IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Enhancement of Efficiency in LSB Steganography Method Using Matrix Multiplication

Tanu Garg[1*], Kamaldeep Joshi[2], Jyoti Pandey[3], Harkesh sehrawat[4], Rainu Nandal[5]

[1*]Computer science &Engg., UIET, Maharshi Dayanand University, Rohtak, India
[2]Computer science &Engg., UIET, Maharshi Dayanand University, Rohtak, India
[3]Computer science &Engg., UIET, Maharshi Dayanand University, Rohtak, India
[4]Computer science &Engg., UIET, Maharshi Dayanand University, Rohtak, India
[5]Computer science &Engg., UIET, Maharshi Dayanand University, Rohtak, India

*Corresponding Author: gargtannu@rediffmail.com, Tel.: +917206296300

Available online at: www.ijcseonline.org

Accepted: 22/Jun/2018, Published: 30/Jun/2018

***Abstract*—**The quick progression in the trading of data through internet made it simpler to trade information correct and quickest to the receiver. The security of information is one of the immense parts of information technology and communication. Steganography is a term utilized for data hiding and it is a craft of concealed written work. In steganography we hide information with a multimedia carrier i.e. image, text, audio, video files, etc. Thus, the observers can't locate the concealed data which we need to send to the recipient. Steganography principle objective is to give robustness, imperceptibility, and limit of hiding information because of which it contrasts from different methods, for example, watermarking and cryptography. In image steganography we shroud our secret information in image with the goal that the observer can't feel its reality. Image steganography procedures are starting late been valuable to send any secret message in the secured image carrier to foresee dangers and assaults, however, it doesn't give any kind of chance to programmers to discover the secret technique. Image steganography is efficient and better type than other types of steganography. The LSB method also faces the same challenge regarding the selection of which bits are used for hiding the data without effect the actual image pixels. This paper proposes a new technique used to hide information by image steganography using matrix multiplication in which we pick 6th , 7th , 8th bit of image pixel and add our message with it by applying some techniques and resultant will be added or placed at 6th , 7th , 8th bit of image. In this we made changes up to 3 bits, i.e. changes of seven in binary term, but we have to add +6 or -6 not the seven to form the stego image.

Keywords— *Steganography, Information hiding, LSB, Matrix multiplication, PSNR, MSE*

I. INTRODUCTION

Data hiding for transferring information in an insecure medium is a vital and dynamic subject [1]. One of the fundamental ways in to secure a data is information hiding, that endeavors to shroud secret "Messages" in such way by which nobody can find them. Data covering has three things, which incorporate watermarking, cryptography, steganography. Cryptography utilized just to secure the communication by encrypting or decrypting our information. Yet, in the event that some irregular individual unscrambles that code, at that point there will be no security framework. Along these lines, another term proposed with cryptography i.e. steganography in which we secure the communication, as well as the presence of the message. As another term watermarking in which we endeavor to keep our hidden data by opposite of changing hidden message, which is a piece of cover information, however, steganography we didn't do this we ensure that nobody knows the presence of the message [2]. Steganography can be utilized as both lawfully and illicitly. Great user utilizes it for securing correspondence while the programmer utilizes it illicitly to increase other information.

The older and noted technique for concealing the data in the digital picture is LSB technique. The LSB is the simplest method for hiding the data and widely used. In this method the secret data is hidden by changing only the last significant bit of the original image. And there is no effect on the quality of the image. And the concealing capacity of the data can be increased up to last four significant bits. But the main drawback is easily detected and there from time to time there are many modifications and changing to strengthen the technique [3].

II. LITERATURE REVIEW

L. Zhi et al. proposed a spatial domain technique i.e. LSB. LSB is one of the most straightforward techniques for hiding information. In this method, the least significant bits of picture pixels are utilized for hiding messages. As though we change the last bit, at that point it doesn't have more effect in the picture and the unauthorized user can't recognize it just by the view. It is anything but difficult to utilize and execute and in this high message payload is there and there is no way of removal of the nature of the actual picture [4].

K. Muhammad et al. discussed about methodologies for LSB strategy. In this he talked about LSB matching (LSB-M) in which by including +1 or - 1 in given pixel it changes the pixel, at that point contrast message bit and the LSB of pixel if not same, at that point keep the estimation of a pixel in the range 0-255. In Cyclic LSB we shred the hidden data on the complete picture, use stego shading cycle procedure for the cover picture. Stego shading cycle conceals information in distinguish channels of the cover picture. The first secret bit in the red channel, second in the green channel, third in the blue channel. In Edge based area pixel can suit more secret bit than smooth zone and less perceivable by HVS. It isolates the data into two squares and explores the pixel starting from the point of convergence of the picture. In LSB-MR embeds 2 mystery bits at one time [5].

B. Siddiqui et al. describes the various techniques using the LSB substitution method to hide the data in images. Image steganography shrouds the information, products and viably with the help of LSB substitution strategies. In this paper, a technique is proposed to expand the measure of the limit of embedding messages and the nature of the stego-picture. A settled number of LSB's is utilized. The cover picture is isolated into two sections and changing procedure is connected to the estimation of a few bits that have the mystery bits in the stego-picture that are obtained by the basic type of LSB substitution system [6].

K. Joshi et al. proposed a new method of image steganography using last three bit plane of gray scale images. In this paper author extracts the last three bits of the chose pixel and afterward plays out the XOR task with third, first and third, second piece. Based on consequence of XOR activity all of the hidden information is inserted one by one on LSB of the chose pixel esteem. That method comes under +1 or -1 change in pixel value of a Stego image. Due to +1 or -1 modification, the method shows minimum degradation in the Stego image [7].

R. Tavoli et al. proposed a new method for text hiding in image by LSB method. In this it utilizes a phase of textual data compression and after that encodes for preceding steganography; first it applies a preprocessing method for the coveted content, and after that put the content in the picture. The described technique ciphers the compacted content and by the help of utilization of a 4*4 mask performs wind check requests. Afterwards it stacks the over compacted and hidden message on image pixels. The distinction between bits if the two bits are 0 or 1 at that point yield is 1; else it is 0 by the XOR operation [8].

S. Batra et al. proposed a method for hiding of the message using last three bits of pixel values of an image. In this we send two images, one is the original image and another is message hidden image. In the1st case we send '0' bit as a message and in case 2nd '1' as a message. Embedding can be done by using complement method and in case unintended person changes the least significant bit of an image pixel at its retrieval in which message has been embedded [9].

S. Arjun et al. proposed an approach to adaptive steganography based on Matrix Embedding. The other research done as such far proposes the utilization for insertion of matrix for a given approach to enhancing its insertion efficiency, yet this current comes' at the cost of reduction in installing limit. In this paper, to mitigate this issue they proposed an approach in view of matrix embedding. This approach gives higher inserting limits while giving higher installing productivity. They find utilization of Weber-Fechner law to make the proposed strategy outwardly subtle [10].

S.K. Muttoo et al. proposed a technique for robust and secure image steganographic algorithm based on matrix embedding. In this approach include a powerful (non-sensitive) steganography procedure in perspective of the lattice inserting using a self-synchronizing variable length T-codes and RS codes. The first message is first encoded using T-codes and after that with the RS-codes. The decision of the plane for embeddings is made in view of variances of force resolutions. The mystery message is then inserted in the chose second, third or fourth plane of the cover picture utilizing the matrix encoding system [11].

C. Zou et al. proposed a technique for information hiding in RGB images using an improved Matrix Pattern approach. In this procedure, firstly, we isolated the RGB image into the non-covering square-sized pieces. Next, 95 vital measured one of a kind matrix patterns are consequently produced utilizing the fourth and fifth bit layers of the green layer of each square. At that point, the blue layer of each square is utilized for installing secret messages by including matrix pattern which is doled out to the characters of the secret message [12].

A. F. Nilizadeh et al. proposed a technique for steganography on RGB images based on a "Matrix Pattern" using random blocks. In this paper, they portrayed a spatial area strategy in RGB images for steganography where in the blue layer of specific pieces mystery message is inserted. In this calculation, each square initially picks a one of a kind t1xt2 matrix of pixels as a "matrix pattern" for every console character, utilizing the bit distinction of neighboring pixels. Next, a mystery message is implanted in the rest of the piece of the square, those with no part in the "lattice design" determination system [13].

A. R. Naghsh Nilchi et al. proposed a new steganography algorithm with two different steganography methods to combine, in particular Matrix Pattern and Least Significant Bit, is introduced for RGB pictures. The Matrix Pattern technique was a calculation that, firstly, separates "Cover-Image" in the non-covering D×D pieces. At that point, it conceals the information in the blue layer of the 4th through 7th bit layers of the "Cover-Image", by creating an exceptionalt1*t2for each character in each MATRIX PATTERN square. The LSB strategy is a calculation that shrouds information at all huge piece of the "Cover-Image" values that has the minimum unmistakable impact on the straightforwardness of the "Stego-Image" [14].

T. Yang et al. proposed a technique for Matrix embedding in steganography with binary Reed-Muller codes. This paper presented an adjusted larger part rationale deciphering calculation of Reed– Muller (RM) codes for framework inserting in steganography. The ideal installing calculation in steganography is proportional to the maximum likelihood decoding calculation to correction of the error codes. The fundamental weakness of framework inserting is that the equal Maximum Likelihood Decoding calculation of protracted implanting codes needs exceedingly tough installing. This examination utilized Reed Muller codes to install information in twofold complete pictures. The creators proposed a low many-sided quality implanting calculation that uses an adjusted lion's share rationale calculation to interpret RM codes, in which a message passing calculation is performed on the most noteworthy request of data bits in the RM codes [15].

K. Joshi proposed LSB steganography method and investigate the PSNR and MSE of LSB information hiding procedure based on various message sizes. The proposed LSB plot takes the main LSB bit of the picture and the principal message bit from the message grid and additions the message into the primary picture. After expansion of first message bits, pixel zone of picture and message is expanded by one. In this PSNR is more for the short message and less if there should be an occurrence of long message size and MSE is less for the short message and more in the event of long message estimate[22].

III. METHODOLOGY

In proposed steganography method we worked on the LSB method using matrix multiplication operation. As with others if we make changes in the 6^{th}, 7^{th} and 8^{th} bit of image pixel value then there will change of +7 in binary terms, but in our method there will be changes of +6 or -6 or less. In section 3.1, selection of the image is done. Then, embedding step is introduced in section 3.2 and, finally the extraction step is defined in section 3.3.

III.1. Image Selection

In the image selection, the sender and receiver first select a cover image in which message to be hidden and they must agree on the same image. Both have a cover image at the time of embedding and extracting of the message. They share a secret key to decide the sequence of random locations in the cover image. Receiver must know the algorithm using which sender hides the message.

III.2. Embedding Phase

In the embedding step, first, we take a cover image, and then convert the cover image into 24 bit or 3 blocks. We then pick 6th, 7th, 8th bit of 1st pixel and made it R1 of matrix and precede until whole matrix form for others pixel. Then, we take message and convert its ASCII value to binary form. Take the secret message bits in matrix form by padding of bits i.e.1^{st} bit at R_{11} value and 2^{nd} bit at R_{12} and 3^{rd} bit at R_{13} and proceed until whole matrix form. Then, apply the matrix multiplication operation and get a resultant matrix. In resultant matrix value of R_{11} goes to 6th bit of 1st pixel, R_{12} goes to 7th bit of 1st pixel; R_{13} goes to 8th bit of 1st pixel and similarly replace R2, R3 for 2nd and 3rd pixel. Then, convert binary values to ASCII value and get the stego image. Figure 1 shows the algorithm to embed message in image. Figure 2 shows the flowchart for embedding of message.

Algorithm for embedding of message:

Input: Cover Image (I^C), Secret Message (M), Secret Key (K^{KEY})

M= {m_1, m2, m3 .... m_n}

1. Initialize I^C ← Cover Image, M ← Secret Message, K^{KEY} ← Secret Key
2. While Counter <= size of the message block do
3. For each pixel
   a. Pick a pixel I (x, y) from the image and convert it into 8 bit binary numbers.
   b. Pick 6th, 7th, 8th bit of pixel and made it R_1 of matrix and proceed until whole 3* 3 matrix is formed and put it inside the variable 'A'.
4. For secret message
   a. Pick a value from secret message and convert it into 8 bit binary numbers.
   b. Check we have a sufficient bit of message to form a matrix if yes, then made a matrix and put it inside the variable 'B'.
   c. If no, then pad bits from next value of a message.
5. Apply matrix multiplication operation i.e. A*B and get a resultant matrix and put it inside the variable 'C'.
6. If A's 6^{th}, 7^{th}, 8^{th} bit is '001' then

If C's 6^{th}, 7^{th}, 8^{th} bit is '000'then add -1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '001'then no change in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '010'then add +1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '011'then add +2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '100'then add +3 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '101'then add +4 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '110'then add +5 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '111'then add +6 in pixel component value.

Else if A's 6^{th}, 7^{th}, 8^{th} bit is '010' then

If C's 6^{th}, 7^{th}, 8^{th} bit is '000'then add -2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '001'then add -1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '010'then add no change in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '011'then add +1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '100'then add +2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '101'then add +3 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '110'then add +4 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '111'then add +5 in pixel component value.

Else if A's 6^{th}, 7^{th}, 8^{th} bit is '011' then

If C's 6^{th}, 7^{th}, 8^{th} bit is '000'then add -3 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '001'then add -2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '010'then add -1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '011'then add no change in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '100'then add +1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '101'then add +2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '110'then add +3 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '111'then add +4 in pixel component value.

Else if A's 6^{th}, 7^{th}, 8^{th} bit is '100' then

If C's 6^{th}, 7^{th}, 8^{th} bit is '000'then add -4 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '001'then add -3 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '010'then add -2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '011'then add -1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '100'then add no change in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '101'then add +1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '110'then add +2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '111'then add +3 in pixel component value.

Else if A's 6^{th}, 7^{th}, 8^{th} bit is '101' then

If C's 6^{th}, 7^{th}, 8^{th} bit is '000'then add -5 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '001'then add -4 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '010'then add -3 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '011'then add -2 in pixel component value.

If C's 6^{th}, 7^{th}, 8^{th} bit is '100'then add -1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '101'then add no change in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '110'then add +1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '111'then add +2 in pixel component value.
Else if A's 6^{th}, 7^{th}, 8^{th} bit is '110' then
If C's 6^{th}, 7^{th}, 8^{th} bit is '000'then add -6 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '001'then add -5 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '010'then add -4 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '011'then add -3 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '100'then add -2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '101'then add -1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '110'then add no change in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '111'then add +1 in pixel component value.
Else if A's 6^{th}, 7^{th}, 8^{th} bit is '111' then
If C's 6^{th}, 7^{th}, 8^{th} bit is '001'then add -6 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '010'then add -5 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '011'then add -4 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '100'then add -3 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '101'then add -2 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '110'then add -1 in pixel component value.
If C's 6^{th}, 7^{th}, 8^{th} bit is '111'then add no change in pixel component value,
Else if A's 6^{th}, 7^{th}, 8^{th} bit is '000' then
If C's 6^{th}, 7^{th}, 8^{th} bit is '000'then add no change in pixel component value
7. Convert binary value to ASCII value and get the stego image.
8. Counter=Counter+1;
9. Repeat steps 3to 7 until complete message is hidden.
10. End.
Output: Stego image (I^S)

III.3. Extracting Phase

In the extracting step, first, we take a cover image, and then convert the cover image into 24 bit or 3 blocks. We then pick 6th, 7th, 8th bit of 1st pixel and made it R1 of matrix and precede until whole matrix form for others pixel. Find the inverse of a matrix. Then, we take stego image and convert the stego image into 24 bit or 3 blocks. We then pick 6th, 7th, 8th bit of 1st pixel of stego image and made it R1 of matrix and precede until whole matrix form for others pixel. Then, apply the matrix multiplication operation and get a resultant matrix. Resultant matrix values are message bits. Then, convert message bits to ASCII value to get original message. Figure 3 shows the algorithm to extract message. Figure 4 shows the flowchart for extracting of message.

Algorithm for extracting of message:

Input: Cover Image (I^C), Stego Image (I^S), Secret Key (K^{KEY})
1. Initialize $I^C \leftarrow$ Cover Image, $I^S \leftarrow$ Stego Image, $K^{KEY} \leftarrow$ Secret Key
2. For each pixel
a. Pick a pixel I (x, y) from the cover image and convert it into 8 bit binary numbers.
b. Pick 6th, 7th, 8th bit of 1st pixel and made it R_1 of matrix and proceed until whole matrix form for others pixel.
c. Find the inverse of a matrix.
3. For each pixel
a. Pick a pixel I (x, y) from the stego image and convert it into 8 bit binary numbers.
b. Pick 6th, 7th, 8th bit of 1st pixel of stego image and made it R_1 of matrix and proceed until whole matrix form for others pixel.
4. Apply matrix multiplication operation and get a resultant matrix.
5. Resultant matrix values are message bits.
6. Convert message bits to ASCII value to get original message.
7. Repeat steps 2 to 6 until complete message is extracted.
10. End.
Output: Secret Message (M)

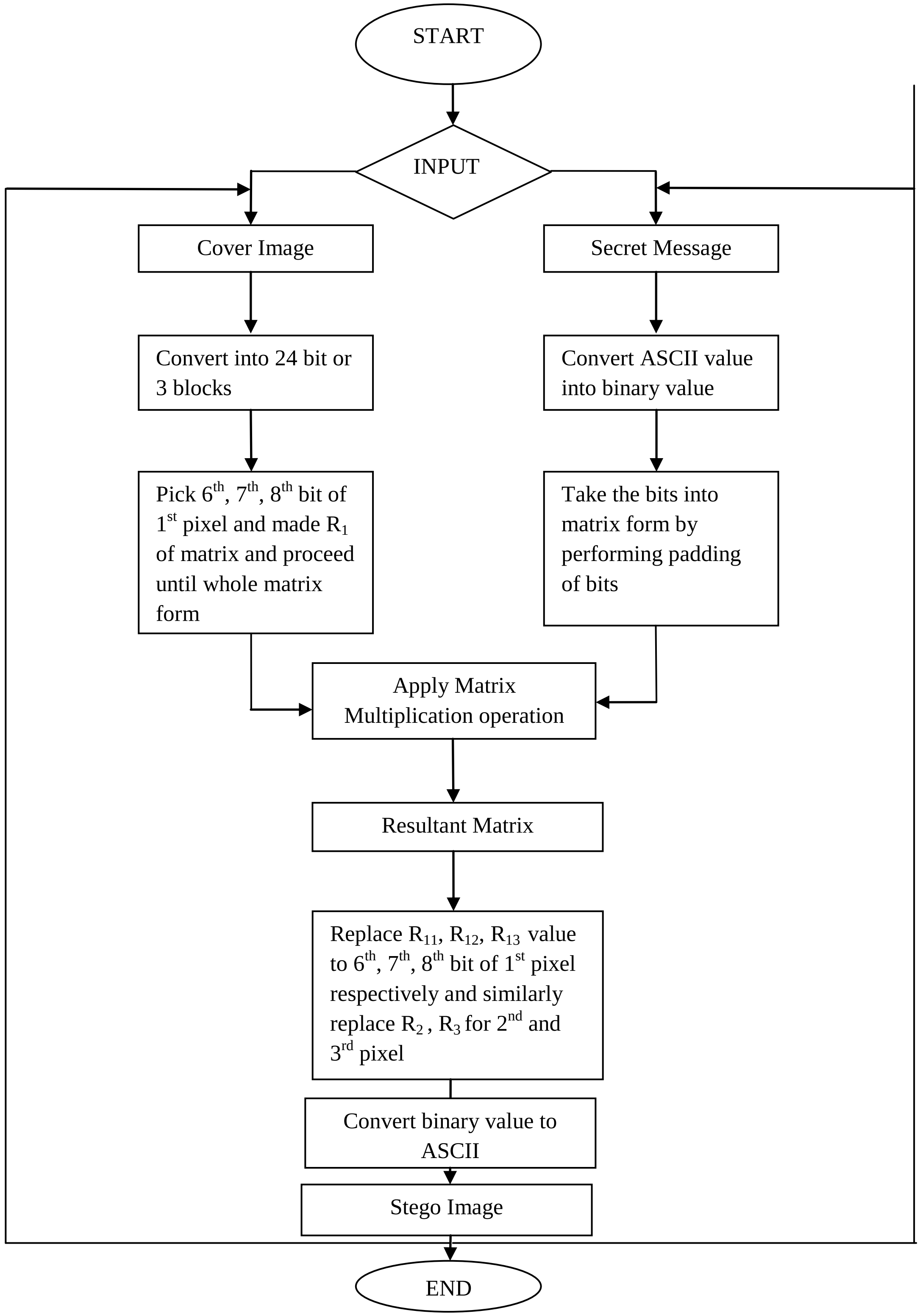


Figure 2: Flowchart for embedding of message

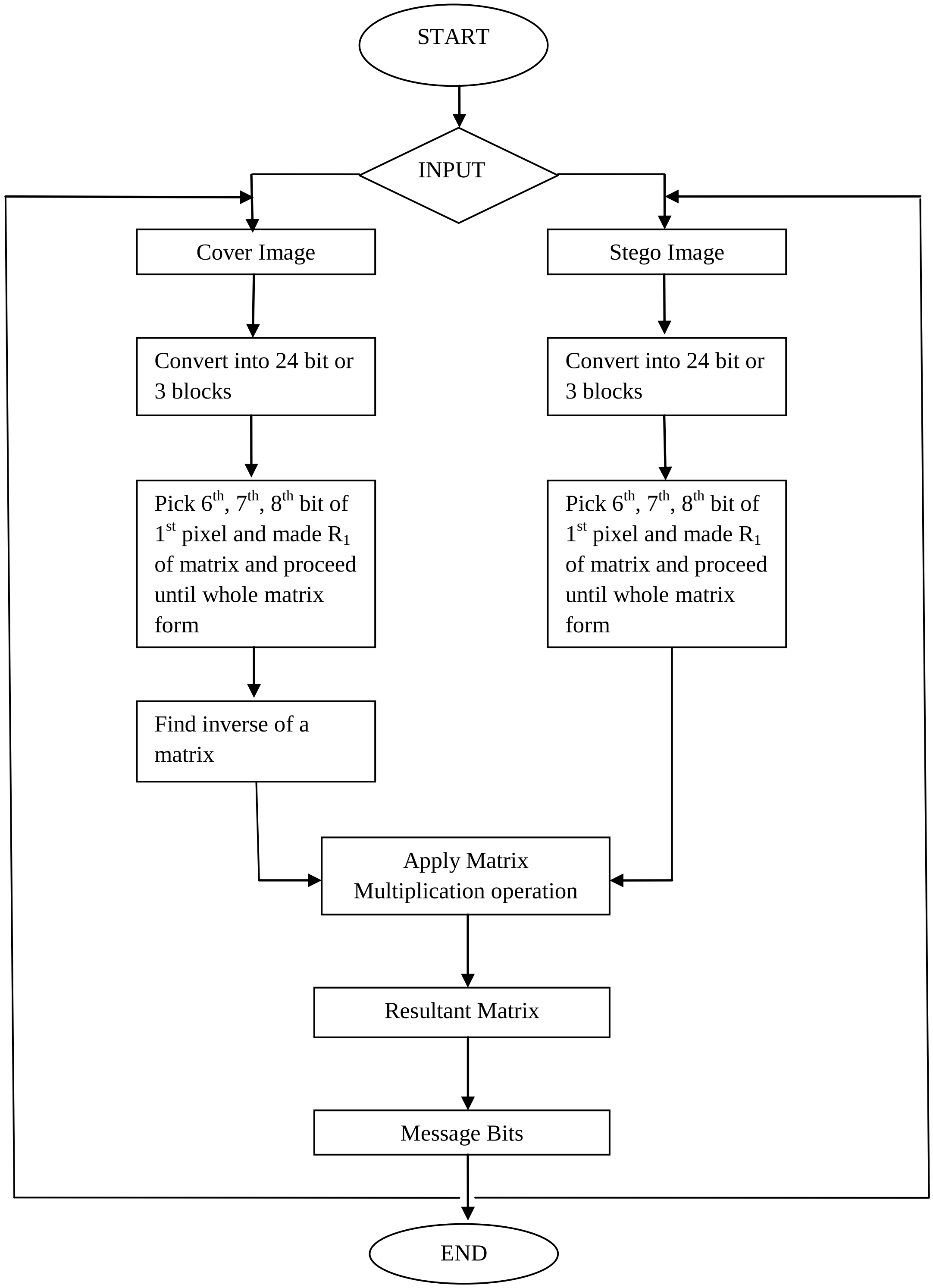


Figure 4: Flowchart for extracting of message

III.3. Example:

Consider some pixel of the cover image and also we have some secret data which we have to hide inside the cover image.

Cover Image Pixel {52, 74, 99, 25} Message: bcd

Binary value of pixel: ASCII value (66, 67, 68)

52= 00110100 74= 01001010 Binary value:

99= 01100011 25= 00011001 {01000010 01000011 01000100}

$$\begin{array}{c|ccc|} & 6^{th} & 7^{th} & 8^{th} \\ 52 & 1 & 0 & 0 \\ 74 & 0 & 1 & 0 \\ 99 & 0 & 1 & 1 \end{array} * \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Stego Image ⟶ 100 replace by 010 ⟶ changes of -2 (50)

010 replace by 000 ⟶ changes of -2 (72)

011 replace by 100 ⟶ changes of +1 (100)

Stego Image Pixel {50, 72, 100, 25}

Extraction:

Binary value stego image pixels:

50= 00110010 72= 01001000

100= 01100100 25= 00011001

$$\begin{array}{c|ccc|} & 6^{th} & 7^{th} & 8^{th} \\ 50 & 0 & 1 & 0 \\ 72 & 0 & 0 & 0 \\ 100 & 1 & 0 & 0 \end{array} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Stego Image Pixel Inverse of Cover Image Message Bits

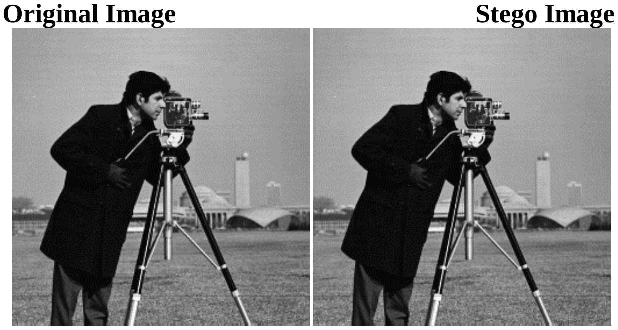
IV. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed technique is simulated using MATLAB R2017a

4.1 Data Set

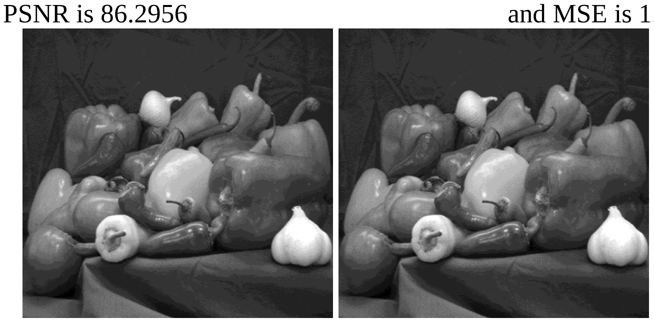
We utilized distinctive images for getting the outcome and did an investigation with grayscale images that are utilized to assess the execution of the proposed strategy. We utilize diverse measurement of images like 256*256, 384*512,128*128,480*640. Images used are like cameraman, peppers, fabric, pears, moon etc... We used different cases. According to case 1, secret message of 2 bits is embedded in different grayscale images. The second case is to embed message of 4 bits and in the third case message of 8 bits. We calculated PSNR, MSE values in all cases. Figure 5.shows the original and stego images of different dimension of the images in which a, c, e, g are original image and b, d, f, h are stego image. Figure 6 shows the histogram of original and stego images in which i, k, m, o are original image histogram and j, l, n, p are stego image histogram. Table 1 shows the PSNR value of different size of embedded message.

Original Image **Stego Image**



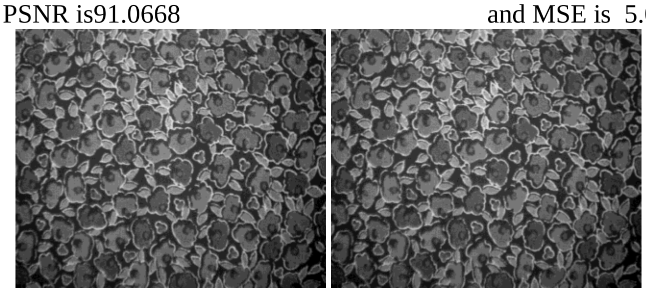
(a) When image size 256*256 and PSNR is 86.2956

(b) When image size 256*256 and MSE is 1.5259



(c) When image size 384*512 and PSNR is91.0668

(d) When image size 384*512 and MSE is 5.0863



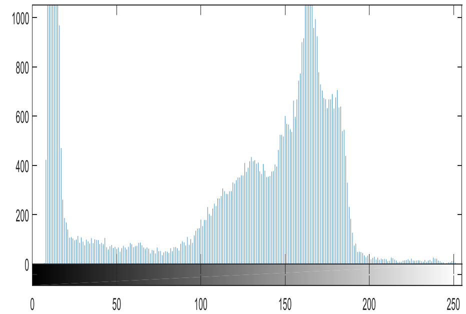
(e) When image size 480*640 and PSNR is 99.9947

(f) When image size 480*640 and MSE is 6.5104

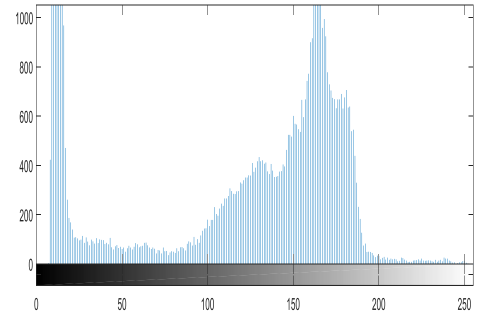
(g) When image size 486*732 and PSNR is 92.1810

(h) When image size 486*732 and MSE is 3.9353

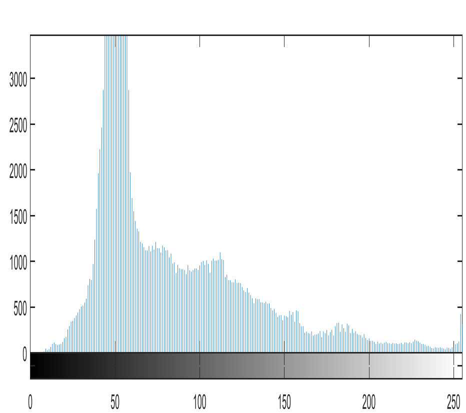
Figure 5: Original and Stego Image of different Size with its PSNR and MSE



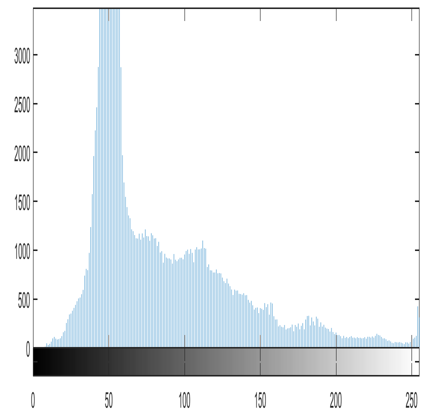
(i) Original Cameraman Image Histogram



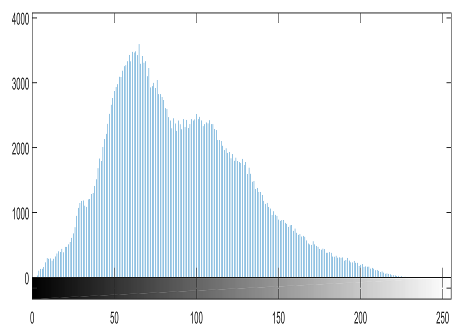
(j) Stego Cameraman Image Histogram



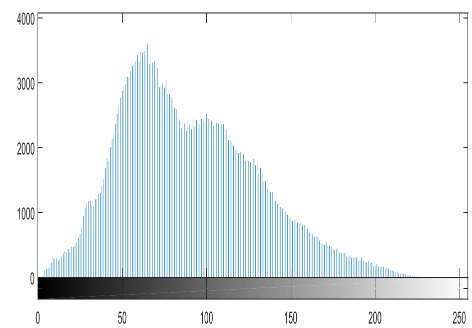
(k) Original Peppers Image Histogram



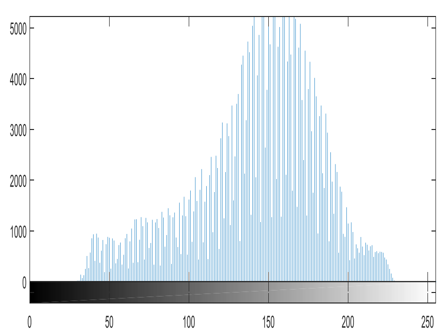
(l) Stego Peppers Image Histogram



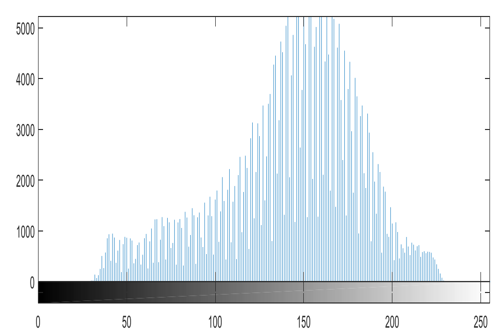
(m) Original Fabric Image Histogram



(n) Stego Fabric Image Histogram



(o) Original Pear Image Histogram



(p) Stego Pear Image Histogram

Figure 6: Original and Stego Image Histogram

TABLE 1: Obtained PSNR at different message size bits using the different image size

Image of different size	Message Size in byte	PSNR	Message Size in byte	PSNR	Message Size in byte	PSNR
Cameraman (256*256)	16	77.2108	4	81.5244	8	80.1678
Peppers (384*512)	16	82.2587	4	90.2750	8	83.6632
Fabric (480*640)	16	84.6799	4	89.7828	8	85.4463
Pears (486*732)	16	84.9500	4	90.4201	8	85.6489
Gantrycrane (264*400)	16	80.8856	4	86.3226	8	81.9329
Moon (537*358)	16	83.1879	4	87.5451	8	87.1673
Tire (205*232)	16	78.2756	4	81.1011	8	79.7181
Trees (258*350)	16	81.8898	4	87.2738	8	83.2161

V. COMPARISON

We are using different methods for comparisons to be done with the proposed method. There we are using a BRL method, Chang method, PVD method, PVD with LSB method, Side Match. This comparison shows that the proposed method is better than other methods. Table2 shows the comparison of the proposed method with different methods [21, 24].

TABLE 2: Comparison of proposed method with different methods

Algorithm	Capacity (bits)	PSNR
Proposed Method	134,792 215,054 316,528 430,006	53.019 51.456 48.567 46.673
BRL Method [16]	283,211 315,078 330,662	47.770 45.720 44.080
PVD with LSB [17]	528,512 766,040	38.80 36.160
PVD [18]	207,520 407,680	48.430 41.790
Side match [19]	389,004 267,242 164,538	41.220 45.030 48.180
Chang Method [20]	112,347	52.040

VI. CONCLUSION

In this paper, we proposed an effective image steganography technique which increases confidentially of the message. Here we are utilizing the image (gray scale image) which is 2 dimensional, which lessens the preparing time and improve the security of concealed information. In this proposed method, large amount of data can be hidden because we are hiding 3 bits of data in one pixel. The stored data at this position is not the actual data, but it is obtained by performing the matrix multiplication operation.

We are performing the matrix multiplication operation using 8th bit, 7th bit, 6th bit, 1st bit of data, 2nd bit of data, and 3rd bit of data. The result produces three bits which are not the actual data and we are storing this obtained bit at the LSB position. On the off chance that an unintended user ready to recognize the LSB bit, at that point we have no need of getting stressed on the grounds that this isn't the real information on this position. The proposed technique is doable as in this strategy is anything but difficult to execute, straightforward and furthermore give security against attack. This proposed strategy additionally improves the stego image of better quality. In the wake of seeing the stego image, nobody could envision the nearness of the message is in the image. After implementation and analysis, we get the good imperceptibility as changes in binary terms are either +6 or -6 not the changes of seven.

REFRENCES

[1] A. Cheddad and J. Condell, "Image Steganography survey and analysis", Faculty of Computing and Engineering, Volume 90, Issue 12, pp. 727-752 , 2010

[2] T. Morkel and J.H.P. Eloff, "Exploring simple steganography", Information and Computer Security Architecture Research Group, Department of Computer Science, University of Pretoria, Volume 87, Issue 2, pp. 26-34, 2005

[3] S. R. Yadav, A. Tiwari and N.K. Mittal, "Image Steganography technique", International Journal of Engineering and Innovative Technology, Volume 3, Issue 7, pp. 19-23, 2014

[4] L. Zhi and S. A. Fen, "LSB Image Steganography", Vehicular Technology Conference IEEE, Volume 3, Issue 5, pp. 2113-2117, 2004

[5] K. Muhammad and M. Sajjad, "Modification on LSB method", Springer Science, Volume 22, Issue 1, pp. 647-654, 2015

[6] B. Siddiqui and S. Goswami, "A Survey on Image steganography using LSB substitution technique", International Research Journal of Engineering and Technology, Volume 4, Issue 5, pp. 345-349, 2017

[7] K. Joshi, R. Yadav , "A new image steganography using three bit plane of gray level images", International Research Journal of Engineering and Technology, Volume 10, Issue 38, pp. 1-8, 2017

[8] R. Tavoli and M. Bakhshi, "A new method for text hiding in the image using LSB", International Journal of advanced computer science and applications, Volume 7, Issue 4, pp. 126-132, 2016

[9] S. Batra, R. Rishi and R. Yadav, "Insertion of message in 6th , 7th , 8th bit of pixel and its retrieval in cases changes the LSB of image pixels", International Journal of security and its applications, Volume 4, Issue 3, pp. 1-10, 2010

[10] S. Arjun, A. Negi and C. Kranthi, "An approach to adaptive steganography based on matrix embedding", Proceedings of the IEEE Conference, 2014

[11] S. K. Azad and S. K. Muttoo, "A Survey on Image steganography using LSB substitution technique", International Journal of Electronic security and Digital Forensics, Volume X, Issue Y, pp. 1-21, 2014

[12] C. Zou, W.Mazurczyk and G.T. Leavens, "Image Steganography technique", Computer Vision and Pattern Recognition Workshop and IEEE Conference, 2017

[13] A.F. Nilizadeh and A. R. Naghsh Nilchi, "Steganography on RGB Images based on matrix pattern using random blocks ", International Journal of Modern Education and Computer Science, Volume 5, Issue 4, pp. 8-18, 2013

[14] A.F. Nilizadeh and A. R. Naghsh Nilchi, "A Novel Steganography method based on matrix pattern and LSB algorithms in RGB images", Swarm Intelligence and Evolutionary Computation and IEEE Conference, 2016

[15] T. Yang and H. Chen, "Matrix embedding insteganography with binary Reed-Muller codes", Institution of Engineering and Technology, Volume 11, Issue 7, pp. 522-529, 2017

[16] V. M. Potdar and E. Chang, "Gray level modification steganography for secret communication", Proceedings of the International Conference on Industrial Informatics, pp. 223-8, 2004

[17] H.C. Wu, N.I. Wu, C.S. Tsai, and M.S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods", IEEE Proceedings of Vision, Image and Signal Processing, Volume 152, Issue 5, pp. 611-15, 2005

[18] D.C. Wu and W.H. Tsai, "A steganographic method for images by pixel-value differencing", Pattern Recognition Letters, Volume 24, Issue 9/10, pp. 1613-1626, 2003

[19] C. C Chang and H.W. Tseng, "A steganographic method for digital images using side match", Pattern Recognition Letters, Volume 25, Issue 12, pp. 1431-1437, 2004

[20] C. C. Chang, T. D. Kieu and Y. C. Chou, "Using nearest covering codes to embed secret information in grayscale images", Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, pp. 315-20, 2008

[21] A. Faruq and H.S. Ghwanmeh, "An innovative information hiding technique utilizing cumulative peak histogram regions", Journal of Systems and Information Technology, Volume 14, Issue 4, pp. 336-352, 2012

[22] K. Joshi, R. Yadav and R. Saini, "A new image steganography approach for information security using gray level images in spatial domain", International Journal on Computer Science and Engineering, Volume 3, Issue 7, pp. 2679-2690, 2011

[23] K. Joshi, R. Yadav and S. Allwadhi, "PSNR and MSE based investigation of LSB", Proceedings of the International Conference on Computational Techniques in Information and Communication Technologies, pp. 280-285, 2016

[24] K. Joshi, R. Yadav and G. Chawla, "An Enhanced method for data hiding using 2 bit XOR in image steganography", International Journal of engineering and technology, Volume 8, Issue 6, pp. 3043-3055, 2017

