Novel Secure Hybrid Image Steganography Technique Based on Pattern Matching

Ali Hamza*, Danish Shehzad, Muhammad Shahzad Sarfraz, Usman Habib, and Numan Shafi
Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan
[e-mail: {f179026, danish.shehzad, shahzad.sarfraz, usman.habib, numan.shafi}@nu.edu.pk]
*Corresponding author: Ali Hamza

Received July 20, 2020; revised January 3, 2021; accepted February 28, 2021; published March 31, 2021

Abstract

The secure communication of information is a major concern over the internet. The information must be protected before transmitting over a communication channel to avoid security violations. In this paper, a new hybrid method called compressed encrypted data embedding (CEDE) is proposed. In CEDE, the secret information is first compressed with Lempel Ziv Welch (LZW) compression algorithm. Then, the compressed secret information is encrypted using the Advanced Encryption Standard (AES) symmetric block cipher. In the last step, the encrypted information is embedded into an image of size 512 × 512 pixels by using image steganography. In the steganographic technique, the compressed and encrypted secret data bits are divided into pairs of two bits and pixels of the cover image are also arranged in four pairs. The four pairs of secret data are compared with the respective four pairs of each cover pixel which leads to sixteen possibilities of matching in between secret data pairs and pairs of cover pixels. The least significant bits (LSBs) of current and imminent pixels are modified according to the matching case number. The proposed technique provides double-folded security and the results show that stego image carries a high capacity of secret data with adequate peak signal to noise ratio (PSNR) and lower mean square error (MSE) when compared with existing methods in the literature.

Keywords: Image Steganography, Information Hiding, Pair Matching, Secret Data Embedding, Least Significant Bits (LSB) Substitution, Compress Encrypted Data Embedding (CEDE), Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE)

http://doi.org/10.3837/tiis.2021.03.013 **ISSN : 1976-7277**

1. Introduction

The data and information of organizations are among the most important assets that must be secured from unauthorized access. The security of assets can be achieved by ensuring confidentiality, integrity, and availability. The secure communication of information is a major concern of the internet today. Data security can be achieved by applying various techniques known as cryptography, watermarking, and steganography.

Cryptography is the branch of information security where secret information is converted into the unreadable format and transferred from one place to another so that an unauthorized person can't decrypt it [1]. The second method is watermarking which is the science and arts of data hiding technique that is used for authentication and copyright act. The aim of watermarking is the prevention of unethical duplication of medium files and pseudo-ownership claims [2]. Whereas steganography is the science and art of hiding secret messages in cover medium, so that the existence of secret data can't be perceived by an attacker [3]. The cover file, secret information, and stego file are three components of steganography. The stego file is the medium where secret data is embedded in the cover file. Steganography has different types according to its cover medium and each type has different techniques to implement steganography. The text, audio, image, video, and DNA files can be used as a cover medium.

For image steganography, secret information is sent over a communication network by hiding inside the cover image [4]. The image steganography is divided into two domains according to their applicability. In the frequency domain, techniques encode message bits in the orthogonal transform coefficients of the image [5]. The image is converted into different frequencies to embed secret data and then the image is converted back to its original form. The orthogonal transforms of the image have two components, the first is a magnitude that contains information about frequencies and the second is a phase that contains information about how the image will come back to its original form. In the spatial domain techniques, secret information is concealed within the image domain directly [4]. There are many techniques of the spatial domain such as LSB, PVD, Histogram shifting, Difference Expansion, Multiple Bit-planes, Palette Based, Pixel Intensity Modulation and Quantization Based etc.

A very well-known method of image steganography in the spatial domain is LSB substitution in which secret data is embedded inside image using different strategies and only LSBs of cover pixels are modified [6]. The 1-4 bits modification in the LSBs of cover pixels are imperceptible to human vision, but the value of peak signal to noise ratio (PSNR) can be decreased, thus degrading original image quality [7]. Only one to four bits are stored in 8 bit pixel of the cover image in classical techniques of LSB steganography, but in proposed method 8 bits of pixel can store up-to 8 bits of secret data with minimal modification in image pixels. The main objective of the proposed compressed encrypted data embedding (CEDE) method is to enhance the embedding capacity of secret data in a cover image with minimum modification so that the quality of the image remains imperceptible to the human visual system.

The remaining part of paper is organized as follows. Section 2 describes the related work that is presented in main four categories i.e, Block Based LSB Substitution, LSB Substitution using Edge Detection, LSB Substitution using Hybrid Mechanism and Optimal LSB Substitution. The proposed CEDE method is presented in Section 3. Section 4 presents the experimental results. Section 5 makes comparisons with literature and Section 6 concludes the paper.

2. Related Work

The LSB substitution is a well-known method of steganography in spatial domain. In this section various techniques of LSB substitution are discussed that emphasize on high payload of secret data, good stego image quality (PSNR), low error rate in stego image (MSE) and security against steganalysis.

2.1 Block-Based LSB Substitution

Shehzad and Dag [8] proposed a technique in which the cover image is divided into non-overlapping blocks and each block consists of four pixels (2 x 2). The determinant value is calculated for each block by using the decimal values of the block. The determinant is used to embed a secret data bit into a pixel of the block. This technique carries only single secret data bit in one block. This method improves the stego image quality, but the limitation is its embedding capacity where single bit is concealed per block. Zakaria et al. [9] proposed a technique that divides secret data bits into pairs of two bits and divide MSBs of cover pixels into two pairs of two bits. Two pairs of secret data are compared with two pairs of cover pixel and comparison makes four possibilities of matching secret pairs with pixel pairs. Each 2 LSBs of the cover image is changed with respect to matching pair number. This method improves the embedding capacity of secret data in cover image with good PSNR. Shehzad and Dag [10] proposed a technique that divides secret data bits into pairs of two bits and divide 3-7 MSBs of cover pixels into four pairs of two bits. One secret pair is compared with each four pairs of cover pixel and each 2 LSBs of the cover image is changed with respect to matching pair number. The embedding capacity of this technique is one pair of two bits per pixel. This technique improves the PSNR value of stego image. Swain et al. [11] proposed a technique that divides the cover pixels into non-overlapping blocks of four pixels (2 x 2). The secret data is embedded in cover image using hybrid approach of PVD and LSB Substitution respectively. This method improves the PSNR value of stego image but embedding of the secret data is below to one bit per pixel.

Khodaei et al. [12] proposed a technique in which the cover pixels are divided into consecutive non-overlapping blocks of two bits. If the value of both pixels are small in the block, then secret data is embedded into cover image using LSB substitution otherwise PVD of two pixels is calculated. Then, the bits of secret information are embedded in the cover pixels based on PVD value using LSB substitution. This method improves the embedding payload along with good stego image quality. Liu et al. [13] proposed a novel scheme that based on LSB substitution in which 2 bits of secret data are embedded in a single cover pixel according to mapping strategy with 3 bits modification in cover pixel. This method improves stego image quality by PSNR 49.04 dB. Lu et al. [14] proposed a method based on PVD and LSB techniques in which a grayscale image is divided into fixed block size of 1-by-3, 2-by-2 and 3-by-3 to enhance embedding capacity. This technique improves the Five Pixel Pair Differencing (FPPD) scheme in which block size is fixed of size 2-by-3. This method improves the secret data payload by 3 bits per pixel with acceptable stego image quality. Horng et al [15] proposed a method based on Quotient Value Differencing (QVD) and LSB substitution. In the first step of this method, secret data is embedded into high and low mean pixel values by using QVD and LSB Substitution. In the second step, low and high mean values are swapping to embed additional bit. In the third step, secret data is embedded into smooth block by LSB substitution. This method embeds average payload with normal PSNR.

2.2 LSB Substitution Using Edge Detection

Bai et al. [16] proposed a technique that uses edge detection algorithm to divide cover pixels into edge and smooth area pixels. Then the LSB substitution method is used to embed secret data in cover image. The pixels of edge area are used to carry more secret bits as compared to smooth area. This method improves the secret data payload with good stego image quality. Hussain et al. [17] proposed a technique in which the cover pixels are divided in high level and low-level blocks. The LSB substitution is used to embed secret data in low level block and Pixel Value Difference (PVD), PVD shift and modification of prediction error (MPE) is used to embed secret data in high level blocks respectively. This technique improves the secret data payload and robustness against attacks with acceptable stego image quality. Khan et al. [18] proposed a technique in which the cover pixels are divided in high level and low-level color ranges. The cover pixels that fall in a range of low-level color carries more secret bits as compared to high level color pixels. The secret data of size four, three, two and one bit is embedded in cover pixels when the ranges of pixels are falling in between 0-31, 32-63, 63-127 and 28-155 respectively. This technique improves the PSNR value of stego image but embedding payload of the secret data is low. Liao et al. [19] proposed a technique in which the cover image is divided into non-overlapping blocks size of 4 x 4. Each block is further divided into four sub blocks; Upper-left, upper-right, lower-left and lower right. Each sub block consists of four pixels (2 x 2). Then the pixels in each block are divided in edge and smooth pixels. The edge area pixels carry more secret bits as compared to non-edge area. This method improves the embedding payload along with acceptable PSNR.

Jung and Yoo [20] proposed a technique in which the cover image is divided into non-overlapping blocks size of 2 x 2. The edge detector is used to classify edge and smooth pixels. Then, PVD of the four pixels is calculated to hide secret data in cover image. This method improves the embedding payload along with good PSNR value. Khalind et al. [21] proposed a technique that classifies cover image into edge and smooth pixels and embeds more information in edge pixels as compared to smooth pixels. According to this technique, adaptive number K is calculated for each pixel to embed secret data. The quality of stego image depends upon the value of variable K. If secret information is embedded in the cover using low value of K, then PSNR will be high and vice versa. Lee et al. [22] proposed technique in which the cover pixels are divided in edge and non-edge area. The four MSBs of each color channel is sorted and only N bits memory required to sort N bits of RBG image. According to this method, the number of pixels in one row is calculated from size of data. The difference value $delta_i$ is sorted in descending order and selected pixels are replaced with secret message using the LSB method in specific area. This technique improves the secret data payload in cover image with stego image quality. Hong et al. [23] proposed fully exploits the modification of the quantization level method to achieve better performance and efficiency of stego image. This method used a specific threshold to classify block into smooth and complex block. More secret information is stored in a smooth blocks and complex blocks stores only one bit per block. Distortion is handled by Adaptive Pixel Pair Matching (APPM) technique that modify block and reduce distortion. The drawback of this method is very low embedding capacity with 32.27 dB PSNR.

Setiadi et al. [24] proposed a hybrid method based that is a combination of canny and sobel edge detection algorithm. Edges of cover image are detected by canny (C) and sobel (S) edge detection algorithm and results are saved as CSea by taking OR operation of both algorithms. The secret bits are embedded in the edge area of cover image using the LSB method. The secret information is stored in smooth area when there is not enough space in the edge area for secret messages. This method improves the stego image quality but embedding capacity of

secret data in cover image is very low. Rashid et al. [25] proposed a method that separate the planes of color image into Red, Green and Blue plane. Then edges are detected in one plane of color image using edge detection algorithm. This method embedded small secret message in one sharper edge pixel and large secret message in more edge pixels. This method improves the stego image quality but payload is very low.

2.3 LSB Substitution Using Hybrid Mechanism

Chikouche et al. [26] proposed a technique that used Advanced Encryption Standard (AES) algorithm to encrypt secret data, deflation algorithm to compress encrypted data and LSB substitution to hide secret information respectively. In LSB substitution, random set of values are generated for each pixel to embed secret information in cover pixels. The secret information is embedded in red, blue and green channel of the image respectively. This method improves the stego image quality with minimum modification. Muhammad et al. [27] proposed a technique that used Multilevel encryption algorithm (MLEA) to encrypt secret data and stego key-directed adaptive least significant bit (SKA-LSB) substitution to hide secret information in the cover image. The SKA-LSB takes XOR operation of the key and LSB of red channel of a cover pixel. If the remainder is one, then embeds one secret data bit in LSB of the green channel, otherwise embeds one secret data bit in LSB of the blue channel. This method improves the PSNR value of stego image but embedding of the secret data is only one bit per pixel. Muhammad et al. [28] proposed a technique that uses Multilevel encryption (MLE) to encrypt secret data and gray level modification (GLM) to hide secret information in the cover image. In GLM mechanism, if the secret bit is equal to 0 and the cover pixel is even, then the current pixel remains unchanged and vice versa. If the secret bit is equal to 0 and the cover pixel is odd, then subtracts one from the value of the current cover pixel and if the value of secret bit is 1 and the cover pixel is even then adds one to the value of the current cover pixel. This technique improves the PSNR value of stego image, but embedding payload is very low. Kuo et al. [29] proposed a technique in which the secret information is compressed using run length encoding (RLE) scheme and multi-bits generalized exploiting modification direction (MGEMD) algorithm to hide secret information in the cover image.

2.4 Optimal LSB Substitution

Mohamed et al. [30] proposed a technique in which the bits of secret information are embedded in cover pixels based on the value of a variable K that lies in between 2-5 using simple LSB substitution. According to this technique, the secret data bits equal to the value of K are replaced in the K LSBs of cover pixel. The quality of stego image depends upon the value of variable K. If secret data bits are embedded in cover using high value of K, then PSNR will be low and vice versa. Leng et al. [31] proposed a technique that uses interpolation mechanism for scaling of the cover image and LSB substitution to hide secret information in the cover image. The cover image can be scaled up or down for higher embedding of secret data in cover image. This method improves the embedding payload along with good PSNR. Jayapandiyan et al. [32] proposed LSB Substitution method using character sequence optimization. In the first step, header values about secret data are embedded in first few pixel of cover image. In the second step, secret data is embedded in cover image using character sequence optimization. This method improves the PSNR with average payload.

Some studies in related work improves high payload of secret data but stego image quality is just acceptable. Similarly, some studies improve stego image quality, but payload is very low. Therefore, there is need to propose a novel method that improve stego image quality with high payload with low Mean Square Error (MSE).

3. Compress Encrypted Data Embedding Method (CEDE)

In this section, the proposed CEDE method is explained that consists of three important phases to ensure the secure transmission of data. These phases include: Lempel Ziv Welch, AES encryption and a novel steganographic technique. In the first phase, the secret information is compressed using the Lempel Ziv Welch coding, then compressed information is encrypted using the AES encryption. In the last phase, compressed and encrypted information is embedded in the cover pixels using the proposed technique of steganography. The embedding process is explained below and is shown in **Fig. 1**. The **Table 1** describes the steps of embedding process.

3.1 Data Compression for CEDE Method

The Lempel-Ziv Welch (LZW) is a lossless and dictionary-based compression algorithm. The 256 entries are initialized in the dictionary of LZW algorithm to represent ACSII table [33]. The LZW algorithm creates a table of unique strings and each string is identified by unique integers. The LZW algorithm reads the possible longest sequence of K characters and checks this sequence with existing string in the table. If K is matched with existing string, then K is identified by the same integer else string K is added in the table.

In the proposed hybrid method, the secret information is compressed with LZW compression encoding scheme. The LZW algorithm can compress a plaintext file size of 379 KB into a 126 KB file. So, the compression ratio of this algorithm is 66.75% in this case.

3.2 Data Encryption for CEDE Method

The Advanced Encryption Standard (AES) is a symmetric block cipher**.** The AES algorithm is the strongest algorithm that encrypts plaintext into ciphertext in mostly block size of 128 bits. Same key size is used for encryption and decryption at both ends that can be 128, 192 and 256 bits long [34]. The 10, 12 and 14 rounds of encryption process are used for key size of 128, 192 and 256 bits respectively. The rounds are further divided into initial, main and final round. In the initial round, the function of add round key is performed in which a block of plaintext is bitwise XORed with block of round keys. The main round of encryption consists of four types of transformations substitution, shift rows, mixing columns and add round key that are performed respectively on output block of initial round. The final round consists of three transformations substitution, shifting rows and add round key. The initial and final rounds are performed only once, but repetitions of main round are 9, 11 and 13 for key size of 128, 192 and 256 bits respectively.

The compressed secret information is encrypted using Advanced Encryption Standard (AES) that is a symmetric block cipher**.** The AES algorithm encrypts compressed plaintext file size of 126 KB into a 126 KB ciphertext file using 256 bit key.

3.3 Data Embedding Process for CEDE Method

In the embedding phase of CEDE, the compressed and encrypted secret information is embedded in the cover image by the proposed technique of steganography. This technique marks two bits pairs P_N from the secret data, then each cover pixel is divided into four pairs. Assign 8^{th} and 7^{th} bits, 6^{th} and 5^{th} bits, 7^{th} and 6^{th} bits, 8^{th} and 5^{th} bits, 4^{th} and 3^{rd} bits, 2^{nd} and 1^{st} bits of cover pixels to Left Pair (P_L, Pixel), Right Pair (P_R, Pixel), Middle Pair (P_M, Pixel), Outer Pair (P_O, Pixel), Embedding Left (E_L, Pixel) and Embedding Right (E_R, Pixel) respectively. The four secret data pairs are compared with four pairs of each cover pixel. Compare first, second, third and fourth secret data pair with Left Pair (P_L, Pixel), Right Pair

(P_R, Pixel), Middle Pair (P_M, Pixel) and Outer Pair (P_O, Pixel) of cover pixel respectively. The comparison of four secret data pairs with four pairs of cover pixel makes sixteen possibilities of matching. The following cases describe the mapping strategy.

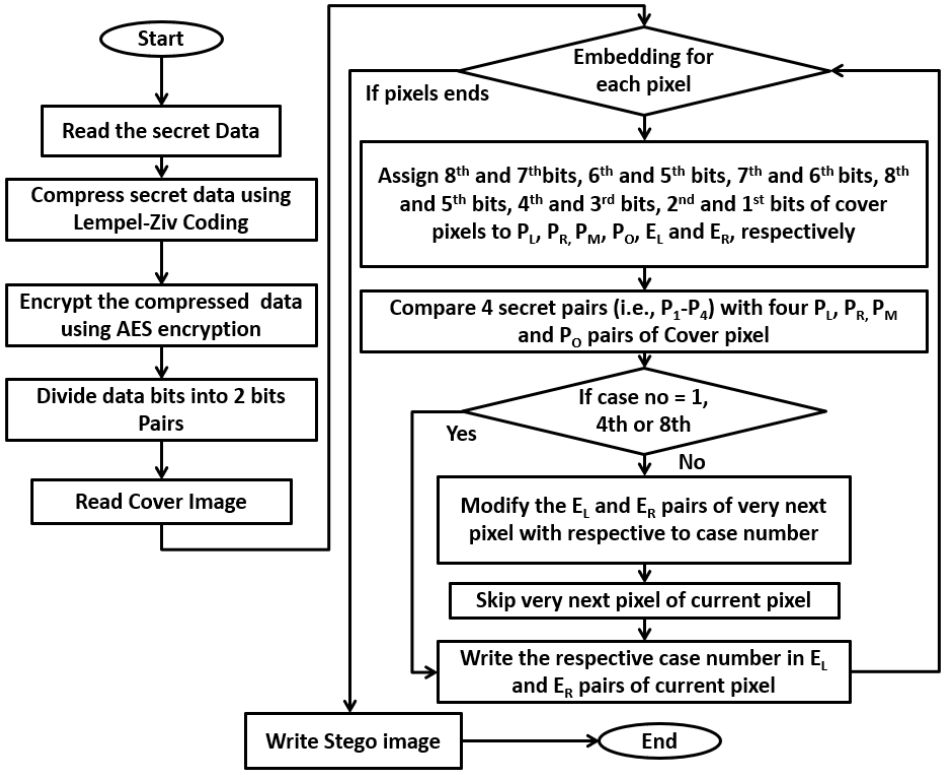


Fig. 1. Embedding Process of CEDE

Case 1:
If first four pairs of secret data (i.e., P_1, P_2, P_3 & P_4) are matched with P_L, P_R, P_M and P_O pairs of cover pixel respectively, then replace E_L and E_R with "11" and "11" respectively. This case carries eight bits of secret data within a single pixel.

Case 2:
If first three pairs of secret data (i.e., P_1, P_2, & P_3) are matched with P_L, P_R and P_M pairs of cover pixel respectively and fourth secret data pair (i.e., P_4) is unmatched with P_O pair, then replace E_L and E_R pairs of current pixel with "11" and "10" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_4" and "P_5" respectively. This case carries ten bits of secret data in two pixels.

Case 3:
If three pairs of secret data (i.e., P_1, P_2, & P_4) are matched with P_L, P_R and P_O pairs of cover pixel respectively and third secret data pair (i.e., P_3) is unmatched with P_M pair, then replace E_L and E_R pairs of current pixel with "11" and "01" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_3" and "P_5" respectively. This case carries ten bits of secret data in two pixels.

Case 4:
If only first two pairs of secret data (i.e., P_1 & P_2) are matched with P_L and P_R pairs of cover pixel respectively and third and fourth secret data pairs (i.e., P_3 & P_4) are unmatched with P_M and P_O pair, then replace E_L and E_R pairs of current pixel with "11" and "00" respectively. This case carries four bits of secret data within single pixels.

Case 5:
If three pairs of secret data (i.e., P_1, P_3, & P_4) are matched with P_L, P_M and P_O pairs of cover pixel respectively and only second secret data pair (i.e., P_2) is unmatched with P_R pair, then replace E_L and E_R pairs of current pixel with "10" and "11" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_2" and "P_5" respectively. This case carries ten bits of secret data in two pixels.

Case 6:
If only two pairs of secret data (i.e., P_1 & P_3) are matched with P_L and P_M pairs of cover pixel respectively and second and fourth secret data pair (i.e., P_2 & P_4) are unmatched with P_R and P_O pairs, then replace E_L and E_R pairs of current pixel with "10" and "10" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_2" and "P_4" respectively. This case carries eight bits of secret data in two pixels

Case 7:
If only two pairs of secret data (i.e., P_1 & P_4) are matched with P_L and P_O pairs of cover pixel respectively and second and third secret data pair (i.e., P_2 & P_3) are unmatched with P_R and P_M pairs, then replace E_L and E_R pairs of current pixel with "10" and "01" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_2" and "P_3" respectively. This case carries eight bits of secret data in two pixels.

Case 8:
If only first pair of secret data (i.e., P_1) is matched with P_L pair of cover pixel and second, third and fourth secret data pairs (i.e., P_2 , P_3 & P_4) are unmatched with P_R, P_M and P_O pairs, then replace E_L and E_R pairs of current pixel with "10" and "00" respectively. This case carries only two bits of secret data within single pixel.

Case 9:
If three pairs of secret data (i.e., P_2, P_3, & P_4) are matched with P_R, P_M and P_O pairs of cover pixel respectively and only first secret data pair (i.e., P_1) is unmatched with P_L pair, then replace E_L and E_R pairs of current pixel with "01" and "11" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_1" and "P_5" respectively. This case carries ten bits of secret data in two pixels.

Case 10:
If only two pairs of secret data (i.e., P_2 & P_3) are matched with P_R and P_M pairs of cover pixel respectively and first and fourth secret data pair (i.e., P_1 & P_4) are unmatched with P_L and P_O pairs, then replace E_L and E_R pairs of current pixel with "01" and "10" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_1" and "P_4" respectively. This case carries eight bits of secret data in two pixels.

Case 11:
If only two pairs of secret data (i.e., P_2 & P_4) are matched with P_R and P_O pairs of cover pixel respectively and first and third secret data pair (i.e., P_1 & P_3) are unmatched with P_L and P_M pairs, then replace E_L and E_R pairs of current pixel with "01" and "01" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_1" and "P_3" respectively. This case carries eight bits of secret data in two pixels.

Case 12:
If only second pair of secret data (i.e., P_2) is matched with P_R pair of cover pixel and first, third and fourth secret data pairs (i.e., P_1 , P_3 & P_4) are unmatched with P_L, P_M and P_O pairs, then replace E_L and E_R pairs of current pixel with "01" and "00" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_1" and "P_3" respectively. This case carries six bits of secret data in two pixels.

Case 13:
If only two pairs of secret data (i.e., P_3 & P_4) are matched with P_M and P_O pairs of cover pixel respectively and first and second secret data pair (i.e., P_1 & P_2) are unmatched with P_L and P_R pairs, then replace E_L and E_R pairs of current pixel with "00" and "11" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_1" and "P_2" respectively. This case carries eight bits of secret data in two pixels.

Case 14:
If only third pair of secret data (i.e., P_3) is matched with P_M pair of cover pixel and first, second and fourth secret data pairs (i.e., P_1 , P_2 & P_4) are unmatched with P_L, P_R and P_O pairs, then replace E_L and E_R pairs of current pixel with "00" and "10" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_1" and "P_2" respectively. This case carries six bits of secret data in two pixels.

Case 15:
If only the fourth pair of secret data (i.e., P_4) is matched with P_O pair of cover pixel and first, second and third secret data pairs (i.e., P_1 , P_2 & P_3) are unmatched with P_L, P_R and P_M pairs, then replace E_L and E_R pairs of current pixel with "00" and "01" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_1" and "P_2" respectively. This case carries four bits of secret data in two pixels.

Case 16:
If first four pairs of secret data (i.e., P_1, P_2, P_3 & P_4) are unmatched with P_L, P_R, P_M and P_O pairs of cover pixel respectively, then replace E_L and E_R with "00" and "00" respectively and also replace E_L and E_R pairs of the imminent pixel with "P_1" and "P_2" respectively. This case carries four bits of secret data in two pixel.

Table 1. Proposed Embedding Steps

Input	AES key (k), Secret Information (I) and Cover Image (C)
Output	Stego Image (S)
Step 1	Compute Secret Information.
(a)	Compress Secret Information (I) using LZW coding.
(b)	Encrypt the compressed Secret Information (I) using AES encryption with AES key (k).

(c)	Convert the compressed encrypted secret information into pairs of two bits i.e., P_1, P_2, P_3, P_4 ……...,
Step 2	Convert bits of each cover pixel into pairs.
(a)	Assign 8^{th} and 7^{th} bits to Left Pair ($P_{L,\ Pixel}$)
(b)	Assign 6^{th} and 5^{th} bits to Right Pair ($P_{R,\ Pixel}$)
(c)	Assign 7^{th} and 6^{th} bits to Middle Pair ($P_{M,\ Pixel}$)
(d)	Assign 8^{th} and 5^{th} bits to Outer pair ($P_{O,\ Pixel}$)
(e)	Assign 4^{th} and 3^{rd} bits to Embedding Left Pair ($E_{L,\ Pixel}$)
(f)	Assign 2^{nd} and 1^{st} bits to Embedding Right Pair ($E_{R,\ Pixel}$)
Step 3	Embedding Process: Compare four secret pairs P_1, P_2, P_3 and P_4 with four pixels pairs P_L, P_R, P_M and P_O.
(a)	If $P_1=P_L$, $P_2=P_R$, $P_3=P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "11" and $E_{R,1}$ with "11".
(b)	If $P_1=P_L$, $P_2=P_R$, $P_3=P_M$ and $P_4\neq P_O$ then replace $E_{L,1}$ with "11", $E_{R,1}$ with "10", $E_{L,2}$ with P_4 and $E_{R,2}$ with P_5.
(c)	If $P_1=P_L$, $P_2=P_R$, $P_3\neq P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "11", $E_{R,1}$ with "01", $E_{L,2}$ with P_3 and $E_{R,2}$ with P_5.
(d)	If $P_1=P_L$, $P_2=P_R$, $P_3\neq P_M$ and $P_4\neq P_O$ then replace $E_{L,1}$ with "11", $E_{R,1}$ with "00", $E_{L,2}$ with P_3 and $E_{R,2}$ with P_4.
(e)	If $P_1=P_L$, $P_2\neq P_R$, $P_3=P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "10", $E_{R,1}$ with "11", $E_{L,2}$ with P_2 and $E_{R,2}$ with P_5.
(f)	If $P_1=P_L$, $P_2\neq P_R$, $P_3\neq P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "10", $E_{R,1}$ with "10", $E_{L,2}$ with P_2 and $E_{R,2}$ with P_4.
(g)	If $P_1=P_L$, $P_2\neq P_R$, $P_3\neq P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "10", $E_{R,1}$ with "01", $E_{L,2}$ with P_2 and $E_{R,2}$ with P_3.
(h)	If $P_1=P_L$, $P_2\neq P_R$, $P_3\neq P_M$ and $P_4\neq P_O$ then replace $E_{L,1}$ with "10" and $E_{R,1}$ with "00".
(i)	If $P_1\neq P_L$, $P_2=P_R$, $P_3=P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "01", $E_{R,1}$ with "11", $E_{L,2}$ with P_1 and $E_{R,2}$ with P_5.
(j)	If $P_1\neq P_L$, $P_2=P_R$, $P_3=P_M$ and $P_4\neq P_O$ then replace $E_{L,1}$ with "01", $E_{R,1}$ with "10", $E_{L,2}$ with P_1 and $E_{R,2}$ with P_4.
(k)	If $P_1\neq P_L$, $P_2=P_R$, $P_3\neq P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "01", $E_{R,1}$ with "01", $E_{L,2}$ with P_1 and $E_{R,2}$ with P_3.
(l)	If $P_1\neq P_L$, $P_2=P_R$, $P_3\neq P_M$ and $P_4\neq P_O$ then replace $E_{L,1}$ with "01", $E_{R,1}$ with "00", $E_{L,2}$ with P_1 and $E_{R,2}$ with P_3.
(m)	If $P_1\neq P_L$, $P_2\neq P_R$, $P_3=P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "00", $E_{R,1}$ with "11", $E_{L,2}$ with P_1 and $E_{R,2}$ with P_2.
(n)	If $P_1\neq P_L$, $P_2\neq P_R$, $P_3=P_M$ and $P_4\neq P_O$ then replace $E_{L,1}$ with "00", $E_{R,1}$ with "10", $E_{L,2}$ with P_1 and $E_{R,2}$ with P_2.
(o)	If $P_1\neq P_L$, $P_2\neq P_R$, $P_3\neq P_M$ and $P_4=P_O$ then replace $E_{L,1}$ with "00", $E_{R,1}$ with "01", $E_{L,2}$ with P_1 and $E_{R,2}$ with P_2.
(p)	If $P_1\neq P_L$, $P_2\neq P_R$, $P_3\neq P_M$ and $P_4\neq P_O$ then replace $E_{L,1}$ with "00", $E_{R,1}$ with "00", $E_{L,2}$ with P_1 and $E_{R,2}$ with P_2.
Step 4	Stop if all pixel of cover image or pairs of secret information have been used.

3.4 Data Extraction Process of CEDE Method

In the extraction phase, read the stego image and make pairs of each pixel. Assign 8^{th} and 7^{th}bits, 6^{th} and 5^{th} bits, 7^{th} and 6^{th} bits, 8^{th} and 5^{th} bits, 4^{th} and 3^{rd} bits, 2^{nd} and 1^{st} bits of cover pixels to Left Pair (P_L, Pixel), Right Pair (P_R, Pixel), Middle Pair (P_M, Pixel), Outer Pair (P_O, Pixel), Embedding Left (E_L, Pixel) and Embedding Right (E_R, Pixel) respectively. Read the E_L and E_R of each stego pixel and extract pairs of secret information from a respective pair of each stego pixel. The flowchart of extraction phase is shown in **Fig. 2**. The **Table 2** describes the steps of extraction process. The following cases describe the extraction strategy.

Case 1:
If both E_L and E_R of stego pixel are "11", then extract four secret data pairs (i.e., P_1, P_2, P_3 & P_4) from P_L, P_R, P_M and P_O pairs of current stego pixel. This case will restore eight secret data bits from the single stego pixel.

Case 2:
If E_L is "11" and E_R is "10" of stego pixel, then extract first three secret data pairs (i.e., P_1, P_2 & P_3) from P_L, P_R and P_M pairs of current stego pixel respectively and also extract two more secret pairs (i.e, P_4 & P_5) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore ten secret data bits from the two stego pixels.

Case 3:
If E_L is "11" and E_R is "01" of stego pixel, then extract first, second and fourth secret data pairs (i.e., P_1, P_2 & P_4) from P_L, P_R and P_O pairs of current stego pixel respectively and also extract two more secret pairs (i.e, P_3 & P_5) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore ten secret data bits from the two stego pixels.

Case 4:
If E_L is "11" and E_R is "00" of stego pixel, then extract first and second secret data pairs (i.e., P_1 & P_2) from P_L and P_R pairs of current stego pixel respectively. This case will restore four secret data bits from single stego pixel.

Case 5:
If E_L is "10" and E_R is "11" of stego pixel, then extract first, third and fourth secret data pairs (i.e., P_1, P_3 & P_4) from P_L, P_M and P_O pairs of current stego pixel respectively and also extract two more secret pairs (i.e, P_2 & P_5) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore ten secret data bits from the two stego pixels.

Case 6:
If E_L is "10" and E_R is "10" of stego pixel, then extract first and third secret data pairs (i.e., P_1 & P_3) from P_L and P_M pairs of current stego pixel respectively and also extract two more secret pairs (i.e, P_2 & P_4) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore eight secret data bits from the two stego pixels.

Case 7:
If E_L is "10" and E_R is "01" of stego pixel, then extract first and fourth secret data pairs (i.e., P_1 & P_4) from P_L and P_O pairs of current stego pixel respectively and also extract two more secret pairs (i.e, P_2 & P_3) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore eight secret data bits from the two stego pixels.

Case 8:
If E_L is "10" and E_R is "00" of stego pixel, then extract first secret data pair (i.e., P_1) from P_L pair of current stego pixel. This case will restore two secret data bits from the single stego pixel.

Case 9:
If E_L is "01" and E_R is "11" of stego pixel, then extract second, third and fourth secret data pairs (i.e., P_2, P_3 & P_4) from P_R, P_M and P_O pairs of current stego pixel respectively and also

extract two more secret pairs (i.e, P_1 & P_5) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore ten secret data bits from the two stego pixels.

Case 10:
If E_L is "01" and E_R is "10" of stego pixel, then extract second and third secret data pairs (i.e., P_2 & P_3) from P_R and P_M pairs of current stego pixel respectively and also extract two more secret pairs (i.e, P_1 & P_4) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore eight secret data bits from the two stego pixels.

Case 11:
If E_L is "01" and E_R is "01" of stego pixel, then extract second and fourth secret data pairs (i.e., P_2 & P_4) from P_R and P_O pairs of current stego pixel respectively and also extract two more secret pairs (i.e, P_1 & P_3) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore eight secret data bits from the two stego pixels.

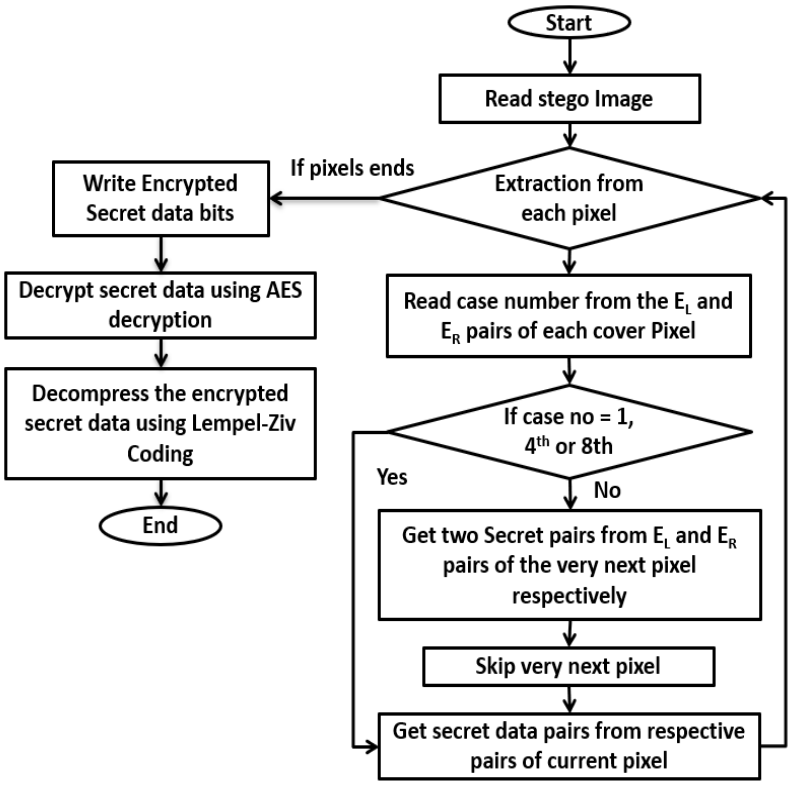


Fig. 2. Extraction Process of CEDE

Case 12:
If E_L is "01" and E_R is "00" of stego pixel, then extract second secret data pairs (i.e., P_2) from P_R pair of current stego pixel respectively and also extract two more secret pairs (i.e, P_1 & P_3) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore six secret data bits from the two stego pixels.

Case 13:
If E_L is "00" and E_R is "11" of stego pixel, then extract third and fourth secret data pairs (i.e., P_3 & P_4) from P_M and P_O pairs of current stego pixel respectively and also extract two more secret pairs (i.e, P_1 & P_2) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore eight secret data bits from the two stego pixels.

Case 14:
If E_L is "00" and E_R is "10" of stego pixel, then extract third secret data pairs (i.e., P_3) from P_M pair of current stego pixel respectively and also extract two more secret pairs (i.e, P_1 & P_2) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore six secret data bits from the two stego pixels.

Case 15:
If E_L is "00" and E_R is "01" of stego pixel, then extract two secret pairs (i.e, P_1 & P_2) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore four secret data bits from the two stego pixels.

Case 16:
If E_L is "00" and E_R is "00" of stego pixel, then extract two secret pairs (i.e, P_1 & P_2) from E_L and E_R pairs from the imminent stego pixel respectively. This case will restore four secret data bits from the two stego pixels.

Table 2. Proposed Extraction Steps

Input	Stego Image (S), AES Key (K)
Output	Secret Information (I)
Step 1	Convert bits of each stego pixel into pairs.
(a)	Assign 8^{th} and 7^{th} bits to Left Pair ($P_{L,\ Pixel}$)
(b)	Assign 6^{th} and 5^{th} bits to Right Pair ($P_{R,\ Pixel}$)
(c)	Assign 7^{th} and 6^{th} bits to Middle Pair ($P_{M,\ Pixel}$)
(d)	Assign 8^{th} and 5^{th} bits to Outer pair ($P_{O,\ Pixel}$)
(e)	Assign 4^{th} and 3^{rd} bits to Embedding Left Pair ($E_{L,\ Pixel}$)
(f)	Assign 2^{nd} and 1^{st} bits to Embedding Right Pair ($E_{R,\ Pixel}$)
Step 2	Extraction Process: Compute Stego image (S) and read four LSBs of each pixel.
(a)	If $E_{L,1}$=11 and $E_{R,1}$=11 then extract secret pairs, P_1 from $P_{L,1,}$ P_2 from $P_{R,1}$, P_3 from $P_{M,1}$ and P_4 from $P_{O,1}$.
(b)	If $E_{L,1}$=11 and $E_{R,1}$=10 then extract secret pairs, P_1 from $P_{L,1,}$ P_2 from $P_{R,1}$, P_3 from $P_{M,1}$, P_4 from $E_{L,2}$ and P_5 from $E_{R,2.}$
(c)	If $E_{L,1}$=11 and $E_{R,1}$=01 then extract secret pairs, P_1 from $P_{L,1,}$ P_2 from $P_{R,1}$, P_3 from $E_{L,2}$, P_4 from $P_{O,1}$ and P_5 from $E_{R,2.}$
(d)	If $E_{L,1}$=11 and $E_{R,1}$=00 then extract secret pairs, P_1 from $P_{L,1,}$ P_2 from $P_{R,1}$, P_3 from $E_{L,2}$ and P_4 from $E_{R,2}$.
(e)	If $E_{L,1}$=10 and $E_{R,1}$=11 then extract secret pairs, P_1 from $P_{L,1}$, P_2 from $E_{L,2}$, P_3 from $P_{M,1}$, P_4 from $P_{O,1}$ and P_5 from $E_{R,2.}$
(f)	If $E_{L,1}$=10 and $E_{R,1}$=10 then extract secret pairs, P_1 from $P_{L,1,}$ P_2 from $E_{L,2}$, P_3 from $P_{M,1}$ and P_4 from $E_{R,2}$.

(g)	If $E_{L,1}$=10 and $E_{R,1}$=01 then extract secret pairs, P_1 from $P_{L,1,}$ P_2 from $E_{L,2}$, P_3 from $E_{R,2}$ and P_4 from $P_{O,1}$.
(h)	If $E_{L,1}$=10 and $E_{R,1}$=00 then extract secret pairs, P_1 from $P_{L,1}$.
(i)	If $E_{L,1}$=01 and $E_{R,1}$=11 then extract secret pairs, P_1 from $E_{L,2}$, P_2 from $P_{R,1}$, P_3 from $P_{M,1}$, P_4 from $P_{O,1}$ and P_5 from $E_{R,2.}$
(j)	If $E_{L,1}$=01 and $E_{R,1}$=10 then extract secret pairs, P_1 from $E_{L,2,}$ P_2 from $P_{R,1}$, P_3 from $P_{M,1}$ and P_4 from $E_{R,2}$.
(k)	If $E_{L,1}$=01 and $E_{R,1}$=01 then extract secret pairs, P_1 from $E_{L,2,}$ P_2 from $P_{R,1}$, P_3 from $E_{R,2}$ and P_4 from $P_{O,1}$.
(l)	If $E_{L,1}$=01 and $E_{R,1}$=00 then extract secret pairs, P_1 from $E_{L,2,}$ P_2 from $P_{R,1}$, P_3 from $E_{R,2}$.
(m)	If $E_{L,1}$=00 and $E_{R,1}$=11 then extract secret pairs, P_1 from $E_{L,2,}$ P_2 from $E_{R,2}$, P_3 from $P_{M,1}$ and P_4 from $P_{O,1}$.
(n)	If $E_{L,1}$=00 and $E_{R,1}$=10 then extract secret pairs, P_1 from $E_{L,2,}$ P_2 from $E_{R,2}$, P_3 from $P_{M,1}$.
(o)	If $E_{L,1}$=00 and $E_{R,1}$=01 then extract secret pairs, P_1 from $E_{L,2,}$ P_2 from $E_{R,2}$.
(p)	If $E_{L,1}$=00 and $E_{R,1}$=00 then extract secret pairs, P_1 from $E_{L,2,}$ P_2 from $E_{R,2}$.
Step 3	Write compressed encrypted Secret Information
(a)	Decrypt the compressed encrypted Secret Information using AES encryption with AES key (K)
(b)	Decompress the secret information using LZW coding.
(c)	Get Secret Information (I) in text file.

3.5 Embedding Example of CEDE Method

In the embedding example of CEDE, compressed and encrypted secret data bits are embedded in the cover pixels as shown in **Fig. 3**. The first four secret data pairs P_1, $P_{2,}$ P_3 and P_4 are 10, 10, 10 and 01 respectively and the pairs of first cover pixel $P_{L,1}$, $P_{R,1}$, $P_{M,1}$ and $P_{O,1}$ are 10, 10, 01 and 10 respectively.

To embed the secret data, the algorithm compares the secret pairs P_1, $P_{2,}$ P_3 and P_4 with four pairs $P_{L,1}$, $P_{R,1}$, $P_{M,1}$ and $P_{O,1}$ of the first cover pixel. Here, case 4 is matched because $P_1 = P_{L,1}$, $P_2 = P_{R,1}$, $P_3 \neq P_{M,1}$ and $P_4 \neq P_{O,1}$. Thus, the algorithm replaces pairs of first cover pixel $E_{L,1}$ with "11" and $E_{R,1}$ with "00". Case 4 carries two secret data pairs within a single pixel.

The next four secret data pairs to be embedded are P_3, $P_{4,}$ P_5 and P_6 and their values are 10, 01, 10 and 10 respectively. The pairs of the second cover pixel $P_{L,2}$, $P_{R,2}$, $P_{M,2}$ and $P_{O,2}$ are 11, 00, 10 and 10 respectively. When secret pairs P_3, $P_{4,}$ P_5 and P_6 are compared with four pairs $P_{L,2}$, $P_{R,2}$, $P_{M,2}$ and $P_{O,2}$ of second cover pixel, case 13 is matched because $P_3 \neq P_{L,2}$, $P_4 \neq P_{R,2}$, $P_5 = P_{M,2}$ and $P_6 = P_{O,2}$. Thus, the algorithm replaces pairs of second cover pixel $E_{L,2}$ with "00" and $E_{R,2}$ with "11" and also replaces pairs of third cover pixel $E_{L,3}$ with "P_3" and $E_{R,3}$ with "P_4". The case 13 carries four secret data pairs in two pixels.

The next four secret data pairs P_7, $P_{8,}$ P_9 and P_{10} are 01, 10, 11 and 00 respectively. The pairs of fourth cover pixel $P_{L,4}$, $P_{R,4}$, $P_{M,4}$ and $P_{O,4}$ are 01, 10, 11 and 00 respectively. The algorithm compares secret pairs P_7, $P_{8,}$ P_9 and P_{10} with four pairs $P_{L,4}$, $P_{R,4}$, $P_{M,4}$ and $P_{O,4}$ of the fourth cover pixel. Here, case 1 is matched because $P_7 = P_{L,4}$, $P_8 = P_{R,4}$, $P_9 = P_{M,4}$ and $P_{10} = P_{O,4}$. Thus, the algorithm replaces pairs of fourth cover pixel $E_{L,4}$ with "11" and $E_{R,4}$ with "11". Case 1 carries four secret data pairs within single pixel.

The next four secret data pairs P_{11}, P_{12}, P_{13} and P_{14} are 10, 00, 10 and 11 respectively. The pairs of fifth cover pixel $P_{L,5}$, $P_{R,5}$, $P_{M,5}$ and $P_{O,5}$ are 01, 11, 11 and 01 respectively. The algorithm compares secret pairs P_{11}, P_{12}, P_{13} and P_{14} with four pairs $P_{L,5}$, $P_{R,5}$, $P_{M,5}$ and $P_{O,5}$ of the fifth cover pixel. For this case, case 16 is matched because $P_{11} \neq P_{L,5}$, $P_{12} \neq P_{R,5}$, $P_{13} \neq P_{M,5}$ and $P_{14} \neq P_{O,5}$. Thus, the algorithm replaces pairs of the fifth cover pixel $E_{L,5}$ with "00" and $E_{R,5}$ with "00" and also replaces pairs of sixth cover pixel $E_{L,6}$ with "P_{11}" and $E_{R,6}$ with "P_{12}". Case 16 carries two secret data pairs in two pixels.

Then, the four secret data pairs P_{13}, P_{14}, P_{15} and P_{16} are 10, 11, 01 and 10 respectively and the pairs of seventh cover pixel $P_{L,7}$, $P_{R,7}$, $P_{M,7}$ and $P_{O,7}$ are 10, 10, 01 and 10 respectively. The algorithm compares secret pairs P_{13}, P_{14}, P_{15} and P_{16} with four pairs $P_{L,7}$, $P_{R,7}$, $P_{M,7}$ and $P_{O,7}$ of the seventh cover pixel and finds that case 5 is matched because $P_{13} = P_{L,7}$, $P_{14} \neq P_{R,7}$, $P_{15} = P_{M,7}$ and $P_{16} = P_{O,7}$. Thus, it replaces pairs of seventh cover pixel $E_{L,7}$ with "10" and $E_{R,7}$ with "11" and also replaces pairs of eighth cover pixel $E_{L,8}$ with "P_{14}" and $E_{R,8}$ with "P_{17}". Case 5 carries five secret data pairs in two pixels.

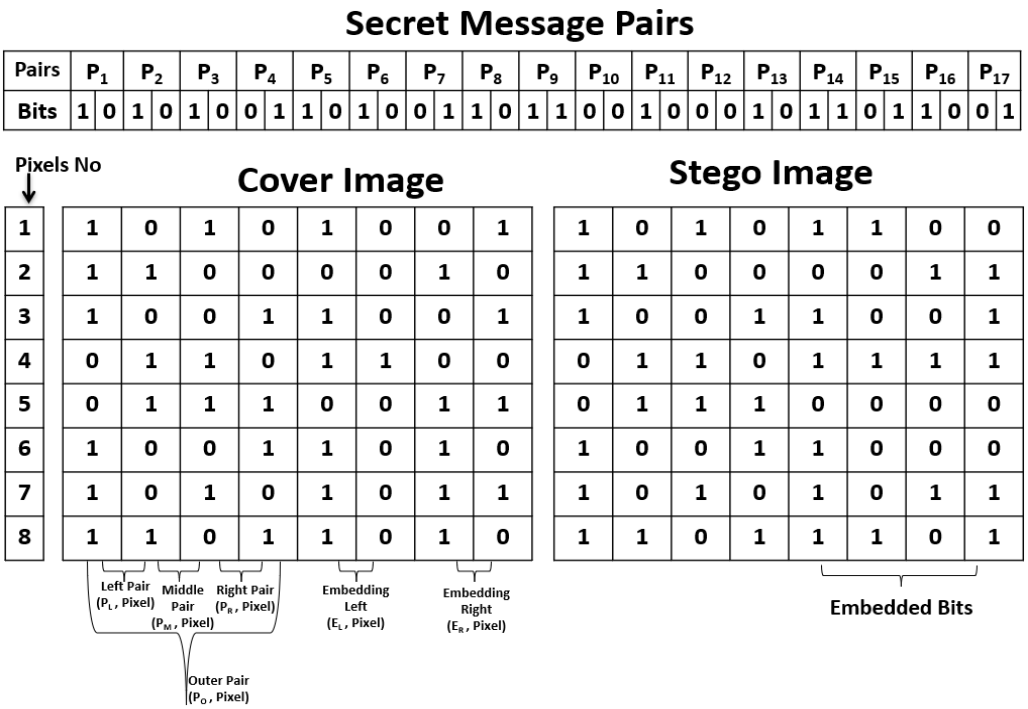


Fig. 3. Example of embedding phase of CEDE

3.6 Extraction Example of CEDE Method

In the extraction example of CEDE, compressed and encrypted secret data bits are extracted from stego pixels as shown in **Fig. 4**. The algorithm reads the values of E_L and E_R pairs of each stego pixel and extracts secret data pairs according to the case number.

The $E_{L,1}$ and $E_{R,1}$ pair of the first stego pixel are "11" and "00". Since case 4 is matched, it extracts first two secret data pairs P_1 and P_2 from $P_{L,1}$ and $P_{R,1}$ pair of the first stego pixel respectively.

The $E_{L,2}$ and $E_{R,2}$ pair of the second stego pixel are "00" and "11". Since case 13 is matched it extracts two secret data pairs P_5 and P_6 from $P_{M,2}$ and $P_{O,2}$ pair of the second stego pixel respectively and extracts two more secret data pairs P_3 and P_4 from $E_{L,3}$ and $E_{R,3}$ pair of third stego pixel respectively.

The $E_{L,4}$ and $E_{R,4}$ pair of the fourth stego pixel are "11" and "11". Since case 1 is matched, it extracts four secret data pairs P_7, P_8, P_9 and P_{10} from $P_{L,4}$, $P_{R,4}$, $P_{M,4}$ and $P_{O,4}$ pair of the fourth stego pixel respectively.

The $E_{L,5}$ and $E_{R,5}$ pair of the fifth stego pixel are "00" and "00". Since case 16 is matched it extracts two secret data pairs P_{11} and P_{12} from $E_{L,6}$ and $E_{R,6}$ pair of the sixth stego pixel respectively. The $E_{L,7}$ and $E_{R,7}$ pair of seventh stego pixel are "10" and "11". Since case 5 is matched, it extracts three secret data pairs P_{13}, P_{15} and P_{16} from $P_{L,1}$, $P_{M,1}$ and $P_{O,1}$ pair of the seventh stego pixel respectively and extract two more secret data pairs P_{14} and P_{17} from $E_{L,8}$ and $E_{R,8}$ pair of the eighth stego pixel respectively.

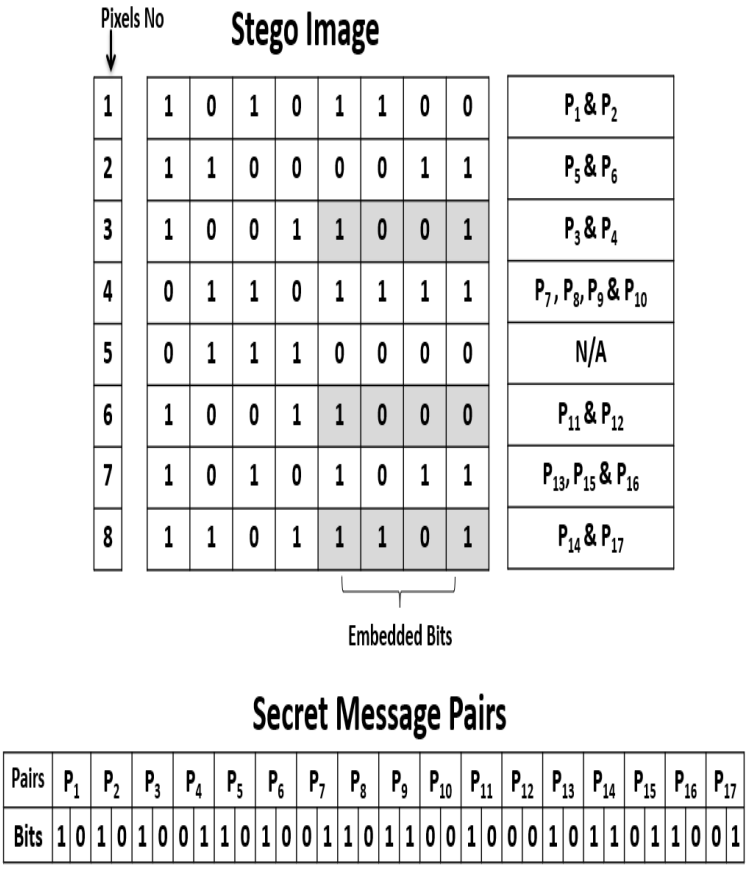


Fig. 4. Example of extraction phase of CEDE

4. Experimental Results

In this section, the numerical results of the proposed CEDE technique are shown. The gray level images of size 512×512 are used to implement and evaluate the proposed technique. We used images of Baboon, Lenna, Jet, Boat, Couple, and Peppers as cover images. These images

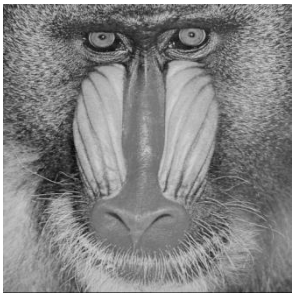
are shown in **Fig. 5**. The stego images as results of the proposed CEDE technique are shown in **Fig. 6**. As it can be noticed from **Fig. 6** embedding of secret data pairs in cover pixels are undetectable by human visual system (HVS).

The results of the proposed CEDE technique are evaluated by using parameters i.e., payload, bits per pixel (bpp), PSNR and MSE. The mathematical formulas of these parameters are as follows:

$$\text{bpp} = \frac{\text{Embedding capacity}}{\text{W} \times \text{H}} \tag{1}$$

$$\text{PSNR} = 10log^{10}\frac{255^2}{MSE} \tag{2}$$

$$\text{MSE} = \sum_{i=1}^{\text{W}\times\text{H}} \frac{(P'i - Pi)^2}{\text{W} \times \text{H}} \tag{3}$$



(a) Baboon Gray

(b) Lenna Gray

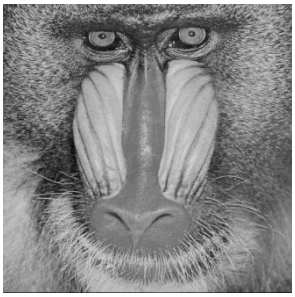
(c) Jet Gray

(d) Boat Gray

(e) Couple Gray

(f) Peppers gray

Fig. 5. Cover Images



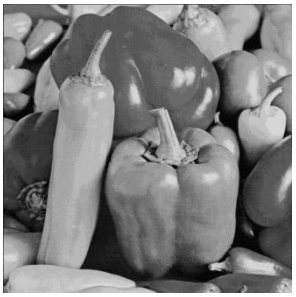
(a) Stego Baboon (PSNR= 33.82)

(b) Stego Lenna (PSNR= 33.36)

(c) Stego Jet (PSNR= 33.84)

(d) Stego Boat (PSNR= 33.52)

(e) Stego Couple (PSNR= 33.25)



(f) Stego Peppers (PSNR= 33.44)

Fig. 6. Stego Images

Table 3 shows that the proposed technique hides 3.01, 3.22, 3.58, 3.27, 3.25 and 3.26 bits/pixel in cover images of Baboon, Lenna, Jet, Boat, Couple and Peppers respectively with an acceptable average 33.53 PSNR. The results of CEDE technique show that proposed technique hides high payload of secret information in cover images with acceptable PSNR and MSE.

Table 3. Performance of Proposed CEDE Method without Compression

	Payload	**Bits Per Pixels (bpp)**	**PSNR**	**MSE**
Baboon	789,270	3.01	33.82	26.98
Lenna	845,972	3.22	33.36	29.99
Jet	939,592	3.58	33.84	26.85
Boat	858,580	3.27	33.52	28.91
Couple	819456	3.25	33.25	30.76
Peppers	854,796	3.26	33.44	29.44
Average	851,277	3.26	33.53	28.84

Table 4 shows that the proposed technique hides 9.03, 9.66, 10.74, 9.81, 9.75 and 9.78 compressed and encrypted bits/pixel in cover images of Baboon, Lenna, Jet, Boat, Couple and Peppers respectively with an acceptable average 33.53 PSNR. The results of CEDE technique show that proposed technique hides high payload of secret information in cover images with acceptable PSNR and MSE.

Table 4. Performance of Proposed CEDE Method with Compression

	Payload	Bits Per Pixels (bpp)	PSNR	MSE
Baboon	2,367,810	9.03	33.82	26.98
Lenna	2,537,916	9.66	33.36	29.99
Jet	2,818,776	10.74	33.84	26.85
Boat	2,575,740	9.81	33.52	28.91
Couple	2,458,368	9.75	33.25	30.76
Peppers	2,564,388	9.78	33.44	29.44
Average	2,553,833	9.79	33.53	28.84

5. Comparison With Existing Techniques

The results of the proposed technique without compression are compared with available existing techniques based on some evaluation parameters: Payload, Bits per pixel (bpp), PSNR and Mean square error (MSE). The comparison of images i.e., Baboon, Lenna, Jet, Boat, Couple and Peppers are shown in **Table 5-10** respectively. The huge modification in LSBs of cover images to embed more secret data causes low quality of stego image. Thus, many existing techniques embed high payload of secret data with low PSNR and MSE and vice versa. The comparison with existing technique shows that proposed CEDE technique embed more secret data in cover image with acceptable stego image quality.

Table 5 shows a comparison between proposed technique and existing LSB steganography techniques for Baboon stego image. This comparison shows that proposed CEDE technique embed 789270 secret data bits in 262,144 pixels of Baboon cover image with 33.82 PSNR and 26.98 MSE. This technique embeds more secret data with acceptable PSNR as compare to existing techniques.

Table 5. Baboon Stego Gray

	Payload	Bits Per Pixels (bpp)	PSNR	MSE
Shehzad et al. [8]	65,536	0.25	57.19	0.12
Shehzad et al. [10]	524,288	2,00	34.92	21.1
Liu et al. [13]	262,144	1.00	52.17	0.39
Lu et al. [14]	778,567	2.97	37.85	10.66
Khan et al. [18]	410,636	1.56	44.37	2.40
Khalind et al. [21]	589,824	2.25	37.54	11.45

Hong et al. [23]	262,144	1.00	26.98	130.34
Setiadi et al. [24]	314,572	1.20	46.57	1.43
Muhammad et al. [27]	87,381	0.33	34.35	23.88
Muhammad et al. [28]	21,845	0.08	51.89	0.42
Kuo et al. [29]	221,987	0.8	44.62	2.24
Mohamed et al. [30]	838,860	3.10	38.85	8.40
Goljan et al. [35]	1,024	0.003	30.0	65.02
Xuan et al. [36]	85507	0.32	36.60	14.22
Celik et al. [37]	74600	0.26	39.26	7.71
Yang et al. [38]	785572	2.9	39.16	7.89
CEDE	789,270	3.01	33.82	26.98

Table 6 shows a comparison for Lenna stego image. This comparison shows that proposed CEDE technique embeds 845,972 secret data bits in 262,144 pixels of Lenna cover image with 33.36 PSNR and 26.98 MSE. This technique embeds more secret data with acceptable PSNR as compare to existing techniques but high MSE (29.99) is limitation of this technique.

Table 6. Lenna Stego Gray

	Payload	**Bits Per Pixels (bpp)**	**PSNR**	**MSE**
Shehzad et al. [8]	65,536	0.25	56.98	0.12
Shehzad et al. [10]	524,288	2.00	35.19	19.52
Swain et al. [11]	812,566	3.09	41.40	4.70
Khodaei et al. [12]	839,028	3.20	37.67	11.11
Liu et al. [13]	262,144	1.00	52.03	0.40
Lu et al. [14]	778,567	2.97	38.09	10.09
Khan et al. [18]	433,224	1.65	43.71	2.79
Liao et al. [19]	810,564	3.09	39.57	7.17
Khalind et al. [21]	589,824	2.25	37.56	11.40
Setiadi et al. [24]	288,358	1.10	48.08	1.01
Muhammad et al. [27]	87,381	0.33	35.60	17.90
Muhammad et al. [28]	21,845	0.08	42.62	3.55
Kuo et al. [29]	222,121	0.8	44.53	2.29
Mohamed et al. [30]	589,824	2.25	44.43	2.34
Leng et al. [31]	828,375	3.16	33.58	28.51
Goljan et al. [35]	1,024	0.003	30.0	65.02
Xuan et al. [36]	85507	0.32	36.60	14.22

Celik et al. [37]	74600	0.26	38.00	10.30
Yang et al. [38]	757332	2.88	39.31	7.62
CEDE	845,972	3.22	33.36	29.99

Table 7 shows a comparison for Jet stego image. This comparison shows that proposed CEDE technique embeds 939,592 secret data bits in 262,144 pixels of Jet cover image with 33.84 PSNR and 26.85 MSE. This technique embeds high payload of secret data than other techniques.

Table 7. Jet Stego Image

	Payload	**Bits Per Pixels (bpp)**	**PSNR**	**MSE**
Swain et al. [11]	814497	3.10	42.51	3.64
Liu et al. [13]	262,144	1.00	52.17	0.39
Lu et al. [14]	778,567	2.97	38.15	9.95
Bai et al. [16]	1,024,983	3.19	33.84	26.85
Hussain et al. [17]	824756	3.15	37.79	10.81
Khalind et al. [21]	589824	2.25	37.68	11.09
Hong et al. [23]	262,144	1.00	31.97	41.31
Setiadi et al. [24]	288,358	1.10	47.82	1.07
Muhammad et al. [27]	87381	0.33	35.90	16.71
Muhammad et al. [28]	21845	0.08	51.92	0.41
Kuo et al. [29]	238872	0.9	46.93	1.31
Leng et al. [31]	825753	3.15	33.60	28.38
CEDE	939,592	3.58	33.84	26.85

Table 8 shows a comparison for Boat stego image. This comparison shows that proposed CEDE technique embeds 858,580 secret data bits in 262,144 pixels of Boat cover image with 33.52 PSNR and 28.91 MSE. This technique embeds more secret data with acceptable PSNR as compare to existing techniques but high MSE (28.91) is limitation of this technique.

Table 8. Boat Stego Image

	Payload	**Bits Per Pixels (bpp)**	**PSNR**	**MSE**
Zakaria et al. [9]	833,482	3.28	42.35	3.78
Swain et al. [11]	846,516	3.22	36.66	14.03
Liu et al. [13]	262,144	1.00	52.12	0.39
Lu et al. [14]	778,567	2.97	38.38	10.30

Hussain et al. [17]	810,735	3.09	37.62	11.24
Khalind et al. [21]	589,824	2.25	37.48	11.61
Hong et al. [23]	262,144	1.00	31.97	41.31
Setiadi et al. [24]	288,358	1.10	48.51	0.91
Muhammad et al. [27]	87,381	0.33	35.66	17.66
Kuo et al. [29]	222,589	0.84	44.84	2.13
Leng et al. [31]	812,646	3.10	32.47	36.81
Yang et al. [38]	753832	2.87	39.20	7.81
CEDE	858,580	3.27	33.52	28.91

Table 9 shows comparison for Couple stego image. This comparison shows that proposed CEDE technique embeds 819,456 secret data bits in 262,144 pixels of Boat cover image with 33.25 PSNR and 30.76 MSE. This technique embeds high payload of secret data than other techniques.

Table 9. Couple Stego Image

	Payload	**Bits Per Pixels (bpp)**	**PSNR**	**MSE**
Zakaria et al. [9]	831,462	3.17	42.66	3.56
Liu et al. [13]	262,144	1.00	52.16	0.39
Lu et al. [14]	778,567	2.97	38.02	10.25
Bai et al. [16]	794,296	3.03	36.30	15.24
Hussain et al. [17]	815,631	3.11	37.62	11.24
Khalind et al. [21]	589,824	2.25	37.50	11.56
Hong et al. [23]	262,144	1.00	34.73	21.88
Setiadi et al. [24]	296,222	1.03	47.98	1.03
Muhammad et al. [27]	87,381	0.33	33.89	26.55
Kuo et al. [29]	221,976	0.84	44.62	2.24
Leng et al. [31]	828,375	3.16	33.40	29.72
CEDE	819,456	3.25	33.25	30.76

Table 10 shows a comparison for Peppers stego image. This comparison shows that proposed CEDE technique embeds 854,796 secret data bits in 262,144 pixels of Boat cover image with 33.44 PSNR and 29.44 MSE. This technique embeds high payload of secret data than other techniques.

Table 10. Peppers Stego Image

	Payload	**Bits Per Pixels (bpp)**	**PSNR**	**MSE**
Swain et al. [11]	815,912	3.11	38.33	9.61
Khodaei et al. [12]	822,042	3.13	37.13	12.59
Liu et al. [13]	262,144	1.00	52.12	0.39
Lu et al. [14]	778,567	2.97	38.08	10.89
Bai et al. [16]	810,024	3.05	37.25	12.24
Hussain et al. [17]	810,501	3.09	38.56	9.05
Liao et al. [19]	805,492	3.07	39.79	10.81
Khalind et al. [21]	589,824	2.25	37.50	11.56
Hong et al. [23]	262,144	1.00	33.42	29.58
Setiadi et al. [24]	285,736	1.09	48.24	0.97
Muhammad et al. [27]	87,381	0.33	34.76	21.73
Muhammad et al. [28]	21,845	0.08	51.99	0.41
Kuo et al. [29]	222,054	0.8	44.57	2.27
Mohamed et al. [30]	589,824	2..25	44.40	2.36
Leng et al. [31]	828,375	3.16	33.61	28.31
Yang et al. [38]	786,016	2.99	39.06	8.07
CEDE	854,796	3.26	33.44	29.44

6. Conclusion

In this paper a novel hybrid technique for optimal data security is proposed; where data compression is done using LZW compression in the first step; followed by the application of AES cryptographic technique; lastly image steganography using LSB substitution is devised which is based on pattern matching mechanism. In the steganographic technique, four 2 bits pairs of secret information are compared with four pairs of cover pixel and sixteen possible cases of matching are generated. The four LSBs of each pixel are modified according to pairs matching number. The best case of this technique carries eight secret data bits within a single pixel, the average case carries eight or ten secret data bits within 2 pixels and the worst case carries four secret data bits within two pixels. The results of proposed method show that stego images carry high capacity of payload with good quality. This hybrid method provides maximal data transmission inside cover images along with ensuring optimum data security.

References

[1] T. Saha, S. Sengupta, and T. Dasgupta, "Chaotic cipher based spatial domain steganography with strong resistance against statistical attacks," in *Proc. of the 3rd International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 365-370, 2017. Article (CrossRef Link)

[2] 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. 335, pp. 299-326, 2019. Article (CrossRef Link)

[3] M. Douglas, K. Bailey, M. Leeney, and K. Curran, "An overview of steganography techniques applied to the protection of biometric data," *Multimedia Tools Applications*, vol. 77, no. 13, pp. 17333-17373, 2017. Article (CrossRef Link)

[4] M. Nosrati and R. Karimi, "Investigating a Benchmark Cloud Media Resource Allocation and Optimization," *World Applied Programming*, vol. 6, no. 1, pp. 5-9, 2016. Article (CrossRef Link)

[5] M. Sravanthi, M. Devi, S. Riyazoddin, and M. Reddy, "A Spatial Domain Image Steganography Technique Based on Plane Bit Substitution Method," *Global Journals*, vol. 12, no. 15, 2012. Article (CrossRef Link)

[6] C. Hosmer, "Discovering Hidden Evidence," *Journal of Digitital Forensic Practice*, vol. 1, no. 1, pp. 47-56, 2006. Article (CrossRef Link)

[7] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," in *Proc. of Workshop on Multimedia and Security New Challenges*, 2001. Article (CrossRef Link)

[8] D. Shehzad and T. Dag, "LSB Image Steganography Based on Blocks Matrix Determinant Method," *KSII Transactions Internet Information Systems*, vol. 13, no. 7, pp. 3778-3793, 2019. Article (CrossRef Link)

[9] A. Zakaria, M. Hussain, A. Wahab, M. Idris, N. Abdullah, and K.H. Jung, "High-Capacity Image Steganography with Minimum Modified Bits Based on Data Mapping and LSB Substitution," *Applied Science*, vol. 8, no. 11, 2018. Article (CrossRef Link)

[10] D. Shehzad and T. Dag, "A novel image steganography technique based on similarity of bits pairs," in *Proc. of IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)*, pp. 99-104, 2017. Article (CrossRef Link)

[11] G. Swain, "A Steganographic Method Combining LSB Substitution and PVD in a Block," *Procedia Computer Science*, vol. 85, pp. 39-44, 2016. Article (CrossRef Link)

[12] M. Khodaei, B. S. Bigham, and K. Faez, "Adaptive Data Hiding, Using Pixel-Value-Differencing and LSB Substitution," *Cybernetics and Systems*, vol. 47, no. 8, pp. 617-628, 2016. Article (CrossRef Link)

[13] Y. Liu, C. C. Chang, and T. Y. Chien, "A Revisit to LSB Substitution Based Data Hiding for Embedding More Information," *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, vol. 63, pp. 11-19, 2016. Article (CrossRef Link)

[14] T. C. Lu and Y. C. Lu, "An Improved Data Hiding Method of Five Pixel Pair Differencing and LSB Substitution Hiding Scheme," *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, pp. 67-74, 2016. Article (CrossRef Link)

[15] J. H. Horng, C. C. Chang, and G. L. Li, "Steganography Using Quotient Value Differencing and LSB Substitution for AMBTC Compressed Images," *IEEE Access*, vol. 8, pp. 129347-129358, 2020. Article (CrossRef Link)

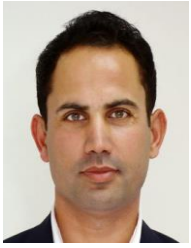
[16] J. Bai, C. C. Chang, T. S. Nguyen, C. Zhu, and Y. Liu, "A high payload steganographic algorithm based on edge detection," *Displays*, vol. 46, pp. 42-51, 2017. Article (CrossRef Link)

[17] M. Hussain, A. W. A. Wahab, N. Javed, and K. H. Jung, "Recursive Information Hiding Scheme Through LSB, PVD Shift, and MPE," *IETE Technical Review*, vol. 35, no. 1, pp. 53-63, 2017. Article (CrossRef Link)

[18] A. U. Islam, F. Khalid, M. Shad, Z. Khan, T. Mahmood, A. Khan, U. Ali, and M. Naeem, "An improved image steganography technique based on MSB using bit differencing," in *Proc. of the 6th International Conference on Innovative Computing Technology (INTECH)*, pp. 265-269, 2016. Article (CrossRef Link)
[19] X. Liao, Q. Wen, and J. Zhang, "A steganographic method for digital images with four-pixel differencing and modified LSB substitution," *Jouranl of Visual Communication Image Representation*, vol. 22, no. 1, pp. 1-8, 2011. Article (CrossRef Link)
[20] K. H. Jung and K. Y. Yoo, "Steganographic method based on interpolation and LSB substitution of digital images," *Multimedia Tools and Applications*, vol. 74, no. 6, pp. 2143-2155, 2014. Article (CrossRef Link)
[21] O. Khalind and B. Aziz, "Single-mismatch 2LSB embedding steganography," in *Proc. of IEEE International Symposium on Signal Processing and Information Technology*, pp. 283-286, 2013. Article (CrossRef Link)
[22] H. Lee, "Data Hiding in Spatial Color Images on Smartphones by Adaptive R-G-B LSB Replacement," *IEICE Transactions on Information Systems*, vol. E101.D, no. 8, pp. 2163-2167, 2018. Article (CrossRef Link)
[23] W. Hong, "Efficient Data Hiding Based on Block Truncation Coding Using Pixel Pair Matching Technique," *Symmetry (Basel)*, vol. 10, no. 2, 2018. Article (CrossRef Link)
[24] D. R. I. M. Setiadi and J. Jumanto, "An Enhanced LSB-Image Steganography Using the Hybrid Canny-Sobel Edge Detection," *Cybernetics and Information Technologies*, vol. 18, no. 2, pp. 74-88, 2018. Article (CrossRef Link)
[25] R. D. Rashid and T. F. Majeed, "Edge based image steganography: Problems and solution," in *Proc. of the 3rd International Conference Communications*, pp. 1-5, 2019. Article (CrossRef Link)
[26] S. L. Chikouche and N. Chikouche, "An improved approach for lsb-based image steganography using AES algorithm," in *Proc. of the 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*, pp. 1-6, 2017. Article (CrossRef Link)
[27] K. Muhammad, J. Ahmad, N. U. Rehman, Z. Jan, and M. Sajjad, "CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method," *Multimedia Tools Applications*, vol. 76, no. 6, pp. 8597-8626, 2016. Article (CrossRef Link)
[28] K. Muhammad, J. Ahmad, H. Farman, Z. Jan, M. Sajjad, and S. W. Baik, "A Secure Method for Color Image Steganography using Gray-Level Modification and Multi-level Encryption," *KSII Transactions Internet Information Systems*, vol. 9, no. 5, pp. 1938-1952, 2015. Article (CrossRef Link)
[29] W. C. Kuo, S. H. Kuo, and L. C. Wuu, "Multi-Bit Data Hiding Scheme for Compressing Secret Messages," *Applied Science*, vol. 5, no. 4, pp. 1033-1049, 2015. Article (CrossRef Link)
[30] M. H. Mohamed and L. M. Mohamed, "High Capacity Image Steganography Technique based on LSB Substitution Method," *Applied Mathematics and Information Science*, vol. 10, no. 1, pp. 259-266, 2016. Article (CrossRef Link)
[31] H. W. Tseng and H. S. Leng, "High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion," *IET Image Processing*, vol. 8, no. 11, pp. 647-654, 2014. Article (CrossRef Link)
[32] J. R. Jayapandiyan, C. Kavitha, and K. Sakthivel, "Enhanced Least Significant Bit Replacement Algorithm in Spatial Domain of Steganography Using Character Sequence Optimization," *IEEE Access*, vol. 8, pp. 136537-136545, 2020. Article (CrossRef Link)
[33] H. Patel, U. Itwala, R. Rana, and K. Dangarwala, "Survey of Lossless Data Compression Algorithms," *International Journal of Engineering Research and Technolgy*, vol. 4, no. 4, pp. 926-929, 2015. Article (CrossRef Link)
[34] S. K. Rao, D. Mahto, and D. A. Khan, "A Survey on Advanced Encryption Standard," *International Jouranl of Science and Research*, vol. 6, no. 1, pp. 711-724, 2017. Article (CrossRef Link)
[35] M. Goljan, J. J. Fridrich, and R. Du, "Distortion-Free Data Embedding for Images," in *Proc. of International Workshop on Information Hiding*, vol. 2137, pp. 27-41, 2001. Article (CrossRef Link)

[36] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," *Electronic Letters*, vol. 38, no. 25, pp. 1646-1648, 2002. Article (CrossRef Link)
[37] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in *Proc. of International Conference on Image Processing*, 2002. Article (CrossRef Link)
[38] H. Yang, X. Sun, and G. Sun, "A high-capacity image data hiding scheme using adaptive LSB substitution," *Radioengineering*, vol. 18, no. 4, pp. 509-516, 2009. Article (CrossRef Link)

Ali Hamza received his BS degree in computer science from the Government College of Faisalabad and MS in computer science from the National University of Computer and Emerging Sciences, Pakistan. His area of research interest is Information security and Image processing specializing in data hiding, steganography and cryptography. He is also working as a reviewer in IEEE access.



Danish Shehzad is working as an Assistant Professor in Computer Science department at FAST-National University of Computers & Emerging Sciences (NUCES), Chiniot-Faisalabad campus. He has teaching and research experience of over 7 years. Before joining NUCES he worked as a researcher on the Brain-Inspired Run-time system for a very large scale brain simulation (BiRTS) project funded by the Scientific and Technological Research Council of Turkey. He also worked as a senior researcher on a joint venture between Kadir Has University and Selcuk University, Turkey for the Automatic Detection of various diseases through medical image processing. He received a BS degree from COMSATS University in 2010, an MS degree in computer sciences from Hazara University, Pakistan in 2014, and a Ph.D. degree from Kadir Has University, Turkey in 2019. His areas of research interests include computer networks, information security and cryptography.

Muhammad Shahzad Sarfraz has been an active player in Computer Science and Geospatial Technologies. He leads the core thematic areas of ICT tools for Health, Agriculture, Disaster management, Location-based systems & their applications for the Environment. His focus areas are the use of web-based and open-source tools related to Remote Sensing & Geographical Information systems. Dr. Shahzad has contributed to 30 journal and conference papers, along with developing tools for organizational needs and readiness assessment, and evaluation of eHealth initiatives.

Usman Habib is working in the capacity of Assistant Professor at the department of computer science, FAST-National University of Computers & Emerging Sciences (NUCES), Chiniot-Faisalabad campus. He holds more than ten years of teaching and research experience spanning from 2006 to date. Before joining FAST-NUCES, he served as Assistant Professor in COMSATS Institute of Information Technology, Abbottabad, Pakistan. In addition to teaching and research, he also successfully completed various industrial projects which include “eXtract” project funded by an Austrian Funding Agency under e!MISSION programme. Dr. Usman regularly publishes his research in conferences and journals of repute. He was program chair of International Conference on emerging technologies (ICET), 2019 and remained active member of organizing committee for a series of international conferences, Frontiers of Information Technology (FIT) for a number of years.

Numan Shafi received a B.S. degree in software engineering from Punjab University College of Information Technology and an MPhil in computer science from Punjab University College of Information Technology, Pakistan. He is serving FAST National University of Computer and Emerging Sciences (NUCES) as a Lecturer. He has expertise in ICT tools for Health, Agriculture, Disaster management, Location-based systems & its applications for Environment. Moreover, he has experience in machine learning, data mining, data analytic, data science, unmanned aerial vehicles (UAV), and its applications.