An Enhanced Data Hiding Approach using Pixel Mapping Method with Optimal Substitution Approach

Prince Kumar Panjabi
Department Of CSE
M.Tech Scholar
DCRUST, Murthal

Parvinder Singh,Ph.D
Department Of CSE
Associate Professor
DCRUST, Murthal

ABSTRACT
Data Hiding is one of the challenging issues in the field of Network Security. Unlike cryptography, Steganography is used to hide the existence of secret message by embedding the message behind any cover object like image, text, audio, video files. Various authors proposed, various methods for hiding secret information behind gray scale images such as least significant method, gray level modification, pixel value differencing, pixel mapping method and pixel mapping method with BPCS, but all these method are not up to the marks that means increasing the embedding capacity of Stego-Image and to provide Stego-Image with an imperceptible quality are still challenges. To get better imperceptible quality, we proposed an enhanced technique "An Enhanced Data Hiding Approach Using Pixel Mapping Method (PMM) With Optimal Pixel Substitution Approach " that provides a better Peak Signal to noise ratio(PSNR) between Cover-Image and Stego-Image with good embedding capacity. The proposed approach is based on four modules – mapping rules, set classifier method, pixel selection method, and minimum differencing function to hide data within an image. This method works by selecting a set of pixels; map secret data into these selected pixels according to mapping rules and produces new Stego pixel value after mapping secret message according to Minimum Pixel Difference function. This integrated proposed approach provides more security to secret data as without knowing the mapping rules and locations of pixels no one could extract the secret data. This proposed approach not only provides larger embedding capacity but also produces an acceptable Stego image quality that can be seen by human eyes.

Keywords
Steganography, Information Hiding, Pixel Mapping, Pixel Value Differencing, Gray Scale Image, Cover Image, Method, Optimal Substitution, Stego Image**.**

1. INTRODUCTION
In Network Security, Data hiding is a broad subject and often involves procedures which could be mathematically complex, but central idea behind an information hiding is quite simple. Steganography is a most powerful data hiding technique for hiding secret information within other files. The word Steganography is a combination of two Greek words – Steganography = STEGANOS (Covered) + GRAPHIE (Writing) [12]. Steganography is an art and science of secret communication which is used to conceal secret information behind any cover media like text, image, audio and video. files in such a way that prevent an unauthorized user to detect hidden message.[10],[1],[3].

A Steganography technique has two main characteristics – Steganographic capacity and imperceptibility. Now days, it is very difficult to increase the embedding capacity of Stego-Image as well as enhancing the imperceptibility of a Stego-Image. Hence, this proposed approach improves the fundamental characteristics of image Steganography. The ultimate goal of Steganography is to hide existence of secret message behind any cover multimedia object and to create a covert communication channel to protect information during transmission from being stolen. [1].

Generic Form of Steganography at Sender Side

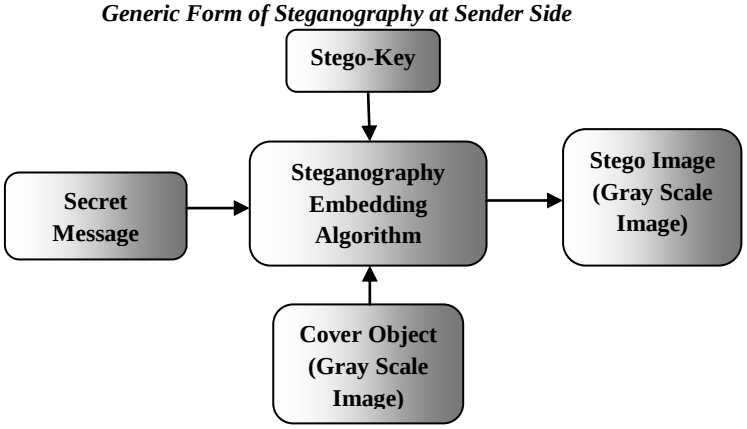

Figure - 1

Generic Form of Steganography at Receiver Side

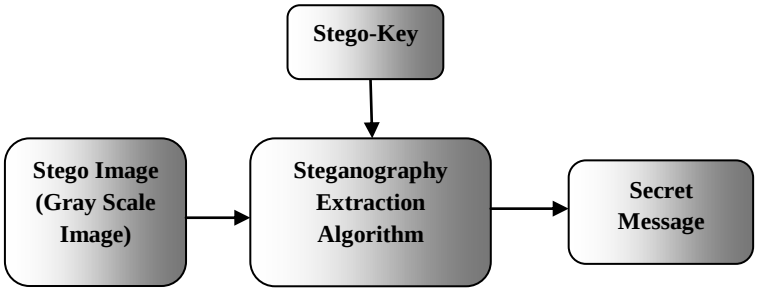

Figure - 2

The basic principle of Steganography is carried out into two phases – 1. Sender Side Phase 2. Receiver Side Phase. In Sender Side Phase, Steganography Embedding Algorithm will takes three inputs - i.) Secret Message ii.) Cover Image iii.) Stego-Key. After embedding data behind cover image embedding algorithm produce a new Stego image as output.

This newly generated Stego Image is transmitted to the designated Receiver over communication channel. In Receiver Side Phase, Steganography Extraction Algorithm will also takes two inputs – i.) Stego Image ii.)Stego-Key. After processing Stego image with the help of Stego-Key extraction algorithm will produce original secret message as output.. Mostly image and audio files are used to hide secret information due to their degree of redundancy. Before embedding data we need to check whether neighbours of seed pixel lie at image's boundary or not [12], [9]. This paper is organized into several following sections - Section – II describes some related existing work in the field of data hiding. Section –III deals with the details of proposed method. Section – IV describe algorithms to embed and extract secret information Section – V describe conclusion and future work. Steganography is different from cryptography so both can be seen as complement to each others. In cryptography plaintext/secret message is encoded into cipher form so that an attacker cannot easily decode cipher text into original secret message on the other hand Steganography is used to hide the existence of secret payload [18].

2. RELATED WORKS

2.1 Data Hiding By Least Significant Method

Various information hiding techniques in spatial domain, to hide secret information in least significant bits of pixel have been proposed [13], [5], [17]. In LSB methods, secret message can be embedded by replacing insignificant or redundant parts of a cover image. Basically, in LSB technique of Steganography, bits of secret message are substituted into the least significant bits of cover image's pixel. The major advantages of substitution technique is easy to use and its simplicity but LSB techniques are highly vulnerable because slightly modifications in least significant bits of cover image can be destroy entire message.

Simple LSB techniques are easier to implement but produces low quality Stego images. So to overcome this problem Wang et al. [16] proposed a genetic algorithm to embed secret data inside host image but this algorithm requires large computation time for approximate solutions. Chang et al. [15] proposed dynamic approach to reduce computational time. This approach selects best solution from all possible solutions. In 2010 Wang et al. [6] proposed new scheme called Transforming LSB substitution method to overcome problem associated with above two approaches.

2.2 Data Hiding By Pixel Value Differencing Method

In 2003 Wu and Tsai [14] proposed a method for high embedding capacity and good quality of Stego image called Pixel Value Differencing approach. In PVD cover image is divided into non overlapping blocks, which contains two connecting pixels (Pi, Pi+1). For each block in cover image, calculate difference between Pi+1 and Pi i.e. di = Pi+1 – Pi. Therefore, block with large di considered as block with sharp edges and block with large di considered as block with smooth area. It means that, more data is hidden inside blocks with sharp edges than smooth area. It uses a range table from 0 to 255, to map data in two consecutive pixels. This same range table is used at receiving end to recover original message. The width Wk of Rk decides how many bits can be embedded into consecutive pixels. Many techniques have been proposed based on PVD [8], [3], [4], [14]. In 2008 Chang et. al. [11] proposed a new method based on PVD called Tri Pixel Value Differencing approach which produces better embedding capacity than PVD approach. In tri value PVD, three different directional edges are used to embed secret data such that it achieves superior embedding capacity than the PVD method. In 2010, M.B. Ould Medeni et. Al [3] proposed a novel approach based on PVD called four value differencing approach which also produces better Embedding capacity over PVD method. In this method cover image is divided into equal size blocks, on the basis of number of 1's in left four bits of pixel, message is embedded into the edge of the block.

2.3 Data Hiding By Gray Level Modification Method

In 2004 Potdar et al. [7] proposed a new technique to hide data by changing the gray level values of the gray scale image pixels called Gray Level Modification technique. In this approach one to one mapping is used on the basis of even and odd numbers to hide data.

Example – Consider Data Bits = 10101100

Cover Image Pixels = {11, 14, 17, 19, 22, 27, 42, 55}

Step 1 – Change All Odd Gray Values into Even Gray Values

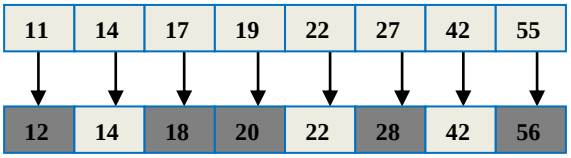

Figure - 3

Step 2 - Changes Modified Gray Values Based On Data Bits

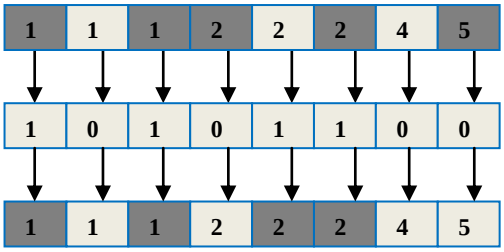

Figure - 4

Step 3 – Extracting Data Bits from Gray Values At Receiving End

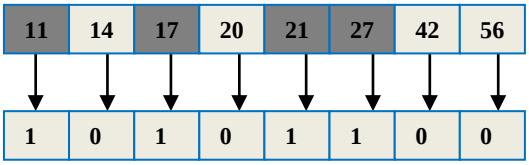

Figure - 5

2.4 Data Hiding By Ahmad et. Al. Method

In 2009 Ahmad et al. [10] proposed a new improved technique over GLM by dividing the cover image into equal size blocks and embed data in the edge of the block depending upon the number of ones in left four bits of pixel. In this approach, for each block in cover image, calculate difference between two consecutive pixels Pi+1 and Pi i.e. di=|Pi – Pi+1|. If di >= B then embed information inside Pi, Pi+1 on the basis of following mapping rules given in table -1. Here in figure –

6, we divide the cover image pixel into two equal parts for embedding secret data on the basis of number of 1's in most part of pixel.

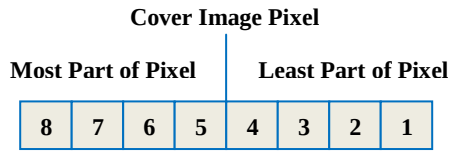

Figure – 6

Table - 1 Data hiding Mapping Rules B of Ahmad et. al Method

No. of 1's in left 4 bits of pixel(Most Part)	No. Of bits can be embedded in right 4 bits of pixel (Least Part)
1 or 3	3 bits
2	2 bits
0 or 4	1 bit

2.5 Data Hiding By Pixel Mapping Method

In 2010 Bhattacharyya, synal *et.al* proposed a new method [9],[12] to map data into image called pixel mapping method. It uses concept of pixel intensity and no of one's in pixel to map data. This approach produces better embedding capacity and PSNR Value over PVD, GLM and *ahmad et. Al* methods. In 2011 Bhattacharyya *et.Al* proposed PMM method with BPCS [2] which produces better image quality over PMM method.

3. DESCRIPTION OF PROPOSED STEGANOGRAPHY SCHEME

In this chapter, author discusses an enhanced proposed method for secret information hiding within spatial domain for gray level images. This proposed approach can be considered as an improved version of [12], [7]. The process of data hiding in this proposed approach is divided into following subsections namely– Pixel Selection Method, Information Hiding Mapping Rules, Pixel Sets Classifier Method, Minimum Pixel Value Difference Method.

3.1 Pixel Selection Method

In our proposed method we are sequentially selecting pixels to embed message bits into selected pixel. We can also use a random function 2r+ 5 % width to select pixels in random manner where r represents row of image. By using random locations we can improve the security of secret message but it will degrade the embedding capacity.

3.2 Pixel Sets Classifier Method

In this section author proposed a method to divide pixels set into subsets of pixels based on pixel's intensity and parity. For embedding data bits set classifier will divide pixels set into 4 pixels subsets –

Pixel$_{EE}$ = {Finite Set Of Those Pixels Having Even Intensity and Even Parity},

Pixel$_{EO}$ = {Finite Set Of Those Pixels Having Even Intensity and Odd Parity},

Pixel$_{OE}$ = {Finite Set Of Those Pixels Having Odd Intensity and Even Parity}

Pixel$_{OO}$ = {Finite Set Of Those Pixels Having Odd Intensity and Odd Parity}

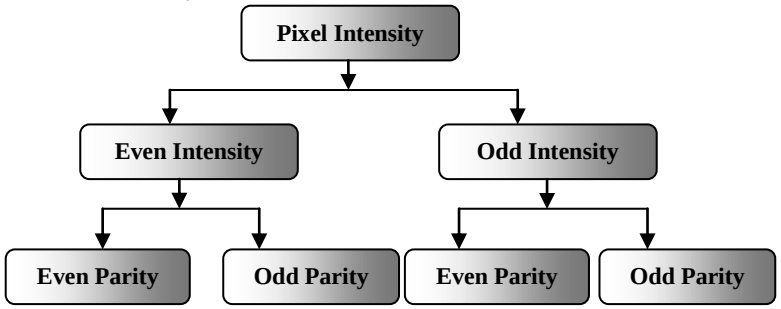

Figure -7

3.3 Data Hiding Mapping Rules

In this section some mapping rules are defined [12] on the basis of pixel intensity and its parity. As we know that intensity and parity of a pixel can be even or odd. For embedding 2 bits data we will follow mapping rules given in table – 2-

- **Mapping Rule 1 –** if data bits are 00 then change the intensity of selected pixel into even intensity and make the parity of selected pixel is even.
- **Mapping Rule 2 –** if data bits are 01 then change the intensity of selected pixel into even intensity and make the parity of selected pixel is odd.
- **Mapping Rule 3 –** if data bits are 10 then change the intensity of selected pixel into odd intensity and make the parity of selected pixel is even.
- **Mapping Rule 4 –** if data bits are 11 then change the intensity of selected pixel into odd intensity and make the parity of selected pixel is odd.

Table - 2* Mapping *Rules for Hiding 2 Bits per Pixel

Message Bits (0^{th} /1^{st} bits pair)	Pixel Intensity	Parity
00	Even	Even
01	Even	Odd
10	Odd	Even
11	Odd	Odd

For embedding 4 bits data we will follow mapping rules given in table – 3, where mask the 0^{th} and 1^{st} pairs of data bits into 5^{th} and 6^{th} position of pixel and change the intensity and parity of pixel on the basis of 2^{nd} and 3^{rd} pair of message bit.

Table -3 Mapping Rules for Hiding 2 Bits per Pixel

Message Bits (2^{nd} /3^{rd} bits and 0^{th} /1^{st} bits pair)		Pixel Intensity	Parity
00	00	Even	Even
	01	Even	Even
	10	Even	Even
	11	Even	Even
01	00	Even	Odd
	01	Even	Odd
	10	Even	Odd
	11	Even	Odd
10	00	Odd	Even
	01	Odd	Even
	10	Odd	Even
	11	Odd	Even
11	00	Odd	Odd
	01	Odd	Odd
	10	Odd	Odd
	11	Odd	Odd

3.4 Minimum Pixel Difference Method

When a pixel is selected to embed data bits then this method returns new minimum difference Stego pixel from specified pixel's subsets on the basis of data bits and mapping rules.

Flow Chart of Proposed Method for Embedding 2 Bits Data –

Let Data = | D0 | D1 | Seedr'c' = Selected Neighbour Pixel of Seed Pixel. Figure – 8 shows flow chart for embedding 2 bit data –

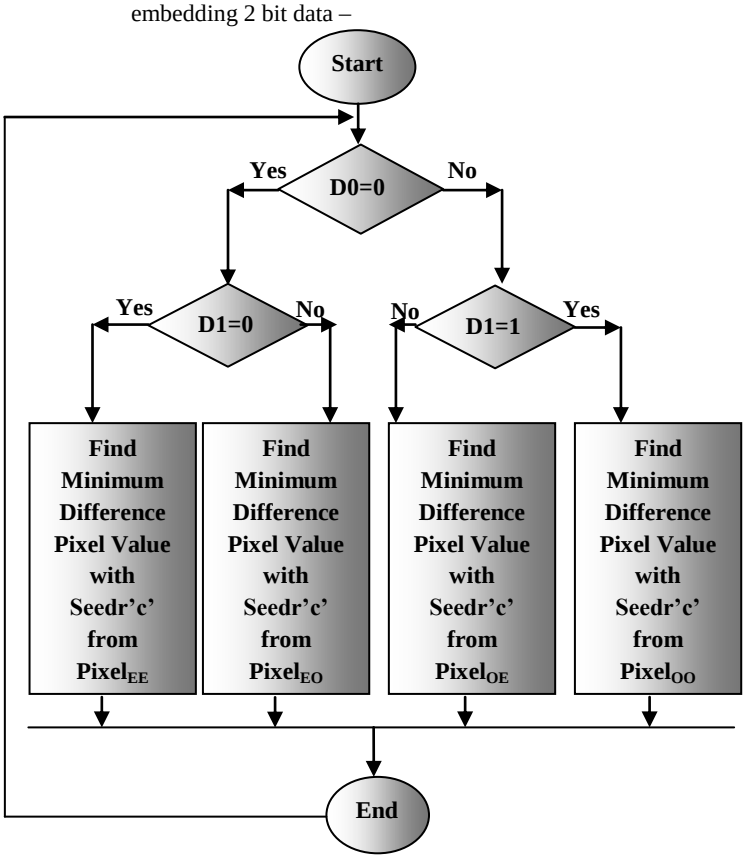

Figure - 8

Flow Chart of Proposed Method for Extraction 2 Bits Data –

Let S_Pix is Stego Image Pixel from which data is to be extracted. D0 & D1 are data bits.

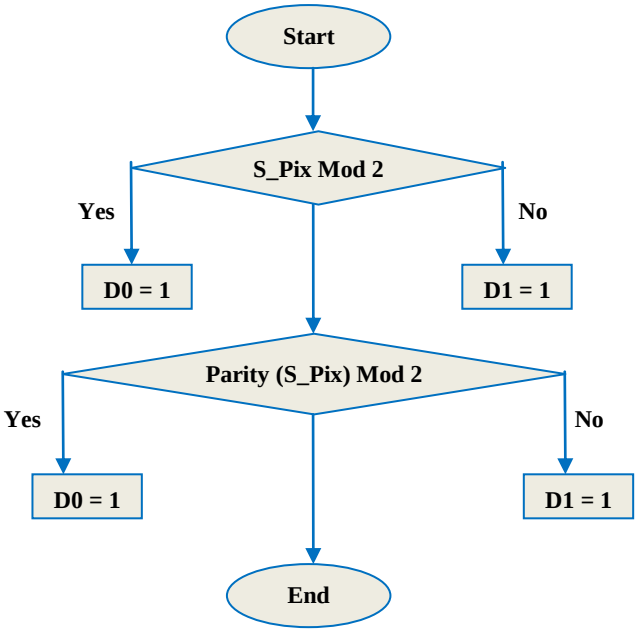

Figure - 9

Flow Chart of Proposed Method for Embedding 4 Bits Data –

Let Data = | D0 | D1 | D2 | D3 |, Seedr'c' = Selected Neighbour Pixel of Seed Pixel. Figure – 10 shows flow chart for embedding 4 bit data –

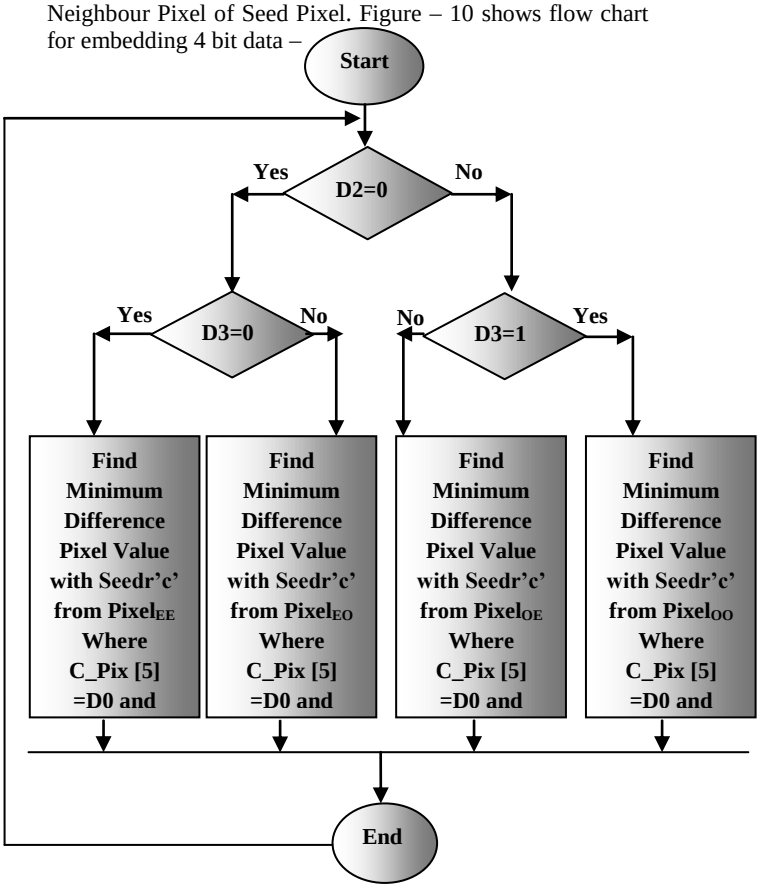

Figure - 10

Flow Chart of Proposed Method for Extraction 2 Bits Data – Let S_Pix is Stego Image Pixel from which data is to be extracted. D0, D1, D2, D3 are data bits.

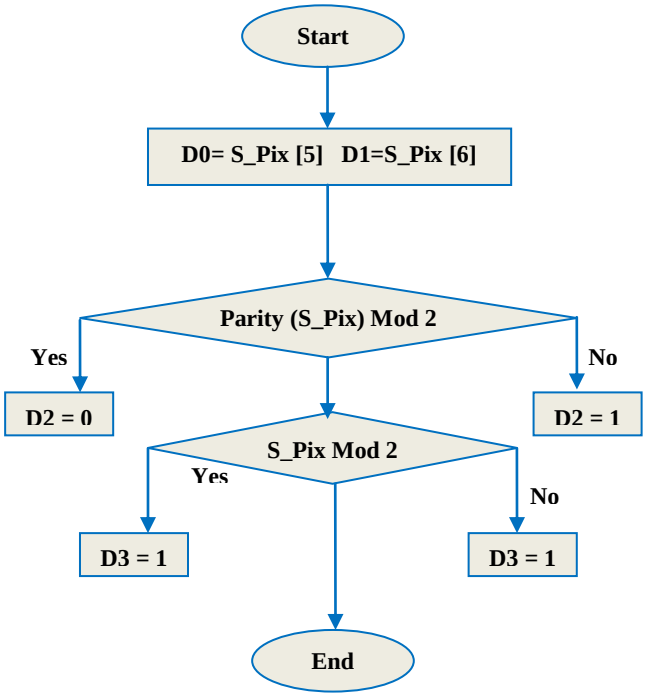

Figure - 11

4. PROPOSED ALGORITHM FOR EMBEDDING AND EXTRACTION DATA BITS

Parameters – Let 'Cimg'= it is 8 bit original gray scale image Of Size M X M where Cimg = {$Pixel_{ij}$ | 0 ≤ i < M, 0 ≤ j < M, and $Pixel_{ij}$ € 0, 1, 2, 3, 4 ..... 255}.
Data[n] = it can be the n bits secret message to be embeds behind cover image where Data[n] = {D_k | 0 ≤ k < n, D_k € 0, 1},
$Pixel_{EE}$ = {Finite Set Of Those Pixels Having Even Intensity and Even Parity},
$Pixel_{EO}$ = {Finite Set Of Those Pixels Having Even Intensity and Odd Parity},
$Pixel_{OE}$ = {Finite Set Of Those Pixels Having Odd Intensity and Even Parity}
$Pixel_{OO}$ = {Finite Set Of Those Pixels Having Odd Intensity and Odd Parity}
The process of embedding secret message will be finished when all n bits are embedded into Cimg.

4.1 Algorithm of Proposed Method for Embedding Two Bits Data at Sender

Input: Cover Image (Cimg), Secret Message (Data[n]), Count, D [2];
Initialize count=0, n=length (Data), D [2] = {0, 0};
Step - 1 Select Cover Image in Which you Want to Hide Data.
Step – 2 Read Message From Txt File And Convert Secret Message Into Binary.
Step – 3 Compute Length of Secret Message In Binary.
Step – 4 Select Cover Image Pixel (C_Pix) On The Basis Of Pixel Selection Method. If Selected Pixel Is Lies On The Boundary Then Ignore It and Select Another Pixel.
Step – 5 Read First Two Bits of Message into D0 And D1.
Step – 6 If (D0==0 && D1== 0) Then
Find Minimum difference Pixel with C_Pix from Pixel Set PixelEE
Else
If (D0== 0 && D1== 1) Then
Find Minimum difference Pixel with C_Pix from Pixel Set PixelEO
Else
If (D0== 1 && D1== 0) Then
Find Minimum difference Pixel with C_Pix from Pixel Set PixelOE
Else
If (D0== 1 && D1== 1) Then
Find Minimum difference Pixel with C_Pix from Pixel Set PixelOO
Step – 7 Repeat Steps from 4 To 6 Until Secret Message Is Embedded.
Step – 8 Return Stego Image & End.

4.2 Algorithm of Proposed Method for Extraction Two Bits Data at Receiver

Input: Stego Image (Simg), Secret Message (Data[n]), Count, D [2];
Initialize count=0, n=length (Data), D [2] = {0, 0};
Step – 1 Select Stego Image from Which you Want to Extract Data.
Step – 2 Select Stego Image Pixel (S_Pix) On The Basis Of Pixel Selection Method. If Selected Pixel Is Lies On The Boundary Then Ignore It and Select Another Pixel.
Step – 3 If (S_Pix Mod 2 == 0) Then
D0 = 0;
Else
D0 = 1;
Step – 4 If (Parity (S_Pix) Mod 2 == 0) Then
D0 = 0;
Else
D0 = 1;
Step – 4 Repeat Steps from 2 To 3 Until Secret Message Is Extracted.
Step – 5 End.

4.3 Algorithm of Proposed Method for Embedding Four Bits Data at Sender

Input: Cover Image (Cimg), Secret Message (Data[n]), Count, D [4];
Initialize count=0, n=length (Data), D [4] = {0, 0, 0, 0};
Step - 1 Select Cover Image in Which you Want to Hide Data.
Step – 2 Read Message From Txt File And Convert Secret Message Into Binary.
Step – 3 Compute Length of Secret Message in Binary
Step – 4 Select Cover Image Pixel (C_Pix) On The Basis Of Pixel Selection Method. If Selected Pixel Is Lies On The Boundary Then Ignore It and Select Another Pixel.
Step – 5 Read First Four Bits of Message into D0, D1, D2 And D3.
Step – 6 If (D2==0 && D3== 0) Then
Find Minimum difference Pixel with C_Pix from Pixel Set PixelEE, & C_Pix [5] == D0 and C_Pix [6] ==D1
Else

If (D2== 0 && D3== 1) Then
Find Minimum difference Pixel with C_Pix from Pixel Set PixelEO, & C_Pix [5] == D0 and C_Pix [6] ==D1
Else
If (D2== 1 && D3== 0) Then
Find Minimum difference Pixel with C_Pix from Pixel Set PixelOE, & C_Pix [5] == D0 and C_Pix [6] ==D1
Else
If (D2== 1 && D3== 1) Then
Find Minimum difference Pixel with C_Pix from Pixel Set PixelOO, & C_Pix [5] == D0 and C_Pix [6] ==D1
Step – 7 Repeat Steps from 4 To 6 Until Secret Message Is Embedded.
Step – 8 Return Stego Image & End.

4.4 Algorithm of Proposed Method For Extraction Four Bits Data at Receiver

Input: Stego Image (Simg), Secret Message (Data[n]), Count, D [4];
Initialize count=0, n=length (Data), D [4] = {0, 0};
Step – 1 Select Stego Image from Which you Want to Extract Data.
Step – 2 Select Stego Image Pixel (S_Pix) On The Basis Of Pixel Selection Method. If Selected Pixel Is Lies On The Boundary Then Ignore It and Select Another Pixel.
Step – 3 D0 = S_Pix [5];
D1=S_Pix [6]
Step – 4 If (S_Pix Mod 2 == 0) Then
D2 = 0;
Else
D2 = 1;
Step – 5 If (Parity (S_Pix) Mod 2 == 0) Then
D3 = 0;
Else
D3 = 1;
Step – 6 Repeat Steps from 2 To 4 Until Secret Message Is Extracted.
Step – 7 End.

5. EXPERIMENTALS RESULTS

The proposed method for data hiding has applied on several images and we have compared the result of our method with other image Steganography methods such as GLM, PMM 2Bit and PMM 4 Bit. To evaluate the effectiveness of our proposed method, we have used the Embedding Capacity and PSNR. By comparing the Embedding Capacity and PSNR of proposed method with GLM, PMM 2Bit and PMM 4 Bit, mathematically we have proved that proposed method is better than GLM, PMM 2Bit and PMM 4 Bit. In this section we present the results on three images (Lena, Cameraman, Barbara) after embedding secret data. Figure – 12 show Stego image of Lena 512 X 512 after mapping **I am an Indian.**

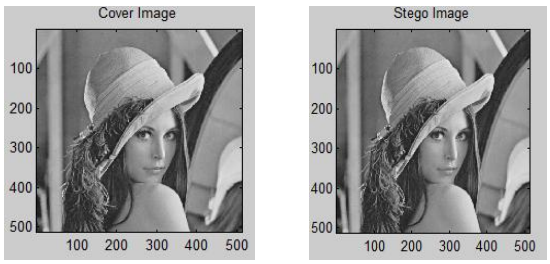

Figure - 12

Figure – 13 show Stego image of Cameraman 512 X 512 after mapping **I am an Indian, India is my country.**

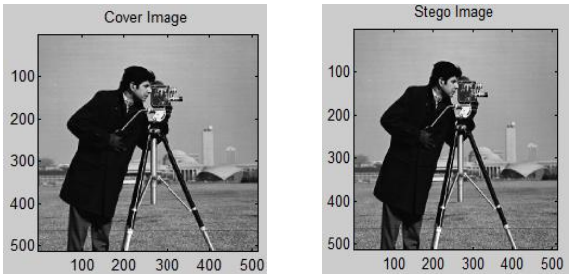

Figure-13

Figure – 14 show Stego image of Barbara 512 X 512 after mapping I am an Indian and i feel proud to be an Indian.

Figure – 14

Mean Squared error (MSE) is used to measures the average squared difference between Cover-Image and Stego-Image i.e. it measures difference between actual output and desired output. Smaller MSE is better. MSE is calculated by -

$$\mathbf{MSE} = \frac{1}{height * width} * \sum_{i=1}^{height} \sum_{j=1}^{width} [C(i,j) - S(i,j)]\text{^}2$$

Where $C(i,j)$, is cover image, S(i, j) is the Stego image, height is maximum no. of rows in image and width is maximum no. of columns in image.

Peak signal to noise ratio (PSNR) is used to measure the quality of Stego-image after embedding secret data in Cover-image i.e. it measures percentage of hidden data to the percentage of image, greater PSNR is better. PSNR is calculated by

$$\mathbf{PSNR} = 20 \log \left(\frac{I_{max}}{\sqrt{MSE}} \right)$$

Where value of I_{max} is 255 for 8 bit gray scale images because maximum value can be defined by using 8 bit is 255.

Table - 4 Shows Embedding Capacity of Different Algorithms (2 bits per pixels)–

Image	Image Size	GLM	PMM 2 Bit	Proposed Method 2 Bit
LENA	512X512	32768	65536	65536
	256X256	8192	16384	16384
	128X128	2047	4094	4094
CAMERAMAN	512X512	32752	65504	65504
	256X256	8192	16384	16384
	128X128	2047	4094	4094
BARBARA	512X512	32768	65536	65536
	256X256	8192	16384	16384
	128X128	2048	4096	4096

Table - 5 Shows Embedding Capacity of Different Algorithms (4 bits per pixels)–

Image	Image Size	GLM	PMM 4 Bit	Proposed Method 4 Bit
LENA	512X512	32768	131072	131072
	256X256	8192	32768	32768
	128X128	2047	8188	8188
CAMERAMAN	512X512	32752	13100	13100
	256X256	8192	32768	32768
	128X128	2047	8188	8188
BARBARA	512X512	32768	131072	131072
	256X256	8192	32768	32768
	128X128	2048	8186	8186

Table - 6 Shows MSE and PSNR Values of Different Algorithms (2 bits per pixels) by Embedding 2000 characters in 128X128, 256X256, and 512X512 images respectively –

Image	Image Size	GLM		PMM 2 Bit		Proposed Method 2 Bit	
		MSE	PSNR	MSE	PSNR	MSE	PSNR
LENA	512X512	0.0306	63.2761	0.0777	59.2289	0.0669	59.8794
	256X256	0.1220	57.2658	0.3070	53.2595	0.2527	54.1053
	128X128	0.4839	51.2828	1.2452	47.1785	1.0845	47.7784
CAMERAMAN	512X512	0.0307	63.2653	0.0776	59.2342	0.0636	60.0992
	256X256	0.1227	57.2441	0.3050	53.2879	0.2560	54.0477
	128X128	0.4856	51.2681	1.2438	47.1834	1.0416	47.9537
BARBARA	512X512	0.0306	63.2777	0.0750	59.3790	0.0634	60.1083
	256X256	0.1201	57.3348	0.3011	53.3432	0.2527	54.1053
	128X128	0.4891	51.2365	1.2081	47.3099	1.0508	47.9154

Table - 7 Shows MSE and PSNR Values of Different Algorithms (4 bits per pixels) by Embedding 2000 characters in 128X128, 256X256, and 512X512 images respectively –

Image	Image Size	GLM		PMM 4 Bit		Proposed Method 4 Bit	
		MSE	PSNR	MSE	PSNR	MSE	PSNR
LENA	512X512	0.0306	63.2761	0.6356	50.0989	0.4043	52.0639
	256X256	0.1220	57.2658	2.7372	43.7578	1.5802	46.1438
	128X128	0.4839	51.2828	10.6140	37.8720	7.4832	39.3900
CAMERAMAN	512X512	0.0307	63.2653	0.6063	50.3043	0.4432	51.6645
	256X256	0.1227	57.2441	2.5149	44.1256	1.8645	45.4252
	128X128	0.4856	51.2681	10.7513	37.8162	8.1234	39.0334
BARBARA	512X512	0.0306	63.2777	0.6516	49.9907	0.4419	51.6778
	256X256	0.1201	57.3348	2.5425	44.0781	1.6644	45.9183
	128X128	0.4891	51.2365	10.4737	37.9298	7.9297	39.1382

Table - 8 Shows Comparison between PMM and Proposed Method –

Data Bits	Pixel Value	Stego Pixel (PMM)	Change In Pixel By PMM	Stego Pixel (Proposed Method)	Change In Pixel By Proposed Method
0010	12	17	+5	9	-3
1100	34	46	+12	30	-4
1010	95	85	-10	101	+6
1101	97	110	+13	94	-3
1011	96	109	+13	93	-3
0000	95	80	-15	96	+1
0011	94	81	-13	97	+3
1011	97	109	+12	93	-4
1011	16	21	+5	13	-3
1111	17	31	+14	7	-10
1110	64	71	+7	63	-1
1111	81	87	+6	79	-2

6. FUTURE WORKS AND CONCLUSION

This is an efficient approach to map secret message into gray scale images to provide better image quality and information embedding capacity. This enhanced approach can also be used to embed 8 bits data by extending mapping rules. Key advantages of this approach are – unauthorized person cannot retrieve data without the knowledge of mapping rules, can provide better security by mapping data into randomly selected pixels, it has low computational overhead over other Steganography approaches because it does not require transform of images into frequency domain.

Future work of this approach will consider following modifications –

- Relate encryption with this approach in which message is encrypted by using random mealy machine to increase security before embedding data into cover image.
- Investigating this method on colour images.
- Modifying this approach by using wavelets.

7. REFERENCES

[1] Souvik Bhattacharyya and Gautama Sanyal. Study and analysis of quality of service in different image based Steganography using PMM. International journal of applied information system – foundation of computer science, New York, USA 2012

[2] Souvik Bhattacharyya and Gautam Sanyal. PMM (Pixel Mapping method) Based Bit plane complexity segmentation (BPCS) Steganography. 978-1-4673-0126-8/11/$26 2011 – IEEE

[3] M.B MEDENI and EI M. Souidi. A novel Steganographic method for gray – level images with four-pixel differencing and LSB Substitution 978-1-61284-7332-0/11/$26 2010 – IEEE

[4] J.K Mandal and Debashis Color image Steganography based on pixel value differencing in spatial domain – international journal of information science and techniques July 2012.

[5] Wang Yan And Ling-di Ping. A new Steganography algorithm based on spatial domain, 978-0-7695-3991-1/09/$26 2009 – IEEE

[6] Transforming LSB substitution for image based Steganography in matching algorithms. Journal of information science and engineering 26, 1199-1212 2010

[7] Potdar V.and Chang E. Gray level modification Steganography for secret communication. In IEEE International Conference on Industrial Informatics., pages 355–368, Berlin, Germany, 2004.

[8] Chung-Ming Wang, Nan-I Wu ,Chwei-Shyong Tsai and Min-Shiang Hwang A high quality Steganographic method with pixel value differencing and modulus function. The journal of system and software – science direct – 2007

[9] Souvik Bhattacharyya. and Gautam Sanyal. Hiding data in images using pixel mapping method (pmm). In Proceedings of 9th annual Conference on Security and Management (SAM) under The 2010 World Congress in Computer Science, Computer Engineering, and Applied Computing (World Comp 2010), Las Vegas, USA, July 12-15,2010.

[10] Ahmad T. Al-Taani. and Abdullah M. AL-Issa. A novel Steganographic, method for gray-level images. International Journal of Computer, Information, and Systems Science, and Engineering, 3, 2009.

[11] P Huang. K.C. Chang., C.P Chang. and T.M Tu. A novel image Steganography method using tri-way pixel value differencing. Journal of Multimedia, 3, 2008.

[12] Souvik Bhattacharyya, lalan kumar and Gautam Sanyal. A novel approach of data hiding using PMM(Pixel Mapping method). International journal of computer science and information security – vol 8. No. 4 2010.

[13] C.K. Chan. and L. M. Cheng. Hiding data in images by simple lsb substitution. Pattern Recognition, 37:469–474, 2004.

[14] D.C. Wu. and W.H. Tsai. A steganographic method for images by pixel value differencing. Pattern Recognition Letters, 24:1613–1626, 2003.

[15] J.Y. Hsiao. C.C. Chang. and C.-S. Chan. Finding optimal least significant- bit substitution in image hiding by dynamic programming strategy. Pattern Recognition, 36:1583–1595, 2003.

[16] C.F. Lin. R.Z. Wang. and J.C. Lin. Image hiding by optimal lsb substitution and genetic algorithm. Pattern Recognition, 34:671–683,2001.

[17] Y. K. Lee. and L. H.Chen. High capacity image steganographic model. IEE Proc.-Vision, Image and Signal Processing, 147:288–294, 2000.

[18] Ross J. Anderson. and Fabien A.P.Petitcolas. On the limits of steganography. IEEE Journal on Selected Areas in Communications (J-SAC),Special Issue on Copyright and Privacy Protection, 16:474–481, 1998.