorithm. Following this, chaotic LM and ACM are employed to generate random values. These values are used to choose the positions of pixels on the cover image within which the secret bits are embedded. The first and second bit planes are used to uniformly distribute bits of the message. Three variations of random values can be constructed using the two aforementioned chaotic maps. Table III summarizes the references related to randomization and encryption-based steganography methods.

Image encryption: Various image encryption techniques are explained in this section. In [78], image encryption based on the confusion process is followed. The RGB image is transformed into a square grayscale image in the first step. Next, ACM is applied to scramble image rows and columns, followed by the HM algorithm for further scrambling to obtain the final cipher image. Authors of [79] suggest enciphering digital images by incorporating the confusion and diffusion concepts. First, the 256x256 RGB image is divided into four quadrants. Each of these quadrants is subsequently divided into four sub-quadrants that are rotated 90^0 anti-clockwise to form 64 sub-blocks in total. Then, modified zigzag transformation is applied to each channel to break the association with the adjacent pixels. Up to this step, confusion is fulfilled. An Enhanced Logistic Map (ELM) is used to generate intermediate encryption keys that pick specific pixel values to guarantee diffusion. The final key 'K' is generated based on the chosen values from the image and external user key, which are then XORed with RGB channels produced earlier after the zigzag step.

TABLE III. SUMMARY OF RANDOMIZATION AND ENCRYPTION BASED STEGANOGRAPHY TECHNIQUES

Ref	Features and Pros	Cons	PSNR (dB)	Evaluations Metrics
[13]	• Simple encryption and reduced computation using XOR encryption • OTP is randomly generated • PRNG used to randomize the encrypted secret message into the cover image • Simple implementation using LSB Substitution-based	• Steganalysis evaluation is missing • Image structure not considered • Image size not mentioned and payload • Not robust against statistical steganalysis • Channels used in the same order • Parameters exchange overhead	RGB images bpp=3 Avg PSNR= 83.27	MSE, PSNR, SSIM
[56]	• Simple encryption using OTP encryption of the secret message • Randomization is achieved by columnar transposition and RGB color plane scattering technique. • Simple implementation • Reduced computation and time complexities compared to conventional encryption	• Steganalysis evaluation is missing • Image structure not considered • Image size not mentioned and payload	RGB images Avg PSNR=58.47	MSE, PSNR, SSIM
[57]	• The secret message is AES encrypted • Pixels are randomly chosen using LCG method • Standard LSB embedding • All RGB LSBs are utilized	• Key exchange overhead • AES complexity • Steganalysis evaluation is missing • Image structure not considered	RGB images. Size of: 607x695 - 1,293×1,480 Payload = 8,230: 76,700 bytes PSNR=59.29:71	MSE, PSNR
[58]	• Three-layer securities through Encryption & randomization • AES and PRNG are used for message encryption and message block selection • Uses NUBASI algorithm for the cover image	• Secret key exchange overhead • Complex • Steganalysis evaluation is missing • Image structure not considered	512x512 RGB Payload= 2.5 KB Avg. PSNR = 63.755	PSNR
[59]	• Strength against electronic attacks by encryption, compressions, and randomization. • Randomization using KT algorithm and arbitrary function • Robust against Chi-square attack • Exploiting Modification Direction embedding based technique	• Encryption key, arbitrary function key, and Huffman table exchange overhead • Complex • Payload in the steganalysis not mentioned • Not tested against other attacks • Image structure not considered	512x512 images from USC-SIPI bpp = 0.5: 1.6 PSNR = 60.84: 55.69	PSNR, MSE and SSIM
[60]	• XOR encoding of the secret message • User selection to hide a secret bit in one of the 4 LSBs • Modified LSB • Simple implementation	• Multiple cover images possibly needed • Steganalysis evaluation is missing • Image structure not considered • Not compared with similar techniques	256x256 grey scale payload: 128x128 image Avg. PSNR = 43.455	MSE, PSNR
[61]	• To enhance security, use of multi-stage randomization, a lossless compression algorithm (DEFLATE algorithm), and AES encryption • ABC algorithm to reduce embedding noise • Tested using ROC curves of the WFLogSv steganalyser	• AES key exchange overhead • Not simple • Time Complexity • Image structure not considered • Other Steganalysis techniques missing	Images from UCID database 150×150 - 1080×1024 PSNR without ABC= 48.1:58.2 with ABC: enhance by magnitude of 3:6	PSNR, SSIM Euclidean norm testing ROC curves of the WFLogSv steganalyser
[62]	• Security is achieved by Compression, randomization, and encryption • The image is compressed and XORed with a generated key • A key based randomization to hide the secret message • Encryption of the image using 3DES	• Steganalysis evaluation is missing • Image structure not considered • Few tests • Not compared with similar techniques • Not typical steganography, encryption alike	Color images: 256x256, 250x360 pixels Avg PSNR: 53.51	MSE, PSNR, and NC
[63]	• Utilizes compression, encryption (Bernoulli map), randomization using KT and HM. • Fibonacci decomposition with the help of NSKA-LSB • Maintains good imperceptibility.	• Complex due to many stages • Steganalysis evaluation is missing • Image structure not considered • Few tests • Not compared with similar techniques	512 x 512 gray images From USC-SIPI Embedding rate = 6.25, 12.5, 18.75 %, Avg, PSNR= 61.14, 66.58, 72.29	PSNR, SSIM, and BER

[64]	• Randomization with help of Duffing map and Circle map • XOR Encryption of the secret message • Chi Square test passed with payload of 4000 bits • Good imperceptibility • LSB replacement based	• Only Chi-square steganalysis is used for evaluation with low payload • Image structure not considered • Key exchange overhead	Non-standard color images: 256×256, 512×512 Payload: 800:16,000 bits PSNR: 256x256 = 63.0: 76.56 512x512 = 69.1: 82.73	Randomness tests, MSE, PSNR and SSIM The Average Difference, Laplacian MSE, NAE, IQI
[65]	• Secret message encryption and compression • Randomization to select random pixels to embed the secret message • Direct sequence spread spectrum technique is used in LSB-1 and LSB-2	• Steganalysis evaluation is missing • Image structure not considered • Few tests • Not compared with similar techniques • Diffie-Hellman to exchange keys	jpg images payload: 50: 1000 characters PSNR 58.528: 71.596	PSNR, MSE
[66]	• Regional adaptive with randomization embedding. • Huffman compression and OTP-XOR encryption of Turkish secret message • OTP generated using Mandelbrot sets with the LM. • Only high entropy regions are considered	• Only Chi-square steganalysis is used for evaluation • Huffman table and parameter exchange overhead • Relatively complicated • Only LSB plane is analysed and considered	NASA&SIPI grayscale images 512x512, Payload: 1: 93.5 KB PSNR= 61.9: 51.15	PSNR, BPP, SSIM UNIVERSAL IMAGE QUALITY INDEX (UIQI)
[67]	• XOR encryption of the secret message with a random sequence of Bernoulli chaotic map function • Randomization of channel and pixel selection • Embedding is done in one of the fourth LSB of the pixel.	• Steganalysis evaluation is missing • Not robust due to Image structure not considered • Low payloads used	RGB: 256x256, 720×576 Payload: 44: 500 characters PSNR: 52.12: 67.82	Statistical Metrics. MSE, PSNR, Max Absolute Squared Deviation, Ratio of the squared norm and CC
[68]	• First security tier is the secret message compression and AES encryption • Pixels randomization using Fisher-Yates Shuffle algorithm • Resistant to the Chi-square • Integrity check guaranteed using message CRC • High capacity with good imperceptibility	• Only used Chi-square steganalysis • Image structure not considered • Key exchange overhead • Relatively complicated	512x512 RGB Payload: 1 : 256 KB PSNR: 40.083: 63.862 Max payload 315392 Bytes (Gzip)	MSE, PSNR, NCC, Average Difference, Maximum Difference, Laplacian MSE and Normalized Absolute Error
[69]	• 3DES encryption of the secret message, • Randomization using mod function with secret seed. • RSA for key exchange. • LSB embedding	• Key exchange overhead • Image structure not considered • Weak randomization as its not chaotic • Steganalysis evaluation is missing	Payload: 12: 48 KB PSNR 53.3150: 52.8734	PSNR, SSIM, retrieval bit, error rate (RBER)
[70]	• Pseudorandom vectors used for image encryption, scrambling and secret message encryption • Embedding is Randomly achieved at Intermediate Significant Bit (ISB) • High security due to encryption and randomizations	• Complex • Master key exchange overhead • Image structure not considered • Steganalysis evaluation is missing • Moderate PSNR	512×512 standard images bpp=2 Avg PSNR= 39.11	PSNR, MSE, Normalized Absolute Error (NAE) and NCC.
[71]	• RSA Encryption of the secret image • Randomization using 2-D ACM • Embedding using an inverted 2-bit LSB steganography • Embedded in one channel (blue) • High capacity	• Image structure not considered • Steganalysis evaluation is missing • Time complexity	512x512 images Secret messages: RGB 64x64 & Grayscale 128x128 Payload: 24,576:32,768 KB PSNR=57.25: 52.5	PSNR, MSE
[72]	• XOR encryption of secret information with help of LM. • Embedding LSB plane randomized • Support image processed for information extraction • Imperceptibility of the first image is ideal	• Image structure not considered • Steganalysis evaluation is missing • Extra overhead due to using two images • Key exchange overhead	128x128 pixels 1st Primary image= 50% data 2nd Support image=50% data PSNR of 1st image =inf PSNR of 2nd image= 53.24	PSNR, MSE
[73]	• AES encryption of the secret message • Randomization based on Bow-tie shape pattern • LSB Embedding using SCC • Stego image is further encrypted	• Image structure not considered • Steganalysis evaluation is missing • Key exchange overhead • Few tests	RGB 256x256, 512x512 PSNR= 39%, 49%	PSNR

[74]	• The secret images separated into multiple parts • Based on RGB planes with several combinations of separated red channel with multiple XOR encryption of the secret and the green and blue LSBs • Randomly embedding in LSB of the green and the blue channels	• High complexity due to using 3 secret images • Image structure not considered • Steganalysis evaluation is missing • Key exchange overhead • Few tests	Cover images jpg 255x255x3 Secret images: .png LSB & LSB1 PSNR: 49.248 LSB1&LSB2 PSNR:44.999	PSNR, MSE
[75]	• AES encryption of the message • Randomization of pixels using Modern Fisher-Yates Shuffle • Embedding using LSB substitution • Pixels sequence is encrypted and sent along with image	• Image structure not considered • Only tested against Chi-square steganalysis • Pixel Locator Sequence exchange overhead • Not compared to similar techniques	RGB images 225×225, 512×512 Avg. PSNR =46.148	PSNR, MSE, Chi square
[76]	• Encrypting the secret message using Blowfish algorithm • Randomization achieved by LCG algorithm • Number of embedded bits is intensity based (2-6 bits)	• Image structure not considered • Only used Chi-square steganalysis • Not compared to similar techniques • Payload capacity is not mentioned in Chi square attack.	256×256, USC-SIPI image database bpp: 2.00: 2.95 Avg PSNR = 47.286	MSE RMSE PSNR Chi-Square Attack
[77]	• The secret message is AES encrypted and the embedding positions are randomized using LM and Cat maps. • 1st & 2nd bit planes are utilized • Using accurate PSNR (weighted PSNR)	• Image structure not considered • Non-standard steganalysis techniques • Not compared to similar techniques • parameters exchange overhead.	Medical images 256 × 256, 512×512 Payload 0.5 bpp. Avg PSNR= 50.89: 50.73 Avg wPSNR= 60.21: 69.84	PSNR, wPSNR, MSE, SSIM, Cross-correlation Coefficient, Entropy and Key sensitivity.

In [80], a color image of H×W size is decomposed into three images consisting of its RGB components at the bit-level. These images are then vertically combined to create one bit-level image (3H rows x 8W columns). Based on Chaotic SKTM, permutation is carried out at a bit level of the image. The initial values of the chaotic function are concluded based on a 128-bit key value, which is constructed from a combination of external user key and information extracted from the plain image. Eventually, the function is iterated 3Hx8W times, and the results are recorded and inverted to be used for permuting the image bits accordingly. A lightweight encryption scheme is presented in [81]. The raw image is converted to a 1D array and permuted using a random sequence produced by the LM. This sequence is transformed into a DNA sequence. Also, the LM is used to generate another random sequence that is likewise used to obtain another DNA sequence. The two sequences are added using DNA computation rules. In the final step, each pixel is XORed with its preceding pixel. In [82], the authors propose a compression and encryption method based on hyper-chaotic function, 2D compressed sensing, and DNA encoding. The aim of using a hyper-chaotic function instead of the 1D chaotic function is to increase the system key space, thereby improving system complexity and security. The cover gray image is compressed using the 2D compressed sensing technique. Fractional-order Chen hyper-chaotic system is employed to control DNA encoding and operation. The initial values of the Fractional-Order hyper-chaotic system are generated using the hash value of the original image. The DNA encoded image is enciphered using Arnold transformation to produce the encrypted image.

In [83] authors propose an encryption scheme that uses intertwining and ELMs. The primary point is to de-associate neighboring pixels by shuffling them. Next, the image is partitioned into blocks, and permutation-substitution operations are performed using an intertwining LM and ELM along with the cosine-based transformation. In addition, the image is rotated 90° anticlockwise. Finally, random order substitution is achieved by utilizing the random sequence generated using a chaotic function. A novel high speed image encryption scheme based on 1-D cosine fractional chaotic map and image encryption scheme (DCF–IES) is proposed in [84], which is based on a new real 1-Dimensional Cosine Fractional (1-DCF) chaotic map. To increase encryption speed, permutation operation is excluded from the architecture design. Despite the absence of the permutation process, a substitution process with a high sensitivity to a plain image guarantees a high level of security. The chaotic function is first utilized to generate two sequences that act as system keys, wherein the keys are used to accomplish substitution. The process is achieved by performing an XOR operation as also addition and MOD functions between the key and raw pixels. This operation is iterated twice over image row values to ensure strong encryption. The output cipher is sensitive to minor changes in the value of the input pixels.

In [85], image scrambling-based encryption is presented. Scrambling is carried out based on a random sequence, which is generated using a formula of two inputs that act as a secret key. Then, the pixels' locations are rearranged utilizing the generated sequence. Permutation based encryption is suggested in [86]. In this method, the Linear Feedback Shift Register (LFSR) produces a PRN that is employed to generate two shuffled images. Permutation of the image rows creates the first one, while permutation of image columns forms the second image. Both images are XORed in the final step to create the encrypted image. The Chaotic Gravitational Search (CGS) concept for encryption is introduced in [87]. In this system, a binary gravitational search algorithm selects a random PRNG, thus starting an encryption key among three algorithms: Mersenne Twister (MT), SIMD-Oriented Fast Mersenne Twister (SFMT), and Combined Multiple Recursive (MRG) algorithms. Then, the Chaotic Gravitational Search Algorithm (CGSA) produces the initial value for the selected generator. The generated keystream is used to encrypt images by applying permutation alone or in combination with the substitution process.

Authors in [88] suggest transposition and substitution encryption-based techniques that utilize two pseudorandom generators: an altered version of the Sophie Germain Prime Generator (ASGPG) and Lehmer Random Number Generator (LRNG), which generate a group of random numbers. The first group is used to assign new values to image pixels by XORing the random numbers with image pixels: a substitution condition. The second group attains the transposition by swapping the positions of pixels. In [89], improved Baker map and LM are used to achieve image encryption. A two-dimensional Baker chaotic map is employed to control the chaotic LM parameters as well as the state variable. It is used to increase the randomness and unpredictability levels. Two random sequences are produced by the chaotic LM. The sequences are then exploited to perform image encryption by first shuffling the pixels, followed by substitution. Likewise, in[90] chaos-based encryption is proposed, which follows the permutation and substitution methodology to encrypt the original image. Clifford's chaotic system and LM are iterated for a particular value followed by a quantization process. Next, positions are shuffled, and grayscale substitution is performed as a final stage. Parameters of the Clifford system map or the attractor, act as encryption keys.

In [91], the authors propose a Hyper-Chaos (HC) based image encryption algorithm. They make use of a 5-D multi-wing hyper-chaotic system to resist cryptanalysis. It generates a chaotic sequence that is used first for pixel-level permutation and then for bit-level permutation. Up to this point, confusion is fulfilled. To achieve diffusion, the chaotic sequence generated earlier is XORed with the confusion step output. It is worth mentioning that the function's parameters are deduced from the plain image properties. In [92] presents a unified image encryption algorithm. This technique utilizes Substitution-Box (S-Box) to realize the diffusion concept. Hence, a keyed-piecewise linear chaotic map creates a key stream sequence. This key is employed for diffusion operations that are based on a specific formula between this key value and image pixels. However, the intermediate encrypted image is 180° rotated and then scrambled before making the second diffusion. In this technique, the decryption process is identical to the encryption process. Table IV summarizes the related references for the image encryption methods.

C. Region based Steganography

In the following paragraphs, some techniques will be presented, wherein steganography is carried out in certain areas of the cover image to primarily enhance security. Some techniques exploit only image edges, while others use edges and smooth areas. Image edges and boundaries are the areas in which the intensity value changes sharply within a short distance, while the smooth area is the area with low-intensity variations. In [18], edge pixels are detected and selected for embedding using the Canny edge detector. Huffman code is applied to the secret bits in the primary step for data compression. Then, the edge pixels are randomized, and the number of pixels intended for embedding is determined by coherent bit length L. Next, 2k correction is applied to achieve better imperceptibility. Edge detection incorporated with morphological dilation as part of the steganographic technique is suggested in [20]. In this method, the secret message is embedded in the image sharp regions, which are detected by the Canny edge operator and optimized by the morphological dilation operator. The Canny operator is applied to a modified version of the original image channels obtained by adding the 4 MSBs of RGB channels. Subsequently, a 3x3 dilation operator is utilized to identify reference pixels. The bits are inserted within the remaining LSB bits using the hybrid XOR technique, such that least possible alterations of edge pixels are guaranteed. This meets the high security demands.

TABLE IV. SUMMARY OF ENCRYPTIONS TECHNIQUES OF IMAGE ENCRYPTION TECHNIQUES

Ref	Encryption Concept
[78]	Confusion based encryption using Arnold Cat map and HM algorithm
[79]	Encryption is achieved by rotation, modified zigzag transformation, and enhanced LM output with XOR operation. It is a confusion and diffusion encryption.
[80]	Bit-level image Permutation using a chaotic skew tent function
[81]	Permutation using a chaotic LM in addition to DNA encoding operations
[82]	Arnold transformation encryption of DNA
[83]	Permutation-substitution encryption utilizing intertwining logistic, enhanced LM, and cosine-based transformation
[84]	Substitution by exploiting 1D-dimensional cosine fractional (1-DCF) chaotic map
[85]	Scrambling based encryption using random sequence based on user formula
[86]	Permutation based encryption using linear feedback shift registers along with XOR operation
[87]	Permutation only or with substitution by exploiting a Binary gravitational search algorithm. Mersenne twister, SIMD-oriented Fast Mersenne twister, and combined multiple recursive.
[88]	Transposition and substitution encryption based on Sophie Germain prime generator and Lehmer random number generator
[90]	Transposition and substitution encryption using an improved baker map and LM
[91]	Permutation - substitution encryption utilizing Clifford chaotic system and LM
[92]	Permutation - substitution encryption using a 5-D multi-wing hyper-chaotic system
[92]	Substitution - permutation encryption by employing a keyed-piecewise linear chaotic map

In [93], the secret data is hidden in the edges of the image using the LSB technique. The Canny edge detection algorithm is employed to identify image edges. The secret message is converted into a binary sequence and encrypted using the OTP encryption method. The encryption process is achieved by modulus addition of the secret bits with the corresponding OTP bits. Laplacian of Gaussian (LoG) detector is used in [94] alongside a chaotic function. The secret information is first encrypted by XORing it with a PRN generated by a chaotic LM. Then, the edges of the green and blue planes are identified using the LoG edge operator. Finally, 2-LSB of the green and blue edge planes are used for the embedding process. In [19], the author suggests using a hybrid edge detector and encryption to secure the steganographic technique. The Vernam cipher is applied to the secret message using a pseudo-random generated key that is generated using a nonlinear feedback shift register, Geffe Generator. This method uses a hybrid edge detector which combines the Sobel operator with Kirch operators. The LSB technique is used to hide secret message bits; here, three bits are embedded in the edge pixels while two bits are embedded in non-edge pixels.

In [95], an Adaptive Multi Bit-Planes image steganography using Block Data-Hiding (MPBDH) is proposed. First, the secret message is encrypted using the AES algorithm. Then, complex regions are chosen for embedding, based on a complexity threshold estimation and the number of bit planes. If the whole message cannot be embedded within the selected pixels, then parameter readjustment is used to compulsorily ensure that all bits are embedded. LSB and Hamming code-based algorithm is suggested in [96]. The cover image is divided into blocks, and then edge detection is performed after zeroing all LSBs of the red channel pixels. The embedding is done based on the pixel's location. If the pixel is a non-edge pixel, standard LSB embedding is used for the RGB channels. Otherwise, LSB is performed for the three channels and the other two LSBs of the RG channels are used to embed the Hamming code. In [97], the complexity is based on the pixel variation among the central pixel and all neighboring pixels. The cover image is divided into small overlapping blocks (3x3). Then, a multidirectional high pass filer bank is used to find eight residual responses, which are then utilized to calculate the corresponding complexity using a proposed Complex Block Prior (CBP) criterion. The blocks are also sorted from high to low complexity, and the blocks with the same complexity level are grouped together. Finally, a single bit or multiple bits are embedded adaptively within the central pixel.

The Block-Wise Edge Adaptive Steganography Scheme (BEASS) method is suggested in [98]. It utilizes the edges obtained by using a fuzzy edge detector as well as the surrounding pixels to embed secret message bits. The cover image is divided into 64x64 blocks, and their corresponding standard deviation is calculated to estimate their local complexity. The blocks are sorted according to their standard deviation, and then the blocks are further divided into 3x3 blocks. Three message bits are hidden in edge pixels using the minimal mean squared error (MSE) that helps determine the embedding capacity of neighboring non-edge pixels within the block. Table V summarizes the related references about the region-based image steganography methods.

TABLE V. SUMMARY OF REGION-BASED STEGANOGRAPHY TECHNIQUES

Ref	Features and Pros	Cons	PSNR (dB)	Evaluations Metrics
[18]	• Secret message compressed using Huffman coding and randomly embedded in edge pixels (Canny edges) • Coherent bit length L determines the number of embedding pixels • 2k correction maintains visual quality	• Huffman Table must be exchanged • Steganalysis evaluation is missing • Steps add complexity • Limited capacity	512×512 gray-scale Payloads =35852:108074 bits Avg. PSNR= 61.9654	PSNR, Universal Image Quality Index (Q)
[19]	• Vernam cipher encryption of the secret message with a pseudo-random key to enhance security • Efficient hybrid edge detector of Sobel and Kirch operators to improve capacity • LSB adaptive embedding in Green & Blue channels.	• 7 bits of 2 block of 3x3 pixels utilized as embedding indicator • Steganalysis evaluation is missing • Encoding complexity	90x90 secret image 512x512 cover image Avg bpp=1.9605 Avg PSNR= 41.533	MSE, PSNR, bpp
[20]	• Canny edge detector with dilation operator to involve edge pixels and their neighbours to improve capacity • Embedding using an improved XOR technique to improve security • Robust against (RS) steganalysis	• Relatively Low embedding rate • Encoding complexity. • Only RS steganalysis tested •	512x512 Greyscale and color images Avg bpp=1.25 Avg PSNR = 44	MSE, PSNR, SSIM, FSIM
[93]	• To add security, the secret message is OTP encrypted • OTP key is obtained using random function • LSB based in Canny edge pixels.	• Handling of message key and host image edges • Steganalysis evaluation is missing • Low capacity	512x512 greyscale images Payload: Up to 1KB PSNR for 1KB =69.1106	CC, PSNR, MSE
[94]	• LoG edge pixels of green and blue channels is used for embedding process • Efficient low complexity XOR encryption of the message with chaotic logistic map sequence	• Parameters exchange overhead • Steganalysis evaluation is missing • Relatively Low capacity as 2 channels are used	512x512 RGB image bpp=1.72 PSNR= 46.1733	PSNR, NCC, Entropy, Number-of-changes-per-rate, and Unified-average changed-intensity.

[95]	• The secret message is AES encrypted and adaptively embedded in noisy regions • Muti bit planes is utilized with block data-hiding • Enhanced image visual quality • Robust against visual attack and ensemble classifier	• RSA key exchange overhead • Other well-known steganalysis are not tested • Complexity is based on the bit planes not value of pixels • Ensemble classifier test for low bpp	512x512 gray images bpp=0.4, PSNR= 56.64, wPSNR= 71.96 bpp=1.50, PSNR=48.08 wPSNR=63.20	PSNR, wPSNR, SSIM, Visual attack, Ensemble classifier
[96]	• Adaptive embedding based on LSB and Hamming code. • Non-edge pixel use LSB embedding • Canny edge pixel to embed secret bits and Hamming code • Imperceptibility improved with adaptive embedding	• Steganalysis evaluation is missing • Few tests • Capacity affected by Hamming code • Cleared red LSBs affects the visual quality	RGB 512x512, 512x384 from USC-SIPI, UCID-Image Database Payload: 9424:30224bits PSNR: 63.397 :68.429	PSNR, MSE
[97]	• A content-adaptive steganography method based on identifying the local texture complexity. • Complexity based on pixel variation with respect to all neighbours using multi-directional High Pass Filter bank • The maximum of the pixel differences in a pixel block determines number of embedding bits • Embedding using LSBMR or multibit XOR	• Too local complexity estimation since small blocks size are utilized • Other steganalysis tests not tested • Relatively complex • Parameter handling	512x512 greyscale SIPI, BOWS2 and BOSSbase dataset Max capacity= 249,729: 549,051 (0.95: 2.09 bpp) PSNR= 44.16:56.23 wPSNR=63.88: 75.87	Capacity, PSNR, WPSNR and SSIM SPAM STEGANALYSIS
[98]	• Block-wise Edge Adaptive Steganography Scheme (BEASS) • Dynamic local complexity measure of Standard Deviation is utilized (image subblocks containing edges are selected) • high payload with minimal distortion embedding utilizing the minimal Mean Square Error • Robust against major attacks	• Standard deviation as a complex measure is not accurate for big blocks as it does not take into account the spatial arrangement of pixels • wPSNR measure is more accurate in such cases • Low embedding capacity • High complexity	512x512 greyscale images from BOSSbase database bpp= 0.3 : 1 PSNR (0.3 bpp) =70.54: 74.66 PSNR (~1 bpp) = 61.28: 65.78	PSNR, KL-Divergence, SSIM, Average Difference, NAE, Execution time, Kurtosis, Skewness, Histogram analysis, RS attacks, and feature based universal steganalyzer

D. Bit-Plane System

The following LSB steganography techniques use the concept of virtual bit-plane. Higher plane systems are exploited instead of using an 8-bit plane to hide secret data. In these systems, the Zeckendorf criterion needs to be satisfied in order to embed the secret information that can be extracted later. "Every positive integer can be represented uniquely as the sum of one or more non-consecutive distinct Fibonacci numbers" [99]. Hence, the number representation is considered valid if no consecutive ones appear in the sequence.

In the scheme presented in [99], secret data is embedded in different bit-planes rather than using the regular binary bit planes. It is based on the Lucas Number system that uses 11 numbers for representations. Hence, the Lucas sequence is utilized for image bit plane representations, which uses 11 bits instead of 8 bits for representing the pixel's intensity. Embedding is achieved in the second bit-plane, which yields deterioration by ±1 in the stego image. Blue and green channels of the RGB color image are used for data embedding, while red is used as an indicator. As mentioned, embedding should comply with the extended Zeckendorf theorem for handling redundant representation. In [100], the method represents the cover image using the Fibonacci sequence. The cover image in this situation is represented in 12-bit planes. The secret message to be embedded is encrypted using a symmetric cipher with the help of a chaotic LM to generate the encryption key. Then, the generated key is XORed with secret bits and embedded in the second LSB.

Authors in [101] suggest improving the Fibonacci data hiding technique by utilizing Catalan numbers. A certain set of Fibonacci numbers and a certain set of Catalan numbers are combined to create a new number set that complies with Zeckendorf's theorem. As a result, 15 virtual bit-planes are obtained, which is three more than the number of bit-planes produced by the Fibonacci method. Authors in [102] propose two embedding techniques by utilizing two different number systems. The first embedding scheme is based on a prime decomposition of pixel value into 15 virtual bit-planes. In contrast, the second is based on natural number decomposition, which yields 23 virtual bit-planes. In these techniques, the secret message can be embedded in higher bit-planes without introducing noticeable distortion. In [103], a new method based on a specific representation is used to decompose pixel intensity values into 16 virtual bit-planes. This scheme necessitates that the sum of all bit-planes must be less than the highest pixel intensity value, which in this case is 2^8-1. For the embedding process, a cover pixel is randomly selected using PRNG, then decomposed, and a particular virtual-bit plane is chosen. To represent numbers that have multiple representations, the one with high lexicographically representation is selected. In addition, if a pixel value post embedding cannot be represented in the system, it is excluded, as the embedded information cannot be extracted. The author in [104] proposes a steganographic technique based on a new pixel value decomposition. This scheme uses a set of decomposition weights that decompose the cover image into 10 bit-planes. The first five LSBs' weights increase slightly while growing exponentially for the rest. The secret message bits are embedded in one or more LSB positions. Embedding is only done for modified pixels with a valid representation in the system. Moreover, in several cases, numbers have more than one representation in the system. Hence, the one with the lowest lexicographical representation is a valid number. Table VI summarizes the references related to bit-plane based methods.

TABLE VI. SUMMARY OF BIT-PLANE BASED STEGANOGRAPHY TECHNIQUES

Ref	Features and Pros	Cons	PSNR (dB)	Evaluations Metrics
[99]	• Using Lucas sequence to represent pixel's intensity • Yields deterioration by ±1 in the stego image. • Reduces visual distortion effects. • Robust against geometrical, statistical and structural attacks	• Low capacity than standard LSB steganography • Produce lower quality stego image • Image structure not considered • More complex than binary representation	512x512, 394x600, 768x512, 200x200 RGB USC-SIPI & Kodak bpp=10%:100% PSNR=57.27:47:34	MSE, PSNR, SSIM histogram differences, image, Chi-square attack, structural attacks and RS attack
[100]	• Cover image represented using Fibonacci sequence • XOR encryption of LM sequence with secret message • Embedding in the 2nd bit plane • Simple and effective encryption using the XOR operation	• Some pixels escaped • Image structure not considered • Steganalysis attacks missing • Computation overhead • Parameters exchange overhead	512x512 greyscale images from CVG-UGR Database bpp=0.3, PSNR = 54.21 bpp=0.9, PSNR = 41.04 Avg PSNR :49.97	ER, MSE, PSNR Robustness tests Bit error rate BER rate-distortion curve
[101]	• Number set composed of union of a certain set of Fibonacci numbers and a certain set of Catalan numbers • 15 virtual bit-planes obtained • Robust against statistical and geometrical attacks	• Some pixels escaped (affects capacity) • Image structure not considered • Steganalysis attacks missing • Computation overhead	Greyscale images Payloads: 1000:2500 bits PSNR= 67.03: 71.45	MSE, PSNR
[102]	• Cover image can be represented using prime sequence (15 bits), or using natural number sequence (23 bits) • Secret message can be embedded in higher bit-planes without introducing distortion.	• Image structure not considered • Steganalysis attacks missing • Computation overhead • Low visual quality for higher bit planes	Greyscale images (Natural, prime) PSNR(bit plane): 0^{th} = 48, 48, 1^{st} = 43, 43, 7^{th} = 30, 24	Worst case Mean Square Error (WMS) PSNR, Histogram
[103]	• Pixels randomly selected and decomposed into 16 bit-planes. • A particular bit plane is chosen • Produces less distortion	• Some pixels cannot be used for embedding • Capacity is affected by unusable pixels • Image structure not considered • Steganalysis attacks missing	512 x 512 PSNR**:** 1^{st} LSB= 52, 2^{nd} LSB= 50, 7^{th} LSB= 37, 8^{th} LSB= 34	Payload capacity, PSNR
[104]	• Cover image is decomposed into 10 virtual bit-planes • The first 5 LSB weight increases slightly • Embedding in one or more LSBs positions (valid pixels) • Robust against statistical	• Not every pixel is valid after embedding • Capacity is affected by unusable pixels • Image structure not considered • Steganalysis attacks missing • Computation overhead	512x512 jpg images Embedding using: 0^{th},1^{st}, 2^{nd},3^{rd},4^{th} bit PSNR: 39.41: 49.02	MSE, PSNR, SSIM

E. Pixel Indicator Techniques

In indicator-based data embedding schemes, the channels are divided into indicator and data channels. The indicator channel determines the data channel for data hiding, based on certain sequences for better security. In [16], the two least significant bits of one channel are used as an indicator of the presence of secret data in the other two data channels. The indicator channel is chosen based a sequence created from R, G, and B, that is, RGB, RBG, GBR, GRB, BRG, and BGR. The length of the secret message is used as an indicator selection criterion to enhance security. In addition, hiding data in channels considers the pixel's intensity.

In [1], the 7^{th} bit of a pixel and the 7^{th} bit of the pixel value +1 are concluded. A comparison between those values and two bits from the secret message is conducted to seek a matching. In the case of a mismatch, the difference is calculated and then added to the pixel value. At this point, the embedding process is completed for this pixel. Eventually, the embedding process alters the pixel value by +2 or -2. The aim of designing this algorithm is to implement robust and secure steganography. In [25], a variant version of PIT is presented. This technique uses indicator channel criteria to determine the data channel order, based on the combinations of the MSB in the RGB channel. The first step is to obtain the message length, which is then stored in the first 8 bytes of the first row. Then, embedding is started from the second row, and the indicator channel selection criteria is utilized to determine the order of the data channel. The selection criteria are based on the message length and the type of the parity bit used to create a shuffled order of RGB channels. Also, the number of bits to be embedded in each channel is specified.

The study [105] suggests a PIT based technique to achieve high capacity. Here, the binary secret message is divided into four parts and the cover image. The two LSB of the first eight bytes of the red channel of the image are used to store the message length. The order of the cover image parts is stored in the two least significant bits of the four first pixels in the blue channel. Next, each pixel is evaluated for the ability to embed using the red channel's three MSBs. The three bits represent the RGB ability: the presence of zero means no embedding has taken place in that channel. Then, the number of zeros in the four MSBs of the candidate channel is counted to determine the number of secret bits to be hidden. The method proposed in [106] utilizes the red channels as an indicator to hide the secret message in the fifth and sixth bits of either the green or the blue channels of the cover image. The count of ones in the red channel determines which channel is currently in use. If the count is even, the green channel is used to embed the secret data; otherwise, the blue is used. A similar technique is found

in [107] in which the green channel is used as an indicator. If the count of the number of ones in the green channel is even, the red channel is used. Otherwise, the blue one is used. Utilizing two bits for embedding process of each pixel leads to reasonable payload capacity but enhances its security. Table VII summarizes the references related to PIT based techniques.

F. Steganographic Techniques Based on Combination of Different Concepts

In this section, several techniques are seen to apply different concepts to achieve secure steganography. In the scheme proposed in [108], Huffman coding is applied to the secret message to reduce its size. Then, based on the location (x, y) of the current pixel, either a bitwise XNOR operation or Fibonacci algorithm is applied to embed the secret message. If x is less than y, the XNOR operation is employed to embed the secret bits in the green or blue channels, and the red channel is used as an indicator. If y is greater than x, the Fibonacci algorithm is applied to embed secret bits in the second LSB of the Fibonacci virtual green bit plane. Otherwise, the pixel is skipped. In [109] the author suggests a steganography technique based on control bit and chaotic bit stream. A chaotic LM is used to generate a chaotic bit stream. Then, this stream is XORed with the LSB of the cover image pixels to create a control bit matching with the corresponding secret message bit. Therefore, the embedding process may change the LSB value or be kept intact based on specific criteria. A technique based on Adaptive Directional Pixel Value Differencing (ADPVD) is suggested in [110]. The aim is to enhance embedding capacity and security. The method starts by evaluating the PVD embedding capacity by traversing three directions: the horizontal, vertical, and diagonal directional edges. That is accomplished after partitioning the original image into two-pixel non-overlapping blocks. The highest capacity rate is calculated for each color channel, and then the appropriate direction is selected.

In [111], the authors suggest a steganography technique based on variant expansion and modulus function. The purpose is to improve embedding capacity as well as the security of the stego image. The cover image is divided into non-overlapping blocks of two pixels. The proposed method considers multiple directions for each color channel and adaptively selects the appropriate embedding direction that achieves the highest embedding capacity. Instead of only considering positive differences, this method selects positive and negative difference values to hide secret data. In [112], the authors present a steganographic method for an RGB image based on the PVD technique. As in most PVD based techniques, the cover image is partitioned into sequential non-overlapping blocks of two pixels. Then, two-color channels combinations are created: red and green color components for the first combination, and green and blue color components for the second. Next, the secret data is embedded in each block using the PVD technique and then examined and readjusted to obtain the modified three-color combination. This operation utilizes a particular threshold to control the embedding capacity of each block in order to minimize distortion.

TABLE VII. SUMMARY OF PIT BASED STEGANOGRAPHY TECHNIQUES

Ref	Features and Pros	Cons	Evaluations	Metrics
[1]	• The embedding is based on the indicator bit, the 7th bit of a pixel value p and p+1 • Robust and secure steganography • Simple implementation • The embedding process alters the pixel value by +2 or -2.	• Lack of steganalysis. • Subject to statistical steganalysis • All image regions are considered affects security and imperceptibility • Low capacity	256x256 from USC-SIPI-ID dataset. Payload: 2: 10 KB PSNR: 55.39: 48.39 bpp=0.031: 0.16	PSNR, MSE, Histogram
[16]	• The indicator channel selection is secret message length dependant to enhance security • Produces low visual distortion • Hiding process takes into account the pixel's intensity • Standard PIT algorithm uses 2-bits LSB's	• It uses a fixed number of bits per channel that could cause noticeable distortion • Uses fixed embedding sequence • Image structure not considered • Steganalysis evaluation is missing	512×384 BMP image 2: 8KB, 256x256 avg PSNR= 43.65: 52.81	PSNR, Histogram, Mean, standard deviations
[25]	• Indicator channel selected based on the message length • Data channel is selected according to 3LSBs of indicator channel • Improved security due to using indicator with multi-mode indicators with adaptive channel embedding	• Image structure not considered • Steganalysis attacks missing • Limited capacity due pixels escaping •	RGB images Payloads: 1Kb, 2Kb, 4Kb Avg. PSNR:64.45, 62.45, 60.45	PSNR, MSE
[105]	• It uses 3 MSB of red channel as an indicator • Number of zero bits in MSB determines storage capacity • To enhance security, input image divided to 4 regions and each part of secret message will store in this region.	• Some pixels are used to store control bits • Image structure not considered • Steganalysis evaluation is missing • Subject to statistical and geometrical attacks	RGB with different sizes 2000 character 2.144122 bpp (avg) avg PSNR 60:57	Histogram, Mean, standard deviations, PSNR
[106]	• Uses 5th & 6th LSBs to enhance security since attacker focus on LSB bits for secret data extraction • Number of ones in the indicator channel determines the data channel • 2 bits of the other channels are used as data bits. • Secret message bits are XORed with predefined secret key to increase security	• All image contents are considered • Steganalysis evaluation is missing • Subject to structural and geometrical attacks • Low payload capacity	RGB with different sizes PSNR (512X512) = 54.65 Max payload =2 bpp	PSNR Histogram Mean values

A steganographic method based on an LM function combined with LSB and PVD to improve security is suggested in [21]. A random key is generated using a chaotic LM that is employed to pick two pixels randomly at a time. In addition, the key value is utilized along with the MOD function to operate either with 3-LSB substitution or PVD, to embed the secret information. In [113], the author claims a novel steganographic approach known as Clustering Modification Directions (CMDs). In this scheme, instead of focusing only on the embedding location clusters of the texture area, clustering of the direction modifications is considered as well. The purpose is to obscure statistical features to resist steganalysis. ±1 LSB embedding approach is considered in this situation. The cover image is segmented into several sub-images as well as the secret message. After embedding the first message portion, the costs of the adjacent segments are updated to cluster the embedding direction. Each time embedding occurs, the costs are updated. A variable-length group of bits substitution based scheme is presented in [114]. In this scheme, embedding for a Group of Bits' Substitution (GBS) is done by replacing a group of bits in a pixel with another group of bits of the same message length. The scheme is designed to work in two variations: 1-bit GBS and 2-bit GBS, which hide 1-bit and 2-bits, respectively. The choice is based on certain predefined conditions. Image imperceptibility and security are improved since most of the pixel values remain unchanged.

The authors in [115] suggest LSB based steganography with Optical Character Recognition (OCR). The concept is based on using a secret message in the form of characters within an image. Character-level features are extracted from the secret image and then embedded into a cover image using the standard LSB. The OCR model has been developed and trained to extract character features. The security perspective of this technique is that, even if the attacker extracts the embedded bits, he still needs to know the OCR model in order to recover the original message. In [116], the authors propose a new method by combining the Right-Most Digit Replacement (RMDR) with an Adaptive Least Significant Bit (ALSB). The cover image is divided into lower texture and higher texture regions. Accordingly, either RMDR or ALSB is chosen to embed the secret message based on RMD rather than bits. RMDR is employed to embed secret bits in the lower texture regions, whereas ALSB is used in high texture regions.

In [117], the RGB cover image is cropped into a predefined number of crops with certain secret coordinates. This number is used to divide the secret text message accordingly. Each part of the message is then embedded using standard LSB into a certain cropped part using a secret sequence. Embedding is achieved using the 3-3-2 sequence. Finally, stego crops are assembled to create the stego image. The coordinates of cropped parts are considered as the key and agreed upon between the two parties. The approach presented in [118] uses two images: a reference image and a cover image. The reference image is divided into N blocks, wherein every block is assigned a unique code. The secret message is also divided into N-bits blocks that are encoded using the block codes obtained earlier. If there is no match between the secret block and the block codes, some LSBs of the reference image need to be altered. Information such as the starting block and traversing direction is incorporated into a secret key, which is then encrypted using the RSA algorithm. Finally, the encoded bit sequence is embedded into the cover image using any LSB technique.

Utilizing bit plane indexes, authors in [119] suggest a secure steganography technique. In this scheme, the secret message is embedded in multiple image bit planes to enhance security without sacrificing capacity payload. It is based on manipulating bit planes indexes. Only the two LSB bits are employed for this purpose. The cover image is initially preprocessed such that the first two bits are not be equal. If the first LSB bit equals the first secret bit, the index is recorded. If they mismatch, the second LSB plane is recorded. In the next turn, the recorded index is in reverse order, for example, if it previously recorded zero, it is one the next time and alteration to LSB is done accordingly. Hence, the final index stream fluctuates between zero and one. In [120], a different perspective is developed to achieve image steganography. Instead of modifying separate image pixels, which causes random noise in the image, this technique changes the image's color palette. All pixels of the same color are transformed into the same color. Therefore, this method achieves a higher user perception. Utilizing quad-trees, authors in [29] present a steganographic method in luminance (L* channel) and chrominance (a* and b* channels) (L*a*b) color space. This approach utilizes a quad-tree segmentation process to partition the spatial domain of the cover image into high correlation and low correlation adaptive size blocks. Embedding is done in the high frequency regions of the DCT of the highly correlated cover image blocks. To improve stego quality, L*a*b color space is utilized. A high quality stego image is guaranteed along with better security, since the embedding takes place only in the high frequency regions that produce minimum image degradation. The performance of this method is affected by the correlation of the image, wherein a highly correlated image is preferred. Table VIII summarizes the related references.

TABLE VIII. SUMMARY OF STEGANOGRAPHIC TECHNIQUES BASED ON DIFFERENT CONCEPTS

Ref	Features and Pros	Cons	PSNR (dB)	Evaluations metrics
[21]	• High capacity and Security by combining LSB&PVD with LM to randomly select two consecutive pixels • Using either LSB or PVD based on mod function and LM • 3 bits embedded in case of LSB	• Lack of steganalysis. • Low visual quality. • All image regions are considered affects security and imperceptibility	512×512 greyscale Payloads: bpp=2.26: 2.37 Avg PSNR: 38.7925	PSNR, Histogram analysis
[29]	• Quad-tree utilized to obtain High & low correlations adaptive-size blocks • Embedding only in high frequency regions	• The performance is affected by the correlation of the image (highly correlated image is preferred	512 x 512 color images with variety of correlations Capacity 76%-90%	SSIM, Combined Capacity Quality Effective-ness

	DCT of chrominance channels (a*b*) • Hence high-quality and better security is guaranteed • Robust against low-density attacks	• Moderate PSNR • Payload capacity not mentioned in the steganalysis	Avg PSNR: 37.317	(CCQE). Attacks: filtering, geometric, and compression attacks.
[108]	• Hybrid bit planes (Fibonacci) with XNOR operation • Huffman coding to compress the secret message. • Effective simple encryption offers high security • High imperceptibility	• Image structure not considered • Steganalysis attacks missing • Computation overhead • Huffman table exchange overhead	RGB 512x512 images Payloads: 8MB & 16KB PSNR: 65.153:74.192	PSNR, MSE, Embedding capacity, Histogram
[109]	• XORing LSB with chaotic bitstream to produce control bit • Embedding based on comparison of LSB with Control bit • Simple implementation • Secure against brute force attack	• System parameters exchange overhead • Image structure not considered • Steganalysis attacks missing • Subject to statistical steganalysis	300x300, 512x512, 1024x1024 greyscale images Payloads: 512: 8192 bytes PSNR= 51.96: 64.82	Correlation coefficient, entropy, PSNR, and Image Fidelity
[110]	• Blocks of two non-overlapping pixels • Adaptively selects the direction for each color channel • Simple implementation. • Enhancement of embedding capacity and security	• Image structure not considered • Steganalysis attacks missing • Moderate visual quality • Three direction calculation - overhead	512x512 bpp =1.65 (greyscale) PSNR=46.71 bpp = 1.63 (RGB) PSNR=51.59	Capacity, PSNR, histogram analysis
[111]	• Based on variant expansion and modulus function • Adaptive embedding direction with positive and negative differences are considered • Simple implementation with enhanced capacity and imperceptibility	• Overhead of parameters exchange • All image regions are considered • Steganalysis attacks missing • Moderate capacity	512x512 RGB from SIPI Highest PSNR= 52.216 Vertical & bpp=1.573 Highest capacity: Diagonal bpp=1.609	MSE, NPCR, embedding capacity, NPCR, UACI, and pixel difference histogram analysis
[112]	• RGB-PVD based scheme. • PVD of the two overlapping channel combination of (R, G) and (G, B) with readjustment of the RGB components • Capacity is improved due to overlapping blocks	• Low visual quality (Low PSNR) • No steganalysis evaluation • Sequential embedding and all image regions are considered affects security and imperceptibility	512x512 RGB images: bpp=2.53 PSNR= 32.79	PSNR, MSE, Payload capacity
[113]	• Modifications are considered along with clustering the directions (+ or -) of embedding modification by updating the cost • Robust against high-dimensional features and ensemble classifiers • Can be used together with schemes with additive distortion functions, such as HILL, S-UNIWARD, WOW	• High complexity • Works with sub images • The costs of pixels within each sub-image are dynamically adjusted	512x512 gray-scale images from BOSSBase image database.	Testing Classification error Steganalytic performance (maxSRMd2) Steganalytic performance (tSRM)
[114]	• A variable-length group of bits substitution-based scheme with two variations (1-bit & 2-bits) • Image imperceptibility and security are improved since the majority of pixel values remain unchanged	• Lack of steganalysis. • Subject to statistical steganalysis • All image regions are considered affects imperceptibility • Moderate capacity	512×512 RGB images 1 bit: Payloads= 1 bpp PSNR=51.64 2 bits: Payloads=2 bpp PSNR= 49.762	PSNR, hiding capacity, image quality index, and pixel difference histograms
[115]	• Enhanced security since character features of text in secret images are used as secret message • Need of a trained OCR model. • standard LSB. • Efficient in training time and accuracy (SMO classifier)	• All image regions are considered (affects imperceptibility) • Computationally expensive • Overhead of classifier training and testing • Preprocessing steps overhead • Character-Feature table handling	RGB Steganalysis Dataset. Payloads: 128×128 grey images. PSNR: 51.107 (1 bpp) 43.094 (2 bpp) 36.444 (3 bpp)	PSNR, MSE, SSIM
[116]	• Combining RMDR with ALSB • The RMDR & ALSB offer high embedding capacity and maintain a good imperceptibility and Security • Texture complexity utilized to use either RMDR or ALSB • Robust against statistical steganalysis.	• Low block size used to determine texture level. • SPAM + ensemble classifier can successfully steganalyze for a higher embedding rate • Not tested against structural detectors • Moderate PSNR	512x512, 256x256, 1024x1024 from UCID, USC-SIPI bpp = 3.052 PSNR= 39.00	PSNR, Q, RS-analysis, pixel difference histogram analysis, and SPAM features under ensemble classifier steganalysis

[117]	• Uses 3-3-2 sequence and cropping cover image into k parts and the secret message is divided into k parts • Embedding using 3-3-2 sequence • Security is related to number of parts, coordinates, and sequence pattern	• Image structure not considered • Steganalysis evaluation is missing • Subject to statistical and structural steganalysis • Fixed embedding sequence	512x512 RGB images PSNR: 62.5332	PSNR, MSE
[118]	• Message encoding using two images divided into k blocks. • Secret message encoded using reference image and then embedded in the LSB • Secret key encrypted using RSA involves: starting block, traversing direction, etc	• Image structure not considered • Steganalysis evaluation is missing • Subject to statistical and structural steganalysis • Two images needed • Key exchange overhead	256x256 Payload= 2000 bits Avg. PSNR = 71.48 Payload= 16000 bits Avg. PSNR = 62.36	PSNR, MSE, Histogram
[119]	• Manipulates bit-planes indexes to enhance security. • 2 LSB should be in the form 01,10. The cover image is pre-processed accordingly and the vector of indices to be sent • Robust against PoV, WS steganalysis	• Handling of the vector of indices • All image contents are considered • Modern steganalysis can attack • Not robust against MLSB-WS steganalyser when bpp=1	512 x 512 Greyscale Payload: 20%: 100% PSNR: ~47 for bpp=1	PSNR, PoV, WS steganalysis, MLSB-WS steganalysis
[120]	• Changes the color palette • All pixels of the same color are changed to the same color • Achieves a higher user perception. • Allows resistance to analysis of adjacent pixel colors	• Its capacity is very dependent on a color palette • Not resistant to color palette analysis and standard palette images • Need to test over modern Steganalysis	512 × 512 bpp = 0.35: 1.37 PSNR: 52.74:58.10	PSNR, SSIM, EC

VI. Observations, Discussion, and Recommendations

A. Observations

- Chaotic based randomness: even though chaotic based LSB steganography achieves higher security, the payload capacity attained is low, and there exists low robustness against statistical and geometric attacks.
- Secret message encryption-based approach: this concept provides higher security, but the complexity is high, especially while using substitution-permutation encryption.
- Image encryption: using standard encryption techniques, multilevel encryption techniques, and chaotic based techniques provides good security. However, the overall system overhead is high.
- Virtual multi-bit plane-based steganography: this paradigm achieves better payload capacity and higher security as the possibility of randomness is greater. Nonetheless, the secret data can deteriorate if there is a slight stego image change by attackers. It is particularly vulnerable to non-statistical steganalysis (geometrical attacks) such as rotation, scaling, and cropping.
- Region based steganography: these techniques achieve good robustness and security, but the embedding capacity in general is low.

B. Discussion

Hiding secret information inside a cover image without introducing suspicious artefacts is the main objective of image steganography. In addition, high security, good imperceptibility, and high embedding rate are desirable and challengeable goals. Researchers have been working on enhancing the performance of steganographic algorithms in terms of achieving high security, high imperceptibility, and high payload capacity. Yet, the optimum goal has not been reached since the challenging criteria oppositely affect each other. When the main consideration is security, the technique should hide the presence of embedded data inside the cover image from the attacker's attention. In addition, it should obscurely hide the data so that attackers cannot identify the original secret message even if they detect its presence. Several concepts have recently been utilized to achieve high security in image steganography.

For securing steganographic techniques, the concept employed most widely is encryption, which has been used to add a layer of security. Encryption can be attained utilizing standard encryption techniques such as AES, 3DES, and RSA along with secret keys to enhance the overall system security. On the other hand, user-defined encryption algorithms are also utilized via the concepts of permutation only, substitution only, or both. The level of security can also be boosted much by embedding the secret data in non-sequential order, that is, by using random sequences to scatter the secret bits all over the cover image. Existing chaotic functions are famous for producing random numbers that can be exploited to create random sequences. Further, some researchers rely on varied concepts to generate such randomness. Pixel channels indicator is another paradigm that has been used to indicate the presence or absence of secret information and identifies the color channel being used.

The visual aspects of an image are also utilized to achieve a level of security, since the image is composed of smooth regions and non-smooth regions also known as low frequency and high frequency regions. Embedding data in a smooth area can raise distortion levels; this breaches the confidentiality of secret data. On the other hand, exploiting the non-smooth regions and edges as embedding locations does not leave evidence of the existence of secret data. Employment of number systems to create virtual bit planes is another means of hiding the secret information without creating noticeable distortion as well as of hiding pixels' relations. Such attributes overcome the security limitations of the standard spatial domain techniques. Frequency domain steganography is another approach that has been followed to guarantee the

security of steganography by choosing appropriate locations to embed secret data. This technique avoids manipulating pixels directly and instead uses transform procedures, thereby leading to good imperceptibility. However, the embedding capacity is limited and the computational cost is higher.

C. Recommendations

The recommendations of this study are as follows:

- Combining edge-based steganography with randomness-based concepts to achieve higher security approaches that resist statistical steganalysis;
- Utilizing the existing encryption methods to add an extra layer of security;
- Applying the adaptiveness concept by combining multiple hiding techniques based on some image attributes or user-defined criteria (techniques such as quad tree search are useful to segregate the cover image into various segments with different attributes);
- Instead of utilizing the entire image to embed the secret data, hiding data in particular regions known as the Region of Interest (ROI), which resists statistical attacks by breaking the statistics relations of adjacent pixels;
- Employing the optimization concept to enhance the security of chaotic based steganography (machine learning techniques are an appropriate method to achieve such a goal);
- Drawing more attention to images in the YCBCR color system, since it has received less attention in this context; and
- Considering 3D for embedding secret data, as very few attempts have been made in this domain.

VII. Conclusion

Image steganography is used to hide secret information inside a cover image. It is frequently used to guarantee confidentiality while sending information over an untrusted network. A comprehensive review of image steganography in the spatial domain was carried out utilizing recent research mainly through IEEE Explore, ScienceDirect, Springer Link, and other databases. Most methods utilize LSB based steganography in the spatial domain and its variants due to its simplicity and effectiveness. In addition, PVD, EMD, PIT have been employed as well. In this work, a review of image steganography in the spatial domain in general and, more precisely, of the security aspect, led to its classification into multiple categories. These categories are as follows: randomization based, encryption based, randomization and encryption based steganographic techniques, image encryption, region-based steganography, multiple bit-planes based, pixel indicator techniques, and other steganographic techniques based on combinations of different concepts. At the end of this review, some discussion of the gaps and the future scope of the above-mentioned concepts has been included. In addition, future recommendations for new and current researchers interested in this field are provided.

Acknowledgment

This work was supported by Universiti Kebangsaan Malaysia under research grant GP-2021-K011439.

References

[1] K. Joshi, S. Gill, and R. Yadav, “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,” *Journal ofComputer Networks and Communications*, vol. 2018, p. 10, 2018, doi: https://doi.org/10.1155/2018/9475142.

[2] O. H. Alhabeeb, F. Fauzi, and R. Sulaiman, “A Review of Modern DNA-based Steganography Approaches,” *IJACSA* , vol. 12, no. 10, 2021, doi: 10.14569/IJACSA.2021.0121021.

[3] M. A. Majeed, R. Sulaiman, Z. Shukur, and M. K. Hasan, “A review on text steganography techniques,” *Mathematics*, vol. 9, no. 21. MDPI, Nov. 01, 2021. doi: 10.3390/math9212829.

[4] M. A. Majeed and R. Sulaiman, “An improved LSB image steganography technique using bit-inverse in 24 bit colour image,” *J Theor Appl Inf Technol*, vol. 80, no. 2, 2015.

[5] A. H. Ali, L. E. George, A. A. Zaidan, and M. R. Mokhtar, “High capacity, transparent and secure audio steganography model based on fractal coding and chaotic map in temporal domain,” *Multimed Tools Appl*, vol. 77, no. 23, pp. 31487–31516, Dec. 2018, doi: 10.1007/s11042-018-6213-0.

[6] S. Kamil, M. Ayob, Siti Norul Huda Sheikh Abdullah, and M. Zulkifli Ahmad, “Lightweight and Optimized Multi-Layer Data Hiding using Video Steganography,” *(IJACSA)*, vol. 9, no. 12, 2018, doi: 10.14569/IJACSA.2018.091237.

[7] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, “Image Steganography : A Review of the Recent Advances,” *IEEE ACCESS*, vol. 9, pp. 23409–23423, 2021, doi: 10.1109/ACCESS.2021.3053998.

[8] P. C. Mandal, I. Mukherjee, G. Paul, and B. N. Chatterji, “Digital image steganography: A literature survey,” *Inf Sci (N Y)*, vol. 609, pp. 1451–1488, Sep. 2022, doi: 10.1016/j.ins.2022.07.120.

[9] A. Rashid and M. K. Rahim, “Critical analysis of stegauography ‘An art of hidden writing,’” *International Journal of Security and its Applications*, vol. 10, no. 3, pp. 259–282, 2016, doi: 10.14257/ijsia.2016.10.3.24.

[10] R. Article, A. K. Sahu, and M. Sahu, “Digital image steganography and steganalysis : A journey of the past three decades,” *Open Computer Science*, vol. 10, no. 1, pp. 296–342, 2020, doi: https://doi.org/10.1515/comp-2020-0136.

[11] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, “Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research,” *Neurocomputing*, vol. Volume 335, pp. 299-326, 2019, doi: 10.1016/j.neucom.2018.06.075.

[12] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, and S. W. Baik, “A Novel Magic LSB Substitution Method (M-LSB-SM) using Multi-Level Encryption and Achromatic Component of an Image,” *Multimed Tools Appl 75, 14867–14893 (2016). https://doi.org/10.1007/s11042-015-2671-9*.

[13] M. C. Alipour, B. D. Gerardo, and R. P. MEDINA, “LSB Substitution Image Steganography Based on Randomized Pixel Selection and One-Time Pad Encryption,” *BDSIC 2020: 2020 2nd International Conference on Big-data Service and Intelligent ComputationDecember 2020 Pages 1–6https://doi.org/10.1145/3440054.3440055*, pp. 1–6.

[14] C. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” *Pattern Recognit*, vol. 37, no. 3, pp. 469–474, 2004, doi: 10.1016/j.patcog.2003.08.007.

[15] J. Mielikainen, “LSB matching revisited,” *IEEE Signal Process Lett*, vol. 13, no. 5, pp. 285–287, May 2006, doi: 10.1109/LSP.2006.870357.

[16] F. Huang and J. Huang, “Edge Adaptive Image Steganography Based on LSB Matching Revisited,” *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 5, no. 2, 2010, doi: 10.1109/TIFS.2010.2041812.

[17] G. Liu, Z. Zhang, Y. Dai, and S. Lian, "Improved LSB-matching steganography for preserving second-order statistics," *J Multimed*, vol. 5, no. 5, pp. 458–463, Oct. 2010, doi: 10.4304/jmm.5.5.458-463.

[18] S. Sun, "A Novel edge based image steganography with 2k correction and Huffman encoding," *Inf Process Lett*, no. September, 2015, doi: 10.1016/j.ipl.2015.09.016.

[19] Z. F. Yaseen, "Image Steganography Based on Hybrid Edge Detector to Hide Encrypted Image using Vernam Algorithm," in *2nd Scientific Conference of Computer Sciences (SCCS), University of Technology - Iraq Image*, IEEE, 2019, pp. 75–80.

[20] K. Gaurav and U. Ghanekar, "Journal of Information Security and Applications Image steganography based on Canny edge detection , dilation operator and hybrid coding," *Journal of Information Security and Applications*, vol. 41, pp. 41–51, 2018, doi: 10.1016/j.jisa.2018.05.001.

[21] S. Prasad and A. K. Pal, "Logistic Map-Based Image Steganography Scheme Using Combined LSB and PVD for Security Enhancement," *Advances in Intelligent Systems and Computing*, vol. 814, pp. 203–214, 2019, doi: 10.1007/978-981-13-1501-5.

[22] M. Tools, K. Bailey, and K. Curran, "An Evaluation of Image Based Steganography Methods," *International Journal of Digital Evidence*, vol. 2, no. 2, pp. 55–88, 2003, doi: 10.1007/s11042-006-0008-4.

[23] K. Muhammad, J. Ahmad, N. U. Rehman, Z. Jan, and R. J. Qureshi, "A Secure Cyclic Steganographic Technique for Color Images using Randomization," *Technical Journal, University of Engineering and Technology Taxila*, vol. 19, no. III, pp. 57–64, 2014.

[24] A. A. Gutub, "Pixel Indicator Technique for RGB Image Steganography," *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE*, vol. 2, no. 1, pp. 56–64, 2010, doi: 10.4304/jetwi.2.1.56-64.

[25] J. Pandey, K. Joshi, M. Jangra, and M. Sain, "Pixel Indicator Steganography Technique with Enhanced Capacity for RGB Images," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, 2019, pp. 738–743.

[26] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Communications Letters*, vol. 10, no. 11, pp. 781–783, 2006, doi: 10.1109/LCOMM.2006.060863.

[27] Z. S. Younus and M. K. Hussain, "Image steganography using exploiting modification direction for compressed encrypted data," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 6, pp. 2951–2963, Jun. 2019, doi: https://doi.org/10.1016/j.jksuci.2019.04.008.

[28] S. Venkatraman, A. Abraham, and M. Paprzycki, "Significance of steganography on data security," *International Conference on Information Technology: Coding Computing, ITCC*, vol. 2, pp. 347–351, 2004, doi: 10.1109/ITCC.2004.1286660.

[29] M. Baziyad, T. Rabie, and I. Kamel, "L * a * b * color space high capacity steganography utilizing quad-trees," *Multimedia Tools and Applications (2020)*, pp. 25089–25113, 2020.

[30] G. G. Rajput and Ramesh Chavan, "A Novel Approach for Image Steganography Based On Random LSB Insertion in Color Images," in *Proceedings of the International Conference on Intelligent Computing Systems*, 2017, pp. 265–273. doi: https://dx.doi.org/10.2139/ssrn.3131654.

[31] M. G. Gouthamanaath, "Hiding Three Binary Images in a Grayscale Image with Pixel Matching Steganography and Randomization technique," in *Proceeding of 2018 IEEE International Conference on Current Trends toward Converging Technologies, Coimbatore, India Hiding*, IEEE, 2018, pp. 1–8.

[32] M. G. Gouthamanaath and A. Kangaiammal, "Hiding binary image in a grayscale image using Pixel Matching and Randomization Technique," in *Proc. of the Fourth Intl. Conf. Advances in Computing, Communication and Information Technology- CCIT 2016*, 2016, pp. 74–78. doi: 10.15224/978-1-63248-092-7-30.

[33] U. A. E. Ali, "A LSB Based Image Steganography Using Random Pixel and Bit Selection for High Payload," no. August, pp. 24–31, 2021, doi: 10.5815/ijmsc.2021.03.03.

[34] J. L. Pichardo Méndez, L. Palacios Luengas, R. F. Martínez González, O. Jiménez Ramírez, and R. Vázquez Medina, "LSB Pseudorandom Algorithm for Image Steganography Using Skew Tent Map," *Arab J Sci Eng*, vol. 45, no. 4, pp. 3055–3074, 2020, doi: 10.1007/s13369-019-04272-0.

[35] C. Pak, J. Kim, K. An, C. Kim, K. Kim, and C. Pak, "A novel color image LSB steganography using improved 1D chaotic map," *Multimed Tools Appl*, vol. 79, no. 1–2, pp. 1409–1425, 2020, doi: https://doi.org/10.1007/s11042-019-08103-0.

[36] Z. Rim, A. Afef, E. Ridha, and Z. Mourad, "Beta Chaotic Map Based Image Steganography," in *12th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2019) and 10th International Conference on EUropean Transnational Education (ICEUTE 2019)*, Springer, Cham, 2019, pp. 97–104. doi: 10.1007/978-3-030-20005-3.

[37] A. V. Gahan and G. D. Devanagavi, "A Secure Steganography Model Using Random-Bit Select Algorithm," in *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, Bengaluru, Dec. 2020. doi: 10.1109/ICAECC50550.2020.9339474.

[38] M. A. F. Al-Husainy and D. M. Uliyan, "A SECRET-KEY IMAGE STEGANOGRAPHY TECHNIQUE USING RANDOM CHAIN CODES," *International Journal of Technology*, vol. 10, no. 4, pp. 731–740, 2019, doi: https://dx.doi.org/10.14716/ijtech.v10i4.653.

[39] S. Dagar, "Highly Randomized Image Steganography using Secret Keys," in *IEEE International Conference on Recent Advances and Innovations in Engineering*, IEEE, 2014, pp. 1–5. doi: doi: 10.1109/ICRAIE.2014.6909116.

[40] S. A. Nie, G. Sulong, R. Ali, and A. Abel, "The use of least significant bit (LSB) and knight tour algorithm for image steganography of cover image," vol. 9, no. 6, pp. 5218–5226, 2019, doi: 10.11591/ijece.v9i6.pp5218-5226.

[41] E. Alrashed and S. S. Alroomi, "Hungarian-Puzzled Text with Dynamic Quadratic Embedding Steganography," vol. 7, no. 2, pp. 799–809, 2017, doi: 10.11591/ijece.v7i2.pp799-809.

[42] M. Saidi, H. Hermassi, R. Rhouma, and S. Belghith, "LSB-Hamming based Chaotic Steganography," in *12th International Conference for Internet Technology and Secured Transactions (ICITST)*, Cambridge, UK: IEEE, 2017, pp. 29–34. doi: 10.23919/ICITST.2017.8356340.

[43] L. Laimeche, A. Meraoumia, and H. Bendjenna, "Enhancing LSB Embedding Schemes Using Chaotic Maps Systems," *Neural Comput Appl*, vol. 32, pp. 16605–16623, 2020, doi: https://doi.org/10.1007/s00521-019-04523-z.

[44] H. Elkamchouchi, Wessam M. Salama, and Yasmine Abouelseoud, "Data Hiding in a Digital Cover Image using Chaotic Maps and LSB Technique," in *12th International Conference on Computer Engineering and Systems (ICCES)*, Cairo, Egypt, 2017. doi: 10.1109/ICCES.2017.8275302.

[45] S. Mukherjee and G. Sanyal, "A chaos based image steganographic system," *Multimed Tools Appl*, vol. 77, no. 21, pp. 27851–27876, 2018.

[46] S. S. Shankar and A. Rengarajan, "Puzzle based Highly Secure Steganography," in *International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, 2017. doi: 10.1109/ICAMMAET.2017.8186742.

[47] B. Mondal, "A Secure Steganographic Scheme Based on Chaotic Map and DNA Computing," *In: Sharma, D.K., Balas, V.E., Son, L.H., Sharma, R., Cengiz, K. (eds) Micro-Electronics and Telecommunication Engineering. Lecture Notes in Networks and Systems*, vol. 106, pp. 545–554, 2020, doi: https://doi.org/10.1007/978-981-15-2329-8_55.

[48] S. SOLAK and U. ALTINIŞIK, "A New Approach for Steganography: Bit Shifting Operation of Encrypted Data in LSB (SED-LSB)," *Journal of information technology*, vol. 12, no. 1, pp. 75–82, 2019, doi: 10.17671/gazibtd.435437.

[49] A. Setyono, D. R. Ignatius, and M. Setiadi, "Securing and Hiding Secret Message in Image using XOR Transposition Encryption and LSB Method," in *Journal of Physics: Conference Series, Volume 1196, International Conference on Information System, Computer Science and Engineering*, 2019. doi: 10.1088/1742-6596/1196/1/012039.

[50] R. Dumre and A. Dave, "Exploring LSB Steganography Possibilities in RGB Images," in *12th International Conference on Computing*

Communication and Networking Technologies, ICCCNT 2021, Kharagpur: IEEE, 2021. doi: 10.1109/ICCCNT51525.2021.

[51] S. Chauhan, Jyotsna, J. Kumar, and A. Doegar, "Multiple layer Text security using Variable block size Cryptography and Image Steganography," in *3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, IEEE, 2017. doi: https://doi.org/10.1109/CIACT.2017.7977303.

[52] M. M. Abdel-aziz, K. M. Hosny, and N. A. Lashin, "Improved data hiding method for securing color images," *Multimed Tools Appl*, vol. 80, pp. 12641–12670, 2021, doi: https://doi.org/10.1007/s11042-020-10217-9.

[53] G. Maji, S. Mandal, and S. Sen, "Cover independent image steganography in spatial domain using higher order pixel bits," *Multimed Tools Appl*, vol. 80, pp. 15977–16006, 2021, doi: https://doi.org/10.1007/s11042-020-10298-6.

[54] S. Zhang, L. Yang, X. Xu, and T. Gao, "Secure Steganography in JPEG Images Based on Histogram Modification and Hyper Chaotic System," vol. 10, no. 1, pp. 40–53, 2018, doi: 10.4018/IJDCF.2018010104.

[55] S. Rahman *et al.*, "A Novel Approach of Image Steganography for Secure Communication Based on LSB Substitution Technique," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 31–61, 2020, doi: 10.32604/cmc.2020.09186.

[56] F. B. Calanda, A. M. Sison, M. R. D. Molato, and R. P. Medina, "A Modified Least Significant Bit Randomized Embedding Method based on Image Partitioning and Columnar Transposition with Encryption," in *ICCBD 2019: Proceedings of the 2nd International Conference on Computing and Big Data*, ACM, 2019, pp. 68–72. doi: https://doi.org/10.1145/3366650.3366662.

[57] O. S. Sitompul, Z. Situmorang, F. R. Naibaho, and E. B. Nababan, "STEGANOGRAPHY WITH HIGHLY RANDOM LINEAR CONGRUENTIAL GENERATOR FOR SECURITY ENHANCEMENT," *2018 Third International Conference on Informatics and Computing (ICIC)*, pp. 1–6, 2018, doi: 10.1109/IAC.2018.8780445.

[58] B. Srinivasan, S. Arunkumar, and K. Rajesh, "A Novel Approach for Color Image , Steganography Using NUBASI and Randomized , Secret Sharing Algorithm," *Indian J Sci Technol*, vol. 8, no. April, pp. 228–235, 2015, doi: 10.17485/ijst/2015/v8iS7/.

[59] Z. S. Younus and M. K. Hussain, "Image steganography using exploiting modification direction for compressed encrypted data," *Journal of King Saud University – Computer and Information Sciences*, 2019, doi: https://doi.org/10.1016/j.jksuci.2019.04.008.

[60] D. Ghosh, A. K. Chattopadhyay, and A. Nag, "A Novel Approach of Image Steganography with Encoding," *In: Chakraborty, M., Chakrabarti, S., Balas, V., Mandal, J. (eds) Proceedings of International Ethical Hacking Conference 2018. Advances in Intelligent Systems and Computing*, vol. 811, pp. 115–124, 2019, doi: 10.1007/978-981-13-1544-2.

[61] S. Elshare and N. N. El-emam, "Modified Multi-Level Steganography to Enhance Data Security," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 10, no. 3, pp. 509–525, 2018.

[62] H. R. Kareem, H. H. Madhi, and K. A. Mutlaq, "Hiding encrypted text in image steganography," *Periodicals of Engineering and Natural Sciences*, vol. 8, no. 2, pp. 703–707, 2020.

[63] M. N. Abdulwahed, "An effective and secure digital image steganography scheme using two random function and chaotic map," *J Theor Appl Inf Technol*, vol. 98, no. 1, pp. 78–91, 2020.

[64] K. Kordov and S. Zhelezov, "Steganography in color images with random order of pixel selection and encrypted text message embedding," *PeerJ Comput. Sci.*, pp. 1–21, 2021, doi: 10.7717/peerj-cs.380.

[65] P. Yadav and M. Dutta, "3-Level Security Based Spread Spectrum Image Steganography with Enhanced Peak Signal to Noise Ratio," in *2017 Fourth International Conference on Image Information Processing (ICIIP)*, IEEE, 2017, pp. 122–126. doi: 10.1109/ICIIP.2017.8313696.

[66] M. C. E. M. Kasapbaşi, "A New Chaotic Image Steganography Technique Based on Huffman Compression of Turkish Texts and Fractal Encryption With Post-Quantum Security," *in IEEE Access*, vol. 7, pp. 148495–148510, 2019, doi: 10.1109/ACCESS.2019.2946807.

[67] R. F. Martínez-González, J. A. Díaz-Méndez, L. Palacios-Luengas, J. López-Hernándezd, and R. Vázquez-Medina, "A steganographic method using Bernoulli' s chaotic maps," *Computers and Electrical Engineering*, vol. 54, no. C, pp. 435–449, 2016, doi: 10.1016/j.compeleceng.2015.12.005.

[68] M. Kasapbas and W. Elmasry, "New LSB-based colour image steganography method to enhance the efficiency in payload capacity , security and integrity check," *Sādhanā*, vol. 43, no. 68, 2018, doi: https://doi.org/10.1007/s12046-018-0848-4.

[69] H. Alhelow, "Highly Secure Steganography-Based System with Three Layers of Protection," *EasyChair*, 2021.

[70] U. I. Assad and G. M. Bhat, "Hiding in encrypted images : a three tier security data hiding technique," *Multidimens Syst Signal Process*, 2015, doi: 10.1007/s11045-015-0358-z.

[71] E. J. Kusuma, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, "A Combination of Inverted LSB , RSA , and Arnold Transformation to get Secure and Imperceptible Image Steganography," *Journal of ICT Research and Applications*, vol. 12, no. 2, pp. 103–122, 2018, doi: 10.5614/itbj.ict.res.appl.2018.12.2.1.

[72] A. Hussain and P. Bora, "A Highly Secure Digital Image Steganography Technique Using Chaotic Logistic Map and Support Image," in *Proceedings of 2018 IEEE International Conference on Information Communication and Signal Processing (ICSP 2018) A*, IEEE, 2018, pp. 69–73.

[73] J. S. Neenu and E. B. Varghese, "A novel approach for SCC algorithm using pattern based image steganography," *Proceedings of the International Conference on Inventive Computation Technologies, ICICT 2016*, vol. 2016, pp. 1–6, 2016, doi: 10.1109/INVENTIVE.2016.7830241.

[74] P. Das, S. C. Kushwaha, and M. Chakraborty, "Data Hiding Using Randomization and Multiple Encrypted Secret Images," in *2015 International Conference on Communications and Signal Processing (ICCSP), 2015*, IEEE, 2015, pp. 298–302. doi: 10.1109/ICCSP.2015.7322892.

[75] K. Tiwari and S. J. Gangurde, "LSB Steganography Using Pixel Locator Sequence with AES," *2021 Second International Conference on Secure Cyber Computing and Communication (ICSCCC)*, vol. 255, pp. 302–307, 2021, doi: 10.1109/ICSCCC51823.2021.9478162.

[76] R. Shanthakumari, S. Varadhaganapathy, S. Vinothkumar, and B. Bharaneeshwar, "Data hiding in Image steganography using Range Technique for secure communication," in *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2021. doi: https://doi.org/10.1109/ICAECT49130.2021.9392480.

[77] T. K. Hue, N. T. Linh, M. Nguyen-duc, and T. M. Hoang, "Data Hiding in Bit-plane Medical Image Using Chaos-based Steganography," in *2021 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, IEEE, 2021. doi: https://doi.org/10.1109/MAPR53640.2021.9585243.

[78] R. K. Sinha, "Chaotic Image Encryption Scheme Based on Modified Arnold Cat Map and Henon Map," *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pp. 1–5, 2018.

[79] P. Ramasamy, V. Ranganathan, S. Kadry, and R. Damaševiˇ, "An Image Encryption Scheme Based on Block Scrambling , Modified Zigzag Transformation and Key Generation Using Enhanced Logistic—Tent Map," *Entropy*, vol. 21, no. 7, p. 656, 2019, doi: 10.3390/e21070656.

[80] L. Teng, X. Wang, and J. Meng, "A chaotic color image encryption using integrated bit-level permutation," *Multimed Tools Appl*, vol. 77, no. 6, pp. 6883–6896, 2017, doi: 10.1007/s11042-017-4605-1.

[81] B. Mondal and T. Mandal, "A light weight secure image encryption scheme based on chaos & DNA computing," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 4, pp. 499–504, 2017, doi: 10.1016/j.jksuci.2016.02.003.

[82] Y. Yang, B. Guan, J. Li, D. Li, Y. Zhou, and W. Shi, "Image compression-encryption scheme based on fractional order hyper- chaotic systems combined with 2D compressed sensing and DNA encoding," *Opt Laser Technol*, vol. 119, no. November 2018, p. 105661, 2019, doi: 10.1016/j.optlastec.2019.105661.

[83] B. kumar Nancharla and M. Dua, "An Image Encryption using Intertwining Logistic map and Enhanced Logistic Map," in *Fifth International Conference on Communication and Electronics Systems (ICCES 2020)*, IEEE, 2020, pp. 1309–1314.

[84] M. Z. Talhaoui, X. Wang, and A. Talhaoui, "A new one-dimensional chaotic map and its application in a novel permutation-less image encryption scheme," *Vis Comput*, 2020, doi: 10.1007/s00371-020-01936-z.

[85] K.S.K.S.Sarma and B.Lavanya, "Digital Image Scrambling based on Sequence Generation," in *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, 2017. doi: 10.1109/ICCPCT.2017.8074317.

[86] S. Saha, R. K. Karsh, and M. Amrohi, "Encryption and Decryption of Images using Secure Linear Feedback Shift Registers," *2018 International Conference on Communication and Signal Processing (ICCSP)*, pp. 295–298, 2018.

[87] B. O. Al-roithy and A. A. Gutub, "Trustworthy image security via involving binary and chaotic gravitational searching within PRNG selections," *IJCSNS International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 167–176, 2020, doi: https://doi.org/10.22937/IJCSNS.2020.20.12.18.

[88] A. Ramesh and A. Jain, "Hybrid Image Encryption using Pseudo Random Number Generators , and Transposition and Substitution Techniques," in *International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15)*, IEEE, 2015. doi: https://doi.org/10.1109/ITACT.2015.7492652.

[89] Y. Luo, J. Yu, W. Lai, and L. Liu, "A novel chaotic image encryption algorithm based on improved baker map and logistic map," *Multimed Tools Appl*, vol. 78, no. 15, Aug. 2019, doi: 10.1007/s11042-019-7453-3.

[90] B. Fathi-vajargah, M. Kanafchian, and V. Alexandrov, "Image Encryption Based on Permutation and Substitution Using Clifford Chaotic System and Logistic Map," *J Comput (Taipei)*, vol. 13, no. 3, pp. 309–326, 2017, doi: 10.17706/jcp.13.3.309-326.

[91] Y. Li, C. Wang, and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Opt Lasers Eng*, vol. 90, no. August 2016, pp. 238–246, 2017, doi: 10.1016/j.optlaseng.2016.10.020.

[92] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-Box," *Inf Sci (N Y)*, vol. 450, pp. 361–377, 2018, doi: 10.1016/j.ins.2018.03.055.

[93] C. Irawan, D. R. I. M. Setiadi, C. A. Sari, and E. H. Rachmawanto, "Hiding and securing message on edge areas of image using LSB steganography and OTP encryption," *Proceedings - 2017 1st International Conference on Informatics and Computational Sciences, ICICoS 2017*, vol. 2018-Janua, no. February 2018, pp. 1–6, 2017, doi: 10.1109/ICICOS.2017.8276328.

[94] A. Jan, S. A. Parah, and B. A. Malik, "A Novel Laplacian of Gaussian (LoG) and Chaotic Encryption Based Image Steganography Technique," in *International Conference for Emerging Technology (INCET)*, IEEE, 2020. doi: https://doi.org/10.1109/INCET49848.2020.9154173.

[95] T. D. Nguyen, S. Arch-int, and N. Arch-int, "An adaptive multi bit-plane image steganography using block data-hiding," *Multimed Tools Appl*, vol. 75, pp. 8319–8345, 2016, doi: 10.1007/s11042-015-2752-9.

[96] Y. Wang, M. Tang, and Z. Wang, "Optik High-capacity adaptive steganography based on LSB and Hamming code," *Optik - International Journal for Light and Electron Optics*, vol. 213, no. March, p. 164685, 2020, doi: 10.1016/j.ijleo.2020.164685.

[97] A. Saeed *et al.*, "An accurate texture complexity estimation for quality-enhanced and secure image steganography," *IEEE Access*, vol. 8, pp. 21613–21630, 2020, doi: 10.1109/ACCESS.2020.2968217.

[98] D. Laishram and T. Tuithung, "A novel minimal distortion-based edge adaptive image steganography scheme using local complexity: (BEASS)," *Multimed Tools Appl*, vol. 80, no. 1, pp. 831–854, Jan. 2021, doi: 10.1007/s11042-020-09519-9.

[99] B. Datta, K. Dutta, and S. Roy, "Data hiding in virtual bit-plane using efficient Lucas number sequences," *Multimed Tools and applications*, vol. 79, pp. 22673–22703, 2020, doi: https://doi.org/10.1007/s11042-020-08979-3.

[100] A. Rehman, T. Saba, T. Mahmood, Z. Mehmood, M. Shah, and A. Anjum, "Data hiding technique in steganography for information security using number theory," *J Inf Sci*, vol. 45, no. 6, pp. 767–778, 2018, doi: 10.1177/0165551518816303.

[101] N. Aroukatos, K. Manes, S. Zimeras, and F. Georgiakodis, "Data Hiding Techniques in Steganography using Fibonacci and Catalan numbers," in *Ninth International Conference on Information Technology*, IEEE, 2012. doi: 10.1109/ITNG.2012.96.

[102] S. Dey, A. Abraham, B. Bandyopadhyay, and S. Sanyal, "Data Hiding Techniques Using Prime and Natural Numbers," *Journal of Digital Information Management*, vol. 6, no. 3, pp. 463–485, 2010, doi: https://doi.org/10.48550/arXiv.1003.3672.

[103] A. A. Abdulla, H. Sellahewa, and S. A. Jassim, "Steganography Based on Pixel Intensity Value Decomposition," in *Proceedings Volume 9120, Mobile Multimedia/Image Processing, Security, and Applications*, Baltimore, Maryland, United States, 2014. doi: https://doi.org/10.1117/12.2050518.

[104] K. Biswas, "A New Pixel Value Decomposition based Image Steganography Method," in *12th International Conference on Computational Intelligence and Communication Networks*, IEEE, 2020, pp. 333–341. doi: 10.1109/CICN.2020.61.

[105] V. Rahmani and M. Mohammadpour, "High hiding capacity steganography method based on pixel indicator technique," in *5th Iranian Joint Congress on Fuzzy and Intelligent Systems - 16th Conference on Fuzzy Systems and 14th Conference on Intelligent Systems, CFIS 2017*, Institute of Electrical and Electronics Engineers Inc., Aug. 2017, pp. 144–149. doi: 10.1109/CFIS.2017.8003673.

[106] A. Sharma, M. Poriye, and V. Kumar, "A Secure Steganography Technique Using MSB," *International Journal of Emerging Research in Management &Technology*, vol. 6, no. 6, pp. 2278–9359, 2017, doi: 10.23956/ijermt.v6i6.270.

[107] S. Ahmed, R. Jaffari & Liaquat, and A. Thebo, "Data Hiding Using Green Channel as Pixel Value Indicator," *International Journal of Image Processing (IJIP)*, vol. 12, no. 3, p. 90, 2018.

[108] A. A. Almayyahi, R. Sulaiman, F. Qamar, and A. E. Hamzah, "High-Security Image Steganography Technique using XNOR Operation and Fibonacci Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 10, pp. 511–522, 2020, doi: 10.14569/IJACSA.2020.0111064.

[109] H. Ogras, "An Efficient Steganography Technique for Images using Chaotic Bitstream," *I. J. Computer Network and Information Security*, vol. 11, no. 2, pp. 21–27, 2019, doi: 10.5815/ijcnis.2019.02.03.

[110] M. A. Hameed, S. Aly, and M. Hassaballah, "An efficient data hiding method based on adaptive directional pixel value differencing (ADPVD)," *Multimed Tools Appl*, vol. 77, pp. 14705–14723, 2018, doi: 10.1007/s11042-017-5056-4.

[111] M. ZULQARNAIN, M. G. GHOUSE, W. SHARIF, G. JILANIE, and A. SHIFA, "AN EFFICIENT METHOD OF DATA HIDING FOR DIGITAL COLOUR IMAGES BASED ON VARIANT EXPANSION AND MODULUS FUNCTION," *Journal of Engineering Science and Technology*, vol. 16, no. 5, pp. 4160–4180, 2021.

[112] S. Prasad and A. K. Pal, "An RGB colour image steganography scheme using overlapping block-based pixel value differencing," *R Soc Open Sci*, vol. 4, p. 161066, 2017, doi: http://dx.doi.org/10.1098/rsos.161066.

[113] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1905–1917, 2015, doi: 10.1109/TIFS.2015.2434600.

[114] G. Swain, "Digital image steganography using variable length group of bits substitution," *Procedia - Procedia Computer Science*, vol. 85, pp. 31–38, 2016, doi: 10.1016/j.procs.2016.05.173.

[115] A. Chatterjee, S. K. Ghosal, and R. Sarkar, "LSB based steganography with OCR : an intelligent amalgamation," *Multimed Tools Appl*, vol. 79, pp. 11747–11765, 2020, doi: https://doi.org/10.1007/s11042-019-08472-6.

[116] M. Hussain, A. W. A. Wahab, N. Javed, and K.-H. Jung, "Hybrid Data Hiding Scheme Using Right-Most Digit Replacement and Adaptive Least Significant Bit for Digital Images," *Symmetry (Basel)*, vol. 8, no. 6, p. 41, 2016, doi: 10.3390/sym8060041.

[117]K. A. Al-afandy, E.-S. M. EL-Rabaie, O. S. Faragallah, A. Elmhalawy, and Gh. M. El-Banby, "High Security Data Hiding Using Image Cropping and LSB Least Significant Bit Steganography," in *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, Tangier, Morocco: IEEE, 2016, pp. 400–404. doi: https://doi.org/10.1109/CIST.2016.7805079.

[118]G. Maji, S. Mandal, S. Sen, and N. C. Debnath, "Dual Image based LSB Steganography," in *2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom) Dual*, IEEE, 2018, pp. 61–66. doi: 10.1109/SIGTELCOM.2018.8325806.

[119]A. A. Abdulla, S. A. Jassim, and H. Sellahewa, "Secure Steganography Technique Based on Bitplane Indexes," in *2013 IEEE International Symposium on Multimedia*, IEEE, 2013, pp. 287–291. doi: 10.1109/ISM.2013.55.

[120]E. Margalikas and S. Ramanauskait, "Image steganography based on color palette transformation in color space," *EURASIP J Image Video Process*, pp. 1–13, 2019, doi: http://doi.org/10.1186/s13640-019-0484-x.