International Journal on Emerging Technologies **10**(2): 29-38(2019)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

An Edge based Steganographic Approach using a two Level Security Scheme for Digital Image Processing and Analysis

Kalyan Kumar Jena[1,2,3], Sasmita Mishra[2], Sarojananda Mishra[2] and Sourav Kumar Bhoi[3]
[1]Department of Computer Science,
Utkal University, Bhubaneswar (Odisha) India
[2]Department of Computer Science Engineering and Applications,
Indira Gandhi Institute of Technology, Sarang (Odisha) India
[3]Department of Computer Science and Engineering,
Parala Maharaja Engineering College, Berhampur(Odisha) India

(Corresponding author: Sourav Kumar Bhoi)

(Received 09 April 2019, Revised 01 July 2019 Accepted 08 July 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: In digital communication, the information should be secured over the network. Steganography provides the solution to hide the secret data within the data. In this paper, a two level secured edge based image steganographic (2L-EBS) approach is proposed to hide secret information of variable size in a cover image. The information is first encrypted using the RSA algorithm to generate a ciphertext, afterwards ciphertext is again encoded using the Huffman coding to generate a two level encrypted data. The Huffman code (encoded data) is then placed at the least significant bits (LSBs) of the cover image edge pixels to generate a Stego image. The edge pixels are generated using Canny edge detection technique. According to the size of the encoded data, the number of edge pixels is selected for both grayscale image and RGB image. The pixels are selected based on the intensity of the pixels. The performance of 2L-EBS is evaluated using a 512×512 grayscale cover image and 992×497 RGB cover image. Results show that when a message of size 35 bits, 77 bits, and 322 bits are augmented in color and grayscale cover images, 2L-EBS performs better than Seeja, Sun, Luo, and Wu schemes in terms of peak signal to noise ratio (PSNR) and mean squared error (MSE).

Keywords: 2L-EBS, RSA, Huffman code, PSNR, MSE

I. INTRODUCTION

Steganography is an important security concept to hide the secret information in a cover media [1-9]. This ensures secret and reliable communication among the sender and the receiver [10]. In this paper, we have chosen the media as a cover image where the secret information is entrenched. The main characteristic of steganography is undetectable of the secret information in the cover media by a third party. Steganography technique is performed so well that the actual image information is lost, however it is undetectable. Two important concepts are mainly maintained while applying a stenographic technique such as undetectability and embedding capacity. Embedding capacity refers to the passel of secret data entrenched in the cover media, so that the data can be decoded from an undetectable media. Therefore, the methods should choose the data size (number of pixels) lower than the enormity of cover media (total bunch of pixels of the cover image). For example, if a secret message is of size S_M bits then the method should select Np number of pixels for data hiding, where Np = S_M and Np≤ (X×Y) (size of the image).Security is an important concern while sending the secret data in a cover media because this data may be intercepted or located by the attackers [1, 11, 12, 13]. Therefore, there should be development of secure algorithms to first encrypt the data to maintain the security attributes like confidentiality, integrity, and availability parameters. Then, after receiving the Stego media the receiver decodes the pixel positions and decrypts the encrypted information stored in pixels to get the correct or original message.

Nowadays, edges of the images are highly used for steganography [1, 14] Edge pixels have either high intensity or low intensity according to the neighbour pixels because coefficient of gradient suddenly changes. The edges have a characteristic that it is difficult to design. Therefore, edge pixels are selected by the researchers to store the secret data. We are mainly motivated from Islam *et al*. [1] method and Seeja *et al*. [15] method to work in this area. Islam *et al*. [1] proposed an edge based steganographic technique where two LSBs of a pixel are accustomed to accumulate the 2 bits of the secret information for data hiding. However, this method degrades the quality of the cover media because changing 2 bits changes the pixel value. For example, if secret information is 10110111 then the first two bits are 10. If a pixel possesses an intensity value of 160 then the bit information is 10100000. Then if two LSBs of the pixel are substituted with 10 then the pixel intensity changes to 162. Therefore, taking only single LSB changes 160 to 161. One more problem with this technique is that if 2 LSB values are decoded from the edge pixels then the third party knows the data. Therefore, this technique lacks encryption scheme which is vulnerable to different type of attacks. Therefore, Seeja *et al*. [15] proposed a two level security scheme by encoding the data initially, and afterwards embedding the data to the edge pixels at the LSBs. The secret data is embedded after a XOR operation is performed at the 1 bit LSB. This two level security scheme enhances the robustness of the Stego image. However, this current technique is a better method for hiding the information in a cover media.

Therefore, we have secured the message by a two level architecture by encrypting the data with RSA algorithm then encoding it again with the Huffman coding and then locating the secret data at the edge pixels. Here, placing the encrypted data at the edge pixels improves the surveillance of the covert message.

The main contribution in this paper is stated as follows:

1. A two level security scheme for edge based image steganography (2L- EBS) is proposed to ensure undetectability. The secret information is first encoded by the help of RSA algorithm. Afterwards, the encrypted data is again encoded using the Huffman coding. The Huffman code is then embedded at the edges of the cover media.
2. The number of edge pixels for data embedding is selected using the high intensity values of the pixels.
3. The Huffman code is embedded by replacing a single bit of Huffman code with the last bit LSB of the selected pixel.
4. Security analysis is performed to show the robustness of the scheme against several types of attacks.
5. The performance of 2L-EBS is evaluated using a 512×512 grayscale cover image and 992×497 RGB cover image. Results show that 2L- EBS carries out improved performance than Seeja *et al.* [15], Sun *et al*. [36], Luo *et al.* [34], and Wu *et al*. [21] schemes by focusing on PSNR and MSE.

The remaining of the paper is arranged as follows. Section 2 focuses on related works. Section 3 presents the methodology used for steganography which describes about the two level security architecture. Section 4 demonstrates the surveillance analysis of 2L-EBS. Section 5 focuses on results and discussion and Section 6 focuses on conclusion of the work.

II. RELATED WORKS

Steganography secures the communication between a sender and a receiver. Many such steganographic techniques are proposed which are normally focused on spatial as well as frequency domain [16, 17]. The conceal data is entrenched at the pixels of the cover media using the LSB replacement as well as matching mechanism in case of spatial domain [18, 19, 20, 21]. LSB replacement means the end bit of the pixel is substituted with the data bit. In case of LSB matching, if the last bit of the pixel does not equal with the second bit of the data then embedding occurs by pixel content, that may increase or decrease [22, 23]. In the transfer domain [24], the secret data embedding is performed using modification of non-zero coefficient DCT of a cover image. Various steganalysis tools are present for detecting the change in a Stego image or detecting a secret message at the Stego image. There are three types of tools such as structural, non structural and visual [25, 26]. Visual attacks can detect the distortion which is visible to the human eyes. Structural attacks can detect the change by using structure detectors such as histogram [27], sample pairing [28], RS method [29], and weighted Stego [30]. Non structural attacks detect the change by extracting the characteristics of cover media and Stego image, and then matching the features for secret data analysis. It uses detection such as subtraction pixel adjacency matrix and spatial rich model [31, 32]. It also use machine learning methods such as support vector machine (SVM) for learning the characteristics and classify images according to the changes [33]. In case of LSB Matching Revisited (LSBMR) method, two pixels are taken for implanting the secret information [19]. The LSB of the pixel is substituted with the first bit of the conceal message. The odd and even association of two pixels shows another bit of the conceal message. One third probability embedding is performed by taking three consecutive bits of the pixel, and it is based on LSBMR technique. LSBCSS (LSB Compatible Substitution) is an extension of LSBMR which that fewer change in the histogram. In 2 LSB method, the two bits of the conceal message are placed at the 2 LSBs of the pixel. Edge based techniques use noisy pixels for storing the secret data [34]. Edge Adaptive Image Steganography on the basis of LSBMR is proposed to select the implanting areas (pixels) in accordance with the conceal message size, and to find the variation among the two pixels at the cover image [35].

The above discussed methods ensure less security to the secret message. The researchers mainly focus on secret data embedding at the pixels. However, the methods lack security service to the secret message. If the message is encrypted and placed at the pixels then the attacker after intercepting the message is unable to decode the message (if it extracts the pixel positions at the cover image). Therefore the security of the conceal message is enhanced by encrypting the message twice with RSA and Huffman coding. The final encoded data is placed at the edge pixels which make the attacker difficult to locate the pixels. The secret message is only decoded by the genuine or authorized receiver by extracting the message from the pixels, decoding the Huffman code, and then decrypting the message using RSA algorithm.

III. METHODOLOGY

The proposed method deals with the following steps to generate a Stego image. This also consists of eradication of the encoded message from the Stego image and then decoding is accomplished to draw out the original hidden message. Fig. 1 and 2 show the 2-level security architecture of 2L-EBS and secret message extraction respectively. The steps used for designing the complete architecture are stated as follows:

1. Secret message encryption using RSA algorithm.
2. Generation of Huffman code for the RSA encrypted message.
3. Cover image edge pixel selection for embedding the encoded Huffman data.
4. Embedding encoded data at the selected pixels for generating the final Stego image.
5. Hidden message eradication from the Stego image.

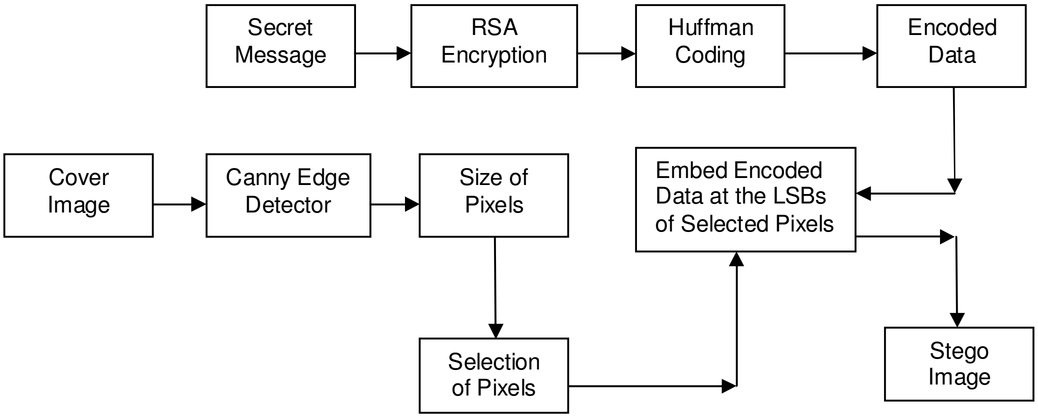


Fig. 1. Generation of Stego image using 2L-EBS scheme.

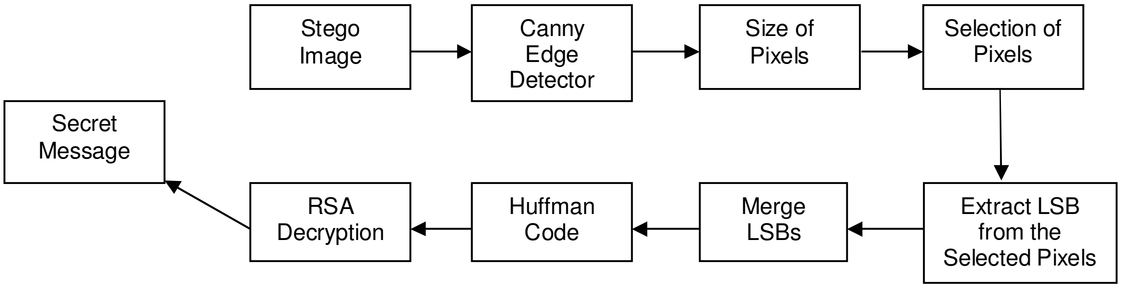


Fig. 2. Hidden message extraction from the Stego image utilizing 2L-EBS.

A. Secret Message Encryption Using RSA

RSA is an asymmetric key cryptographic algorithm that uses two keys for encoding and decoding purposes [15]. The encryption is carried out by encrypting the data utilizing the receiver public key and then the receiver decrypts the ciphered data utilizing the private key of itself. Let, a secret message S_M is encrypted to get the ciphertext C = $S^e{}_M$ mod n where e refers to the public key of the receiver and n is the product of the two prime numbers. The encrypted message is decrypted using P = C^d mod n, where P refers to the plaintext, d refers to the private key of the receiver. For example, suppose a message hello is encrypted then it has to first convert to ASCII. Then the data S_M is encrypted using the RSA algorithm. RSA algorithm is more secure because it is better algorithm to encrypt shorter messages, and it will be more applicable for steganography. Algorithm 1 shows the pseudocode for RSA encryption.

Algorithm 1: RSA Encryption of Secret Message S_M

```
Input: $S_M$
Output: C
Choose $a$ and $b$, where $a$ and $b$ are prime numbers
$t = p \times q$
$\Phi(t) = (a-1) \times (b-1)$
Choose integer $k$, where $1<k<\Phi(t)$ and k is co-prime to $\Phi(t)$
$d \times k \equiv 1 \bmod \Phi(t)$
$C = S^k{}_M \bmod t$   (Encryption)
```

B. Generation of Huffman Code Using RSA Encrypted Data

The encrypted message in the decimal form is transformed to the Huffman code. The Huffman code is mainly used for lossless data compression [36]. In this method, a set of codewords with a minimum expected codeword is generated. The decompression is performed by converting the prefix codes into bytes by traversing each node of Huffman tree. After generating the secret message S_M, the decimal form of S_M is changed to Huffman code *hcode* using the dictionary generation. The dictionary is generated using the Huffmandict() in MATLAB. Huffman code is used because to compress the ciphered data, so that the message can be easily embedded in minimum number of pixels. Algorithm 2 shows the pseudocode for Huffman encoding.

Algorithm 2: Huffman encoding of ciphertext C

```
Input: $S_M$
Output: hcode
binary=decimalToBinaryVector(C); (MATLAB function to convert decimal to binary)
dict=Huffmandict(symbols,p); (MATLAB dictionary dict generation function with symbols and the probability of each symbol p)
hcode=Huffmanenco(sig,dict); (MATLAB function for Huffman encoding where sig is the data sequence for encoding)
```

C. Cover Image Pixel Selection for Data Embedding

Pixel selection in cover image is an important problem for concealing the covert information, so that the data is undetected and the image feature is not humiliated. If the pixels of noisy area (edges) are selected then the message is undetected. There are different standard edge detection algorithms such as Canny, Log, Prewitt, Roberts, Sobel, fuzzy logic, etc. However, Canny seems to be a better algorithm to identify the edges of an image [37]. Canny uses the low threshold, high threshold, and Gaussian kernel width for finding the image edges. Therefore, to detect the edges, Canny edge detection algorithm is used in this work. After finding the edges using the Canny method the edge pixels $P_e = \{P_1, P_2, \ldots, P_n\}$ of the cover image I are stored for further processing. If the image is a color image then the red plane intensities, green plane intensities, and blue plane intensities are collected from each edge pixel and the luminance value L is calculated for each edge pixel using Eq. (1) [15].

$$L(k) = 0.2126\times red(m,n) + 0.7152\times green(m,n) + 0.0722\times blue(m,n) \tag{1}$$

Then, the luminance values are sorted in decreasing order to find the high luminance values. The positions are also recorded with respect to the high luminance values to low luminance values of the edge pixels. Then the size of the Huffman code is extracted using the length(). Then the numbers of sorted pixels of same size are selected for data embedding. For example, if the Huffman code is 10110 then there are 5 bits, hence 5 edge pixel positions with high luminance values are selected for data embedding. If the image is a grayscale image then the intensities are collected from each edge pixel. Then, the intensity values are sorted in decreasing order to find the high intensity values. The positions are also recorded with respect to the high to low intensity values the edge pixels. Then the size of the Huffman code is extracted using the length(). Then the numbers of sorted pixels of same size are selected for data embedding. Algorithm 3 shows the pseudo code for cover image edge pixel selection.

Algorithm 3: Cover image pixel selection
Input: I, hcode, k=0, l=0
Output: Pixel position (Edge(i,j))

```
I=imread("image");
Edge = edge(I,'Canny'); ( Find edge of cover image)
L = length(hcode); (Find the number of pixels where a single bit of hcode will be embedded)
if I==RGB then (Color image)
    R(i,j) = I(:,:,1);
    G(i,j) = I(:,:,2);
    B(i,j) = I(:,:,3);
    [M,N]=size(R);
    for i=1 to M do
        for j=1 to N do
            if I(i,j)==Edge pixel then
                LM(k)=0.2126×R(i,j) + 0.7152×G(i,j) + 0.0722×B(i,j); (Luminance value calculation)
                m(k,1)=i;
                m(k,2)=j;
                k=k+1;
            end if
        end for
    end for
else (Grayscale image)
    for i=1 to M do
        for j=1 to N do
            if I(i,j)==Edge pixel then
                m(k,1)=i;
                m(k,2)=j;
                k=k+1;
            end if
        end for
    end for
end if
if RGB then
    Sort(LM) and record positions according to sorted LM; (Pixel positions stored in matrix-m)
else
    Sort(I) and record positions according to sorted I;
end if
for i=1 to L do (Pixel Selection)
    for i=1 to 2 do
        PixelSelected(l,1)=m(l,1); (i point)
        PixelSelected(l,2)=m(l,2); (j point)
        l=l+1;
    end for
end for
End
```

D. Embedding Encoded Data at the Selected Pixels to Generate Stego Image

In data implanting at the cover image, the main problem is to embed the data in such a way that the conceal message is undetected. If a pixel totally changes after data embedding then the image quality reduces and detectability rate increases. Therefore, LSB of a pixel is utilized for data embedding. If a single bit changes at LSB then there will be a very little change at the pixel level. This maintains the secrecy, quality, and undetectability to generate a final Stego image. The conceal message is then augmented with the pixel positions and communicated to the receiver. The augmented message format is represented in Fig. 3.

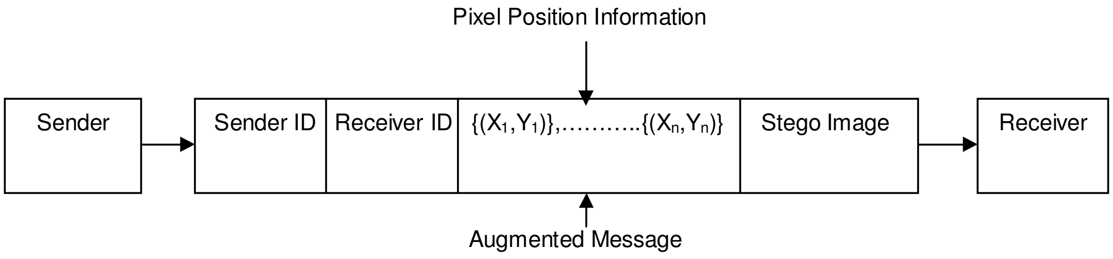


Fig. 3. Sender sending the Stego image using augmentation.

For instance, if a pixel value in binary form is 10110100 and the Huffman code is 10110, then the LSB 0 is replaced with the first bit of the Huffman code (1). The Huffman code has 5 bits, hence 5 pixels are selected for such LSB replacement. If the image is a color image, then the red plane pixels are selected for such LSB replacement because red has a high luminance then green and blue. Algorithm 4 shows the pseudocode for Stego image generation.

Algorithm 4: Generation of Stego Image by Data Embedding at the Selected Pixels
Input: I, hcode, k=1, l=1
Output: Stego image

```
1: if I==RGB then
2:     for i=1 to L do
3:         for j=1 to 2 do
4:             x=PixelSelected(k,1);
5:             y=PixelSelected(k,2);
6:             Pred=I(x,y,1); . Embedding at red plane
7:             B=decimalToBinaryVector(Pred);
8:             len=length(B);
9:             B(len)=hcode(i);
10:        end for
11:    end for
12: else
13:    for i=1 to L do
14:        for j=1 to 2 do
15:            x=PixelSelected(k,1);
16:            y=PixelSelected(k,2);
17:            P=I(x,y); . Embedding at selected edge pixel
18:            B=decimalToBinaryVector(P);
19:            len=length(B);
20:            B(len)=hcode(i);
21:        end for
22:    end for
23: end if
```

E. Secret Message Extraction from the Stego Image

After receiving the augmented message format, the receiver draws out the conceal message from the Stego image by the help of pixel positions. According to the pixel position order the receiver traverses to each pixel position and extracts the LSB from the pixel. Then the LSBs are merged to form a final Huffman code. This Huffman code is decoded by the receiver using the Huffman dictionary using *C = Huffmandeco* (*hcode, dict*), where *C* is the cipher text. The receiver contains the Huffman decode method to decode the encoded data. Then the cipher text *C* is decrypted using the RSA decryption algorithm ($S_M = C^d \bmod n$) to draw out the final secret message. Algorithm 5 shows the pseudo code for secret message extraction.

Algorithm 5: Extraction of Secret Message from the Stego Image

Input: Stego image, pixel positions

Output: Secret message S_M

```
1: if I==RGB then
2:     for i=1 to L do
3:         for j=1 to 2 do
4:             x=PixelSelected(k,1);
5:             y=PixelSelected(k,2);
6:             Pred=I(x,y,1);   (Embedding at red plane)
7:             B=decimalToBinaryVector(Pred);
8:             len=length(B);
9:             hcode(i)=B(len);
10:        end for
11:    end for
12:    C = Huffmandeco(hcode,dict);
13:    $S_M = C^d \bmod n$
14: else
15:    for i=1 to L do
16:        for j=1 to 2 do
17:            x=PixelSelected(k,1);
18:            y=PixelSelected(k,2);
19:            P=I(x,y);   (Embedding at edge pixel selected)
20:            B=decimalToBinaryVector(P);
21:            len=length(B);
22:            hcode(i)=B(len);
23:        end for
24:    end for
25:    C = Huffmandeco(hcode,dict);
26:    $S_M = C^d \bmod n$
27: end if
28: End
```

IV. SECURITY ANALYSIS OF 2L-EBS

As the sender use 2L-EBS to send the encrypted message, the security goals such as confidentiality, integrity, and availability are achieved [38-43].

A. Confidentiality

2L-EBS uses RSA encryption to design the message more intelligible by *C* and the receiver decrypts the message with its private key to generate *P*. It satisfies the confidentiality principle. It again changes the binary code (encrypted message) to a compressed code using Huffman encoding as a 2-level security policy and designs the system more confidential. The augmented message M_A communicated from the sender to the receiver contains M_A =< *StegoImage, Pixel Positions* >.

B. Integrity

If the augmented message is intercepted while transmission (*modification attack*), it cannot be modified after knowing the pixel position information because it lacks LSB policy. If the LSBs are generated and merged by the attacker, then also it is unable to generate the code because it has to perform Huffman decoding and RSA decryption. The attacker does not have the private key of the receiver also. Therefore, 2L-EBS is resilient against the modification attack. The 2LEBS also checks the *spoofing attack* because the attacker public key is unauthorized and the sender catches the attacker (as public key is provided by the Public Key Infrastructure (PKI) to an authorized user). The *replay attack* is also checked by identifying the unauthorized public key of the attacker. Therefore, from the three attacks it is concluded that the integrity is maintained by the 2L-EBS scheme.

C. Availability

The 2L-EBS scheme checks the *Denial of Service (DoS) attack* by restricting the unauthorized public identities to set a secure communication. The authorized users using 2L-EBS are able to perform the secure communication service.

D. Time Complexity and Message Complexity of 2L-EBS

Algorithm 1 takes a time complexity of O(n^3) for decryption and O(n^2) for encryption with n digits in Z space. Algorithm 2 has time complexity of O(nlogn) for both encoding and decoding, where n is referred to as the characters in n subtrees. Algorithm 3 has many steps: Steps 9-18, 12-16, 20-28, 23-26, 31 and 33 possess a time complexity of O(n^2) having n pixels of the cover image, O(n) having n edge pixels, O(n^2) with n pixels of the cover image, O(n) having n edge pixels, O(n^2) with n elements and O(n^2) with n pixels same as the size of Huffman binary code respectively. Algorithm 4 has many steps: Steps 2-11, 4-9, 13-22 and 15-20 possess a time complexity of O(n^2) having n selected edge pixels of the cover image, O(n) with n selected edge pixels of red plane, O(n^2) having n selected edge pixels of red plane of the cover image and O(n) with n selected edge pixels respectively. Algorithm 5 has many steps: Steps 2-11, 4-9, 13-22 and 15-20 possess a time complexity of O(n^2) with n selected edge pixels of the cover image, O(n) with n selected edge pixels of red plane, O(n^2) with n selected edge pixels of red plane of the cover image and O(n) having n selected edge pixels respectively. The message complexity of 2L-EBS is O(s×t), where s and t represent the rows and columns respectively in a cover media for choosing the number of pixels for data embedding. The message complexity for data embedding is O(k) with k edge pixels selected in a cover media.

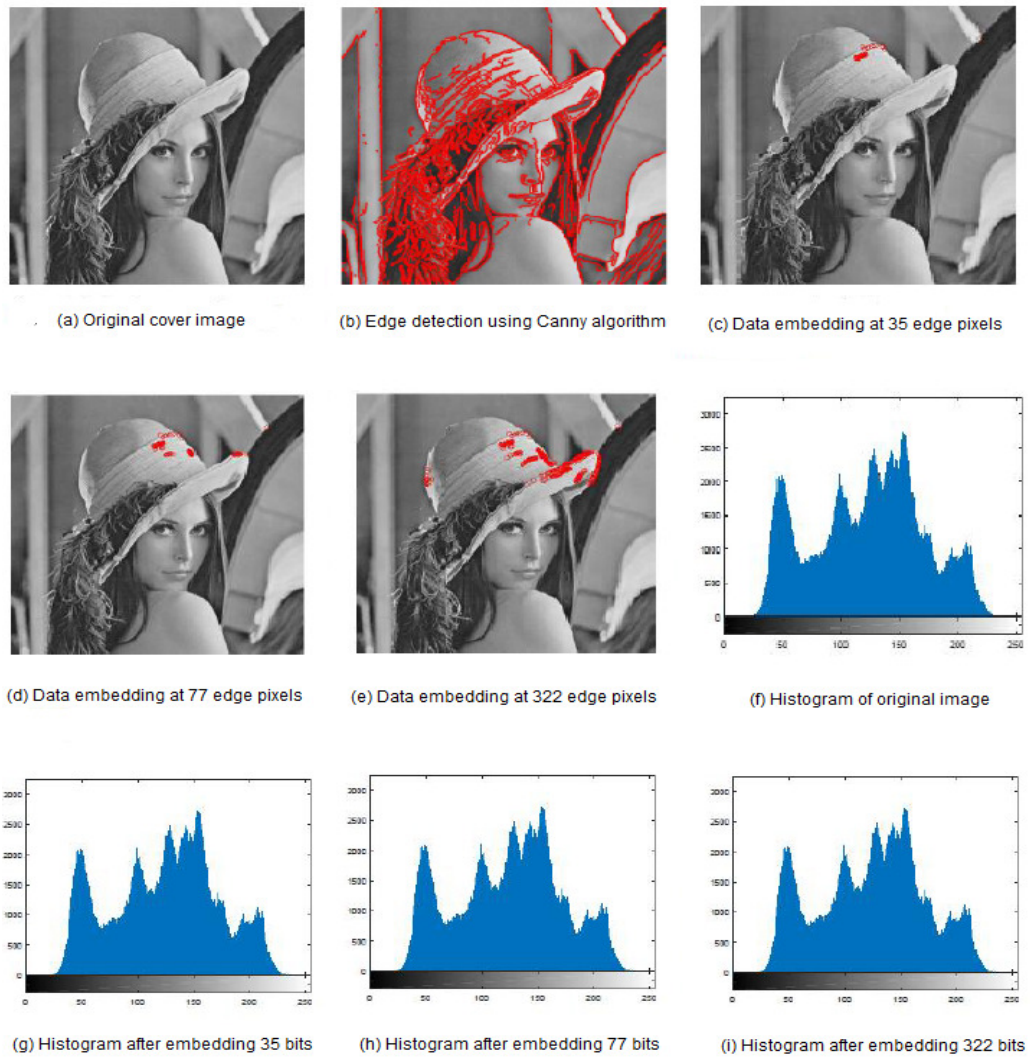
V. RESULTS AND DISCUSSION

The performance of the proposed 2L-EBS scheme is analyzed using MATLAB R2015a [44]. The proposed method is carried out in a machine of core i5 processor, 6 GB RAM, and Windows 10 platform. The 2L-EBS scheme is compared with Seeja *et al*. [15], Sun *et al*. [36], Luo *et al*. [34], and Wu *et al*. [21] schemes. The performance metrics for the comparison are MSE and PSNR.

- **MSE** uses the immse(X,Y) which computes the error between the arrays X and Y .
- **PSNR** uses psnr(A, ref) which computes the apex signal to noise ratio for the image A , with the reference image ref.

For implementation, three secret messages with different message sizes are used for performance evaluation. The messages with the sizes are shown as follows:

- **Hello** = 35 bits
- **Hello World** = 77 bits
- **Edge Based Steganography Using RSA and Huffman** = 322 bits



(a) Original cover image

(b) Edge detection using Canny algorithm

(c) Data embedding at 35 edge pixels

(d) Data embedding at 77 edge pixels

(e) Data embedding at 322 edge pixels

(f) Histogram of original image

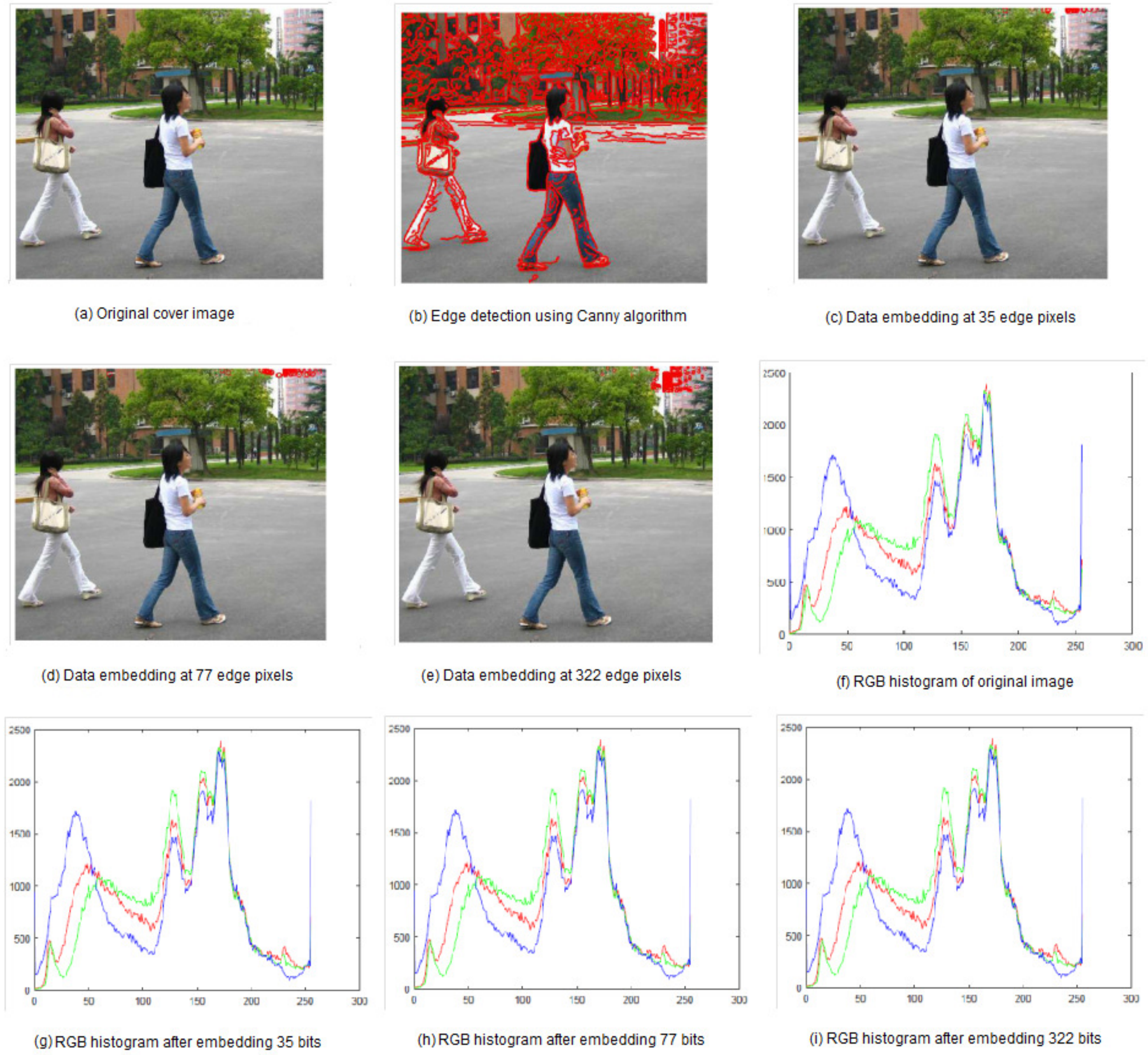
(g) Histogram after embedding 35 bits

(h) Histogram after embedding 77 bits

(i) Histogram after embedding 322 bits

Fig. 4. Histogram representation of proposed method for variable message size data embedding in Lena Image.

The messages are embedded in the cover images of size 512×512 (Lena [44]) and 992×497 (Fudan [45]). Penn-Fudan database for pedestrian detection has 174 images where 74 images are taken in Fudan University [45]. From 74 images we have tested 10 images for performance evaluation. We observed that 2L-EBS outperforms other methods in terms of MSE and PSNR. In this result and discussion section, we have only shown the result for a Fudan pedestrian image along with Lena image [44,45]. 35, 77, and 322 number of edge pixels are selected for embedding the 2-Level secure data of size 35 bits, 77 bits, and 322 bits to generate the Stego image. Fig. 4 demonstrates the results of 2L-EBS in Lena image. Fig. 4(a) shows the original cover image used for implanting the data. Fig. 4(b) shows the edge pixels generated using the Canny operator. Fig. 4(c), 4(d) and 4(e) show the 2-Level secure data embedding at 35, 77 and 322 selected edge pixels respectively. Fig. 4(f) demonstrates the histogram of original cover image. Fig. 4(f) and Fig. 4(g) show the comparison of the histograms when data is embedded at 35 selected edge pixels. From these two figures, it is observed that the histograms are similar to one another because only single bit LSB is changed and it is undetectable. Fig. 4(f) and Fig. 4(h) show the comparison of the histograms when data is embedded at 77 selected edge pixels. From these two figures, it is observed that the histograms are similar to one another. Fig. 4(f) and Fig. 4(i) show the comparison of the histograms when data is embedded at 322 selected edge pixels. From these two figures, it is observed that the histograms are similar to one another.


(a) Original cover image

(b) Edge detection using Canny algorithm

(c) Data embedding at 35 edge pixels

(d) Data embedding at 77 edge pixels

(e) Data embedding at 322 edge pixels

(f) RGB histogram of original image

(g) RGB histogram after embedding 35 bits

(h) RGB histogram after embedding 77 bits

(i) RGB histogram after embedding 322 bits

Fig. 5. Histogram representation of proposed method for variable message size data embedding in Fudan pedestrian image.

Fig. 5 demonstrates the results of 2L-EBS using pedestrian detection Fudan University image. Fig. 5(a) shows the original cover image used for implanting the data. Fig. 5(b) shows the edge pixels generated using the Canny operator. Fig. 5(c) shows the 2-Level secure data embedding at 35 selected edge pixels. Fig. 5(d) shows the 2-Level secure data embedding at 77 selected edge pixels. Fig. 5(e) shows the 2-Level secure data embedding at 322 selected edge pixels. Fig. 5(f) demonstrates the histogram of original cover image. Fig. 5(f) and Fig. 5(g) show the comparison of the RGB histograms when data is embedded at 35 selected edge pixels. From these two figures, it is observed that the RGB histograms are similar to one another because only single bit LSB is changed and it is undetectable. Fig. 5(f) and Fig. 5(h) shows the comparison of the RGB histograms when data is embedded at 77 selected edge pixels. From these two figures, it is observed that the RGB histograms are similar to each other. Fig. 5(f) and Fig. 5(i) shows the comparison of the RGB histograms when data is embedded at 322 selected edge pixels. From these two figures, it is observed that the RGB histograms are similar to each other. Table 1 and Table 2 demonstrate the comparison of MSE and PSNR for 2L-EBS scheme, Seeja *et al*. [15], Sun *et al*. [36], Luo *et al*. [34], and Wu *et al*. [21] schemes for Lena and Fudan pedestrian images respectively. From the results, it is observed that 2L-EBS carries out better performance than Seeja *et al*. [15], Sun *et al*. [36], Luo *et al*. [34], and Wu *et al*. [21] schemes when the variable size secret data is embedded at the edge pixels. Huffman code compresses the data where lower number of pixels is selected for data implanting which increases the undetectability rate and the feature of image is maintained.

Table 1. Comparison of 2L-EBS and other methods for Lena Image

Method	**Message Size**	**Performance Metrics**		**Message Detail**
		PSNR (dB)	MSE	-“Hello” : 35 bits -“Hello World”: 77 bits -“Edge Based Steganography Using RSA and Huffman”:322 bits
2L-EBS	35 bits	92.31	3.18	
	77 bits	89.52	7.24	
	322 bits	82.67	3.50	
Seeja *et al*. [15]	35 bits	91.52	4.57	
	77 bits	88.51	9.15	
	322 bits	81.78	4.71	
Sun *et al*. [36]	35 bits	91.12	4.88	
	77 bits	87.34	9.72	
	322 bits	80.94	5.18	
Luo *et al*. [34]	35 bits	89.77	5.94	
	77 bits	85.69	10.16	
	322 bits	78.56	6.38	
Wu *et al*. [21]	35 bits	86.25	7.44	
	77 bits	82.55	12.81	
	322 bits	74.78	7.33	

Table 2. Comparison of 2L-EBS and other methods for Fudan Pedestrian Image.

Method	**Message Size**	**Performance Metrics**		**Message Detail**
		PSNR (dB)	MSE	-“Hello” : 35 bits -“Hello World”: 77 bits -“Edge Based Steganography Using RSA and Huffman”:322 bits
2L-EBS	35 bits	93.33	4.16	
	77 bits	90.51	8.26	
	322 bits	81.67	4.50	
Seeja *et al*. [15]	35 bits	91.52	4.57	
	77 bits	86.51	10.21	
	322 bits	80.78	5.67	
Sun *et al*. [36]	35 bits	90.22	5.22	
	77 bits	85.44	11.23	
	322 bits	78.64	5.98	
Luo *et al*. [34]	35 bits	87.89	6.58	
	77 bits	83.63	12.66	
	322 bits	75.48	6.08	
Wu *et al*. [21]	35 bits	83.56	7.44	
	77 bits	80.42	14.28	
	322 bits	70.34	8.92	

VI. CONCLUSION

In this work, a two level secured edge based image steganographic approach is proposed to hide information of size 35 bits, 77 bits, and 322 bits in a cover image. The information is first encrypted using the RSA algorithm to generate a ciphertext, followed by encoded process by using the Huffman coding in order to generate a two level secured data. Finally, the encoded data is then placed at the LSBs of the cover image edge pixels to generate a Stego image. The performance of 2L-EBS is evaluated by randomly selecting a size of 512×512 and 992 ×497 for both gray-scale and RGB color image respectively. The experimental analysis revealed that 2L-EBS performs better than Seeja *et al*. [15], Sun *et al*. [36], Luo *et al*. [34], and Wu *et al*. [21] schemes in terms of PSNR and MSE. The Stego image generated has also high quality and high detectability. Therefore, this method will be a better solution for edge based image steganography. The proposed scheme outperforms better than its counterparts; hence, this 2-level architecture can be further studied in different application oriented optimized system.

ACKNOWLEDGEMENT

We owe our deep gratitude to Utkal University, Bhubaneswar and the department of Computer Science Engineering and Applications, Indira Gandhi Institute of Technology, Sarang (nodal centre of Utkal University, Bhubaneswar) for providing better research environment for successfully completing our research work. Also, we would like to extend our sincere esteems to all authors of reference.

Conflicts of interest: There is no conflict of interest.

REFERENCES

[1]. Islam, S., Modi, M.R., and Gupta, P. (2014). Edge-based image steganography. *EURASIP Journal on Information Security*, **8**(1): 1-14.

[2]. Hussain, M., Wahab, A.W.A., Idris, Y.I.B., Ho, A.T., and Jung, K.H. (2018). Image steganography in spatial domain: A survey. *Signal Processing: Image Communication*, **65**: 46-66.

[3]. Cheddad, A., Condell, J., Curran, K., and Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. *Signal processing*, **90**(3): 727-752.

[4]. Kumar, V., and Kumar, D. (2010). Performance evaluation of dwt based image steganography. In *2010 IEEE 2nd International Advance Computing Conference*: 223-228.

[5]. Subhedar, M.S., and Mankar, V.H. (2014). Current status and key issues in image steganography: A survey. *Computer science review*, **13**: 95-113.

[6]. Kanan, H.R., and Nazeri, B. (2014). A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm. *Expert Systems with Applications*, **41**(14): 6123-6130.

[7]. Feng, B., Lu, W., and Sun, W. (2015). Secure binary image steganography based on minimizing the distortion on the texture. *IEEE transactions on Information Forensics and Security*, **10**(2): 243-255.

[8]. Bailey, K., and Curran, K. (2006). An evaluation of image based steganography methods. *Multimedia Tools and Applications*, **30**(1): 55-88.

[9]. Selvi, G.K., Mariadhasan, L., and Shunmuganathan, K.L. (2012). Steganography using edge adaptive image. In *2012 IEEE International Conference on Computing, Electronics and Electrical Technologies*: 1023-1027.

[10]. Mishra, R., Mishra, A., and Bhanodiya, P. (2015). An edge based image steganography with compression and encryption. In *2015 IEEE International Conference on Computer, Communication and Control*: 1-4.

[11]. Swain, G., and Lenka, S. K. (2012). A Novel Approach to RGB Channel Based Image Steganography Technique. *Int. Arab J. e-Technol.*, **2**(4): 181-186.

[12]. Modi, M.R., Islam, S., and Gupta, P. (2013). Edge based steganography on colored images. In *Springer International Conference on Intelligent Computing*: 593-600.

[13]. Mandal, J.K., and Das, D. (2012). Colour image steganography based on pixel value differencing in spatial domain. *International journal of information sciences and techniques*, **2**(4): 83-93.

[14]. Arora, S., and Anand, S. (2013). A proposed method for Image Steganography using Edge Detection. *International Journal of Emerging Technology and Advanced Engineering*, **3**(2): 296-297.

[15]. Seeja, K.R., Rana, J., Priya, S., and Ahuja, L. (2016). A Novel Edge Based Image Steganography Technique. In *Springer International Conference on Soft Computing and Pattern Recognition*: 66-75.

[16]. Yang, C.H., Weng, C.Y., Wang, S.J., and Sun, H.M. (2008). Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Transactions on Information Forensics and Security*, **3**(3): 488-497.

[17]. Maheswari, S.U., and Hemanth, D.J. (2015). Frequency domain QR code based image steganography using Fresnelet transform. *AEU-International Journal of Electronics and Communications*, **69**(2): 539-544.

[18]. Chan, C.K., and Cheng, L.M. (2004). Hiding data in images by simple LSB substitution. *Pattern recognition*, **37**(3): 469-474.

[19]. Ker, A.D. (2004). Improved detection of LSB steganography in grayscale images. In *Springer International workshop on information hiding*: 97-115.

[20]. Tan, S., and Li, B. (2012). Targeted steganalysis of edge adaptive image steganography based on LSB matching revisited using B-spline fitting. *IEEE Signal Processing Letters*, **19**(6): 336-339.

[21]. Wu, H.C., Wu, N.I., Tsai, C.S., and Hwang, M.S. (2005). Image steganographic scheme based on pixel-value differencing and LSB replacement methods. *IEE Proceedings-Vision, Image and Signal Processing*, **152**(5): 611-615.

[22]. Pevný, T., Filler, T., and Bas, P. (2010). Using high-dimensional image models to perform highly undetectable steganography. In *Springer International Workshop on Information Hiding*: 161-177.

[23]. Ker, A.D. (2007). Steganalysis of embedding in two least-significant bits. *IEEE Transactions on Information Forensics and Security*, **2**(1): 46-54.

[24]. Fridrich, J., Pevný, T., and Kodovský, J. (2007). Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. In *ACM Proceedings of the 9th workshop on Multimedia & security*: 3-14.

[25]. Ker, A.D. (2005). A general framework for structural steganalysis of LSB replacement. In *Springer International Workshop on Information Hiding*: 296-311.

[26]. Provos, N., and Honeyman, P. (2003). Hide and seek: An introduction to steganography. *IEEE security & privacy*, **99**(3): 32-44.

[27]. Westfeld, A., and Pfitzmann, A. (1999). Attacks on steganographic systems. In *Springer International workshop on information hiding*: 61-76.

[28]. Dumitrescu, S., Wu, X., and Wang, Z. (2002). Detection of LSB steganography via sample pair analysis. In *Springer International Workshop on Information Hiding*: 355-372.

[29]. Fridrich, J., Goljan, M., Hogea, D., and Soukal, D. (2003). Quantitative steganalysis of digital images: estimating the secret message length. *Multimedia systems*, **9**(3): 288-302.

[30]. Fridrich, J., and Goljan, M. (2004). On estimation of secret message length in LSB steganography in spatial domain. In *Security, steganography, and watermarking of multimedia contents VI*, *International Society for Optics and Photonics*, **5306**: 23-34.

[31]. Pevny, T., Bas, P., and Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, **5**(2): 215-224.

[32]. Fridrich, J., and Kodovsky, J. (2012). Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, **7**(3): 868-882.

[33]. Kodovsky, J., Fridrich, J., and Holub, V. (2012). Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, **7**(2): 432-444.

[34]. Luo, W., Huang, F., and Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on information forensics and security*, **5**(2): 201-214.

[35]. Zhang, X., and Wang, S. (2004). Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. *Pattern Recognition Letters*, **25**(3): 331-339.

[36]. Sun, S. (2016). A novel edge based image steganography with 2k correction and Huffman encoding. *Information Processing Letters*, **116**(2): 93-99.

[37]. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(6): 679 -698.

[38]. Manikandan, G., Bala Krishnan, R., Preethivi, E., Sekar, K.R., Manikandan, R. and Prassanna, J. (2019). An Approach with Steganography and Scrambling Mechanism for Hiding Image over Images. *International Journal on Emerging Technologies*, **10**(1): 64-67.

[39]. Kalyane, Sangamesh and Patil and Nagaraj B. (2019). Secure and Efficient Data Communication for Hierarchal Cluster using Identity based Signatures. *International Journal on Emerging Technologies*, **10**(1): 54-58.

[40]. Naqvi, I. and Khan, A.S. (2019). Image Binarization of Deteriorated Historical Documents for Document Image Analysis. *International Journal of Electrical, Electronics and Computer Engineering*, **8**(1): 1-10.

[41]. Kumar, N., and Jain, S. (2018). A Review of Digital Watermarking in Protect Information Copyright info Privacy Techniques. *International Journal on Emerging Technologies*, **9**(2): 54-61.

[42]. Chouhan, R. and Vyas, A. (2018). A Review Content-Aware Dark Image Enhancement in Image Fusion Techniques. *International Journal on Emerging Technologies*, **9**(1): 36-43.

[43]. Pandey, S. and Shrivastava, A. (2018). A Image Encryption Scheme is Based on Scan Pattern for Colour Image. *International Journal of Electrical, Electronics and Computer Engineering*, **7**(1): 01-05.

[44]. https://www.mathworks.com [Online; accessed 02-January-2019].

[45]. https://www.cis.upenn.edu/~jshi/ped_html/ [Online; accessed 03-January- 2019].

How to cite this article: Jena, K.K., Mishra, S., Mishra, S. and Bhoi, S.K. (2019). An Edge based Steganographic Approach using a two Level Security Scheme for Digital Image Processing and Analysis. *International Journal on Emerging Technologies*, **10**(2): 29-38.