International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-1 Issue-14 March-2014

Text Hiding Scheme Using Mapping Technique for Spatial Domain

Tanusree Podder[1], Lalita Kumari[2], Abhishek Majumder[3]

Abstract

In this paper, a new steganographic method for spatial domain has been proposed, which includes a new mapping technique for the secret message. The algorithm selects random pixels to hide the mapped secret message. A "seed" is used as a secret key to randomly select pixel from the cover image, which successively helps to spread the message bits over the entire image. The result of the proposed algorithm is analyzed and discussed using MSE, PSNR, SC, AD, MD and NAE. The histogram of the cover and stego image is also shown.

Keywords

Steganography, Least Significant Bit, Spatial Domain, MSE, PSNR, SC, AD, MD, NAE.

1. Introduction

There is no doubt about the immense scope for the internet technology with respect to data collection, information availability, transfer of information, maintenance; in other words almost everything related to keeping, processing and manipulation of information is done by internet. Internet is upgrading as the largest medium to send and receive message of non-importance or utmost weight, somewhere, data hiding for secure reason came to be necessitate. These situations provoke the concept of 'transforming' secret information from one to another where the resulting data can be understood only by those who knows how to get the original secret information i.e., encryption. But, with encryption, data though unreadable still remains as data which is somewhat vulnerable to easy decryption.

This work was supported in part by the Department of Computer Science and Engineering, National Institute of Technology, Agartala.

Manuscript received February 05, 2014.

Tanusree Podder, Computer Science and Engineering, National Institute Of Technology, Agartala, India.

Lalita Kumari, Computer Science and Engineering, National Institute Of Technology, Agartala, India.

Abhishek Majumder, Computer Science & Engineering at SSCET, Badhani, Punjab, India.

This problem came to be solved by 'steganography' [1]. Steganography is one such pro-security advance in which secret data is embedded in a cover, where the actual message to be sent is completely changed to a new form, hidden under a cover and being sent to the destination. Only those who know the technique used to encrypt the message can recover the message. The performance of various steganographic methods can be rated by three Parameters: security, capacity and imperceptibility [2]. Many Steganographic algorithms are developed to perform secret communication. The noisy information of cover work has been used to hide covert communication. The steganographic concept can be split two categories, Statistics-aware Steganography and Model-based Steganography. Statistics-aware Steganography considers all statistical aspects which are used by steganalyst to detect the presence of hidden message to resist all kind of attack and get a sound stegogramme. In Model-based Steganography preserves a chosen model of cover medium. The main aspire is to built a steganographic system where changes are as minimal as possible. Good steganographic algorithm requires perceptibility, capacity, robustness and speed. In spatial domain, two different kind of embedding schemas are possible sequential and randomized. The algorithm starts from the first pixel of cover work in sequential embedding until the secret message gets over. Randomized process scatters the secret message in random location of cover work.

2. Literature Review

LSB is one of the widely popular steganographic techniques. In LSB Steganography, the least significant bits of the cover media's digital data are used to conceal the message. The simplest of the LSB techniques is LSB replacement. LSB replacement Steganography flips the last bit of each of the data values to reflect the message that needs to be hidden. The embedding process starts from the first pixel onwards. Let us consider a pixel containing binary value '10101100' and a secret message bit '1' is need to be hidden inside the pixel, so the new pixel value will be '10101101'.Where the LSB of the pixel contains the secret bit value. Similar to LSB replacement, LSB matching[3] uses the LSB's of

cover work to hide the secret message but the method decrements the odd value by 1 or else leave them unmodified ,while increments the even value by 1 or else leave them unmodified. Unlike LSB replacement, LSB matching creates more imbalances after hiding in cover image. Selected Least Significant Bits (SLSB) [4] algorithm first applies compression to the secret message and color selection, filtering on cover image .It hides the secret message in one color component instead of all three. At last applies lsb matching algorithm. First component alteration technique [5] uses a secret key which is stored at the beginning pixel and after that it hides 8bit of secret message in 1 pixel, it changes entire blue component to hide the secret message.

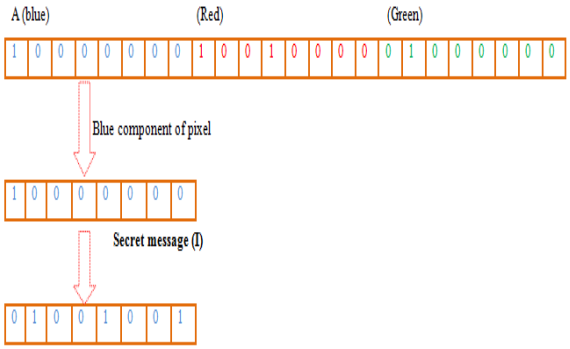

Figure 1: First Component Alteration Technique

LSB Array Based Image Steganography Technique [6] uses RSA algorithm to encrypt the secret message, divide it into 4 blocks and hides the secret message in 4 LSBs of cover image. Steganography Algorithm to Hide Secret Message inside an Image [7] transfers secret message into text file and converts the zipped text file, key into binary codes to hide it in last 2 bits of pixel. A New Method in Image Steganography with Improved Image Quality [8] searches for two identical bits of secret message in 8-bit pixel by dividing it into 4 2-bit parts. If the message bit doesn't find the identical match then technique hides the message into 2-lsb of the pixel.

3. Proposed Method

In the proposed algorithm, sender need to input secret message, a seed value for selecting random pixel and a cover image .Mapped secret message gets embedded into the cover image with the help of embedding algorithm. For extraction of the secret message, user needs to input the stego image, seed and message length. fig.1.and fig.2 Shows the embedding and extraction process of proposed algorithm. In the proposed algorithm, we are converting each character of secret message into 8bit binary and constructing a binary message array of secret message. Let the secret message which need to be embedded is of k-bit as $b_0, b_1, \ldots\ldots, b_{k-2}, b_{k-1}$ .Now ,we will take 4bit from the binary array(eg-b_0, b_1, b_2, b_3) and will map it into a predefined ASCII value . In this way, there can be 16 combination of mapping possible from 0000 to 1111.we will map each 4bit of secret message into an ASCII value means we will map each 4bit into 8bit value. After mapping we will select one pixel from cover image and hide 1-bit at a time in the 2^{nd} lsb of that pixel .If the first bit of the message is matching with the 2^{nd} -lsb of the pixel then we will not change the pixel value else if the message bit is 0 then we will make 2^{nd} lsb as 0 and 1^{st} as 1.If the message bit is 1 then we will make 2^{nd} lsb as 1 and 1^{st} as 0.To find the data close to the original minimum-error replacement method is used .In our proposed algorithm, replacing 1-bit data in the i-th LSB will create maximum error of $\pm 2^{i-1}$ and minimum 0.

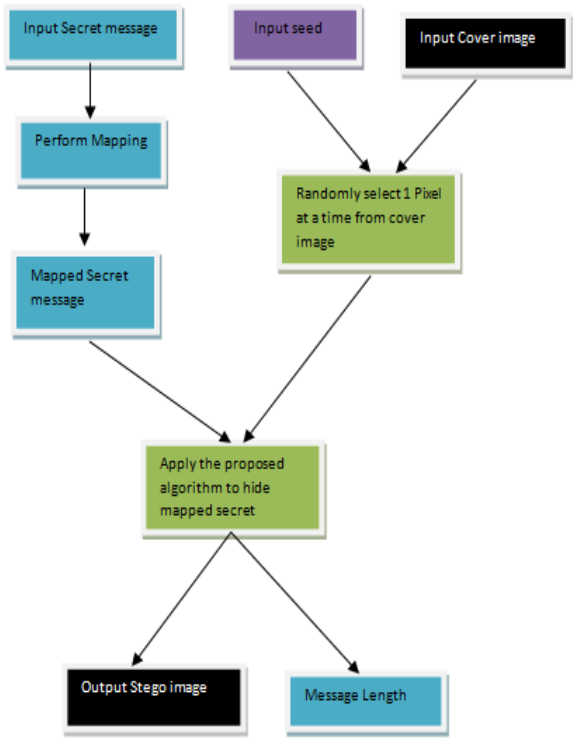

Figure 2: The embedding process of proposed algorithm

Figure 3: The extraction process of proposed algorithm

Table 1: Predefined Mapping Table

4-Bit Of Secret Message From Message Array	ASCIIVALUE Of The Character After Mapping	8-Bit Binary Representation Of The Character
0000	35	00100011
0001	38	00100110
0010	64	01000000
0011	87	01010111
0100	125	01111101
0101	91	01011011
0110	184	10111000
0111	198	11000110
1000	224	11100000
1001	181	10110101
1010	223	11011111
1011	246	11110110
1100	174	10101110
1101	190	10111110
1110	200	11001000
1111	207	11001111

A. Embedding process:

1. Let the secret message is "I will come to meet you on 21st dec, time and place will be same as previous.".
2. Converting a short part of secret message "I will" in binary, we will get "01001001001000000111011101101001011 0110001101100".
3. Take the first 4-bit b_0, b_1, b_2, b_3 and map it according to the predefined table. Repeat the same process until whole secret message is mapped.

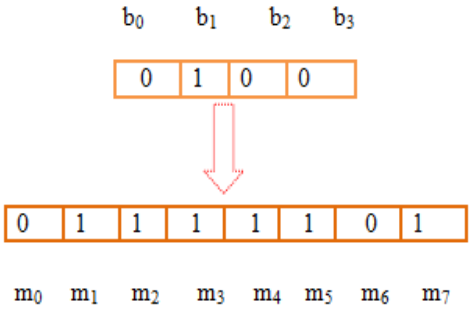

Figure 4: Mapping technique of secret message

4. After mapping we will get the mapped secret message as'01111101101101010100000000 10001111000110110001101011100010110101 01101110001010111010111000101011110'.
5. Now, take a random pixel using a "seed" from the cover image and select the value at 2nd LSB if it matches with the 1st bit m_0 of the mapped secret message then keep the pixel value same.

Case1: while $m_0 = 0$ and Pixel P_1=01010100.Here 2nd LSB is 0 and it is matching with m_0 .So, we won't change the pixel value.

Case2: while $m_0 = 0$ and Pixel P_1=10010010.Here 2nd LSB is 1 and it is not matching with m_0 .So , we will change the 2nd LSB as 0 and if the 1st LSB is 0 then make it 1.

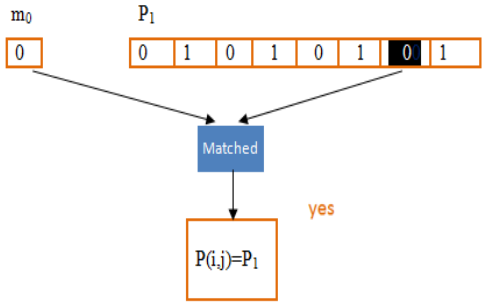

Figure 5: Case 1 of proposed method

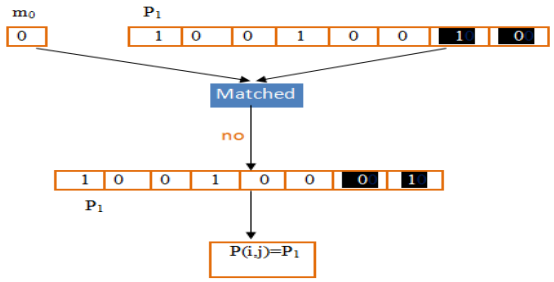

Figure 6: Case 2 of proposed method

Case3: while m_0 = 0 and Pixel P_1=10010011.Here 2nd LSB is 1 and it is not matching with m_0 .So , we will change the 2nd LSB as 0 and if the 1st LSB is 1 then no change .

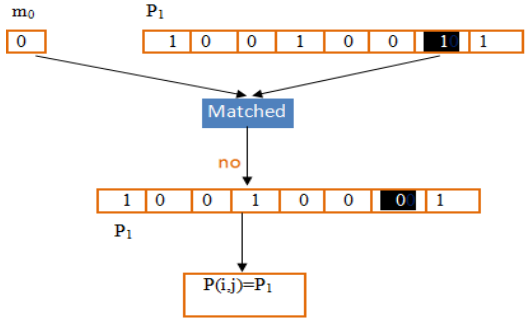

Figure 7: Case 3 of proposed method

Case4: while m_0 = 1 and Pixel P_1=10010001.Here 2nd LSB is 1 and it is not matching with m_0 .So , we will change the 2nd LSB as 0 and if the 1st LSB is 0 then make it 1. Case5: while m_0 = 1 and Pixel P_1=10010000.Here 2nd LSB is 0 and it is not matching with m_0 .So , we will change the 2nd LSB as 1 and if the 1st LSB is 0 then make no change.

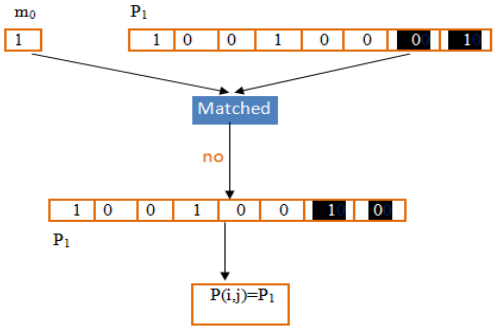

Figure 8: Case 4 of proposed method

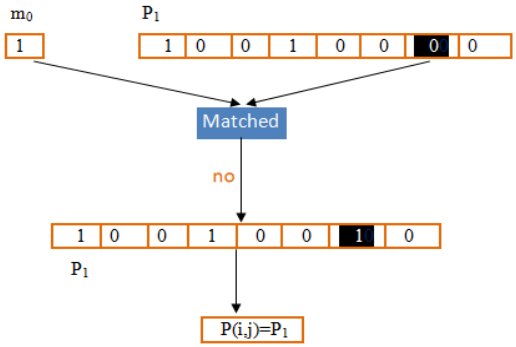

Figure 9: Case 5 of proposed method

6. Repeat step -5 until all mapped secret message m_0 , m_1,……,m_{2l-1} are emdedded.
7. We will get stego image as output.

A. Extraction Process:

1. Stego image,message length and seed are the inputs for extraction process.Seed generates pseudo-random number to randomly select pixels, used as a secret key.
2. Retrive the 2nd LSB of every random pixel in an array.After retriving 2nd LSB ,let the array contains '01111101101101010100000000100 01111000110110001101011100010110101101111000101011101011100001010111010111000101011110'.
3. Take the first 8-bit m'_0 , m'_1, m'_2, m'_3 , m'_4 , m'_5, m'_6, m'_7 from the array and map it back to 1 of the 4-bit value from 0000 to 1111 according to the table. After mapping combine 8-bit and get character from ASCIIvalue.
4. Repeat Step-3 until we get the secret message.

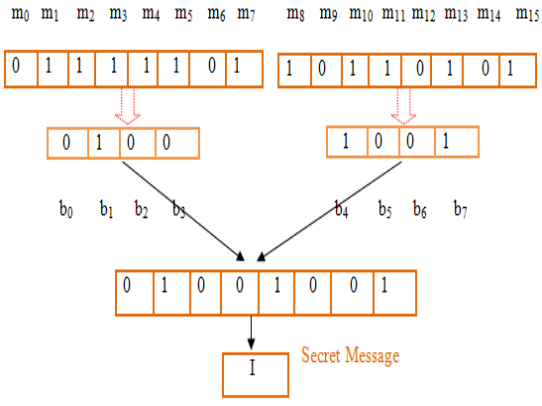

Figure 10: -Reverse mapping to get Secret Message

4. Proposed Algorithm

Embedding Algorithm:

I. Select a cover image and with the help a "seed" ,randomly select pixels from the cover image into an array called cover array.
II. Select a secret message and take 8-bit representation of each character into an array called message array.
III. Take 4-bit from the message array and map it into a predefined 8-bit value which will represent another character.
IV. After mapping the secret message we will get another array called mapped array.
V. Select 1-pixel at a time from the cover image and match the 2rd lsb of the pixel with the 1st bit from the mapped array.
VI. If they are matching keep the pixel as it is.
VII. Else if mapped bit is 0 and 1st LSB of cover image is 1
VIII. Make the 2rd LSB as 0.
IX. Else if mapped bit is 0 and 1st LSB of cover image is 0
X. Make the 2rd LSB as 0 and 1st LSB as 1.
XI. Else if mapped bit is 1 and 1st LSB of cover image is 0
XII. Make the 2rd lsb as 1
XIII. Else if mapped bit is 1 and 1st LSB of cover image is 1
XIV. Make the 2rd lsb as 1 and 1st lsb as 0
XV. Apply the same procedure until all the bits are mapped.
XVI. After embedding the message the resulting image will be a stego image.

Extraction Algorithm:

I. From the stego image select the same random pixel by giving the same seed and also give the length of the message as input.
II. Collect 2nd LSB of every pixel and store them in an array.
III. Take 8-bit from the array and map it back to 4-bit and store it. Repeat this process until full array is mapped back.
IV. After mapping again take 8-bit and convert it back to a character.
V. Repeat step-4 to get the secret message.

5. Result and Analysis

For analysis of the proposed algorithm we took 'peppers.png' as cover image and 1KB, 2KB, 3KB, 4KB data as secret message. After applying the proposed algorithm we got good quality image. Mean Square Error(MSE,)Peak Signal-to-Noise Ratio (PSNR) ,Structural Content(SC),Average Difference(AD), Maximum Difference(MD), Normalized Absolute error(NAE) are used to measure the image quality after embedding the secret message .Histogram of the Cover image is shown in Figure 11 and stego image is in Figure 12 after hiding 1KB,2KB,3KB and 4KB secret data.

a) Mean Square Error (MSE): Poor quality Stego image is produced after embedding the secret message if MSE gives higher value.

$$MSE = \frac{1}{M*N}\sum_{y=1}^{M}\sum_{y=1}^{N}[x(m,n)-y(m,n)]^2$$

Table 2: Result of MSE after Applying on Cover and Stego image

Serial NO	Size of Secret message	MSE (Red channel)	MSE (Green channel)	MSE (Blue channel)
1.	1KB (1024 bytes)	0.005044	0.004907	0.004716
2.	2KB (2048 bytes)	0.009979	0.009868	0.009610
3.	3KB (3072 bytes)	0.014902	0.014869	0.014452
4.	4KB (4094 bytes)	0.020092	0.019669	0.019231

From the table it is clear that the proposed algorithm produced less error after embedding.

b) Peak Signal to Noise Ratio (PSNR): Higher PSNR value indicates the proposed algorithm produced good quality stego image.

$$PSNR = 20\log_{10}[MAXPIX/RMSE]$$
$$RMSE = \sqrt{MSE}$$

Table 3: Result of PSNR after Applying on Cover and Stego image

Serial NO	Size of Secret message	PSNR (Red channel)	PSNR (Green channel)	PSNR (Blue channel)
1.	1KB (1024 bytes)	71.103467	71.222282	71.395304
2.	2KB (2048 bytes)	68.140062	68.188692	68.303486
3.	3KB (3072 bytes)	66.398401	66.408127	66.531569
4.	4KB	65.100644	65.192883	65.290676

From the table it is clear that the proposed algorithm produced higher PSNR after embedding.

c) Structural Content (SC): The Proposed algorithm produces poor quality stego image if SC gives higher value.

$$SC = \frac{\sum\sum(f(u,v))^2}{\sum\sum(f'(u,v))^2}$$

Table 4: Result of SC after Applying on Cover and Stego image

Serial NO	Size of Secret Message	SC (Red channel)	SC (Green channel)	SC (Blue channel)
1.	1KB (1024 bytes)	1.000000	0.999998	0.999997
2.	2KB (2048 bytes)	1.000000	0.999999	0.999991
3.	3KB (3072 bytes)	1.000000	0.999998	0.999994
4.	4KB (4094 bytes)	1.000000	0.999998	0.999988

From the table it is clear that the proposed algorithm produced low Structural Content value after embedding.

d) Average Difference (AD): Low value of AD indicates better quality of stego image produced by proposed algorithm.

$$AD = \frac{abs(\sum\sum(f(u,v) - f'(u,u)))}{m * n}$$

Table 5: Result of AD After Applying on Cover and Stego image

Serial NO	Size of Secret message	AD(Red channel)	AD (Green channel)	AD (Blue channel)
1.	1KB (1024 bytes)	0.002660	0.002587	0.002488
2.	2KB (2048 bytes)	0.005256	0.005227	0.005071
3.	3KB (3072 bytes)	0.007870	0.007869	0.007657
4.	4KB (4094 bytes)	0.010606	0.010421	0.010167

The table shows the proposed algorithm gives low AD in between original and stego image.

e) Maximum Difference (MD): The proposed algorithm is designed to give low MD. Large value of MD shows poor quality of stego image.

$$MD = \max(\max(|f(u,v) - f'(u,v)|))$$

Table 6: Result of MD after Applying on Cover and Stego image

Serial NO	Size of Secret Message	MD (Red channel)	MD (Green channel)	MD (Blue channel)
1.	1KB	2.000000	2.000000	2.000000
2.	2KB	2.000000	2.000000	2.000000
3.	3KB	2.000000	2.000000	2.000000
4.	4KB	2.000000	2.000000	2.000000

The table shows fix maximum difference after embedding secret message.

f) Normalized Absolute Error (NAE): Low value indicates high quality stego image is produced.

$$NAE = \frac{\sum\sum abs(f(u.v) - f'(u,v))}{\sum\sum abs(f(u,v))}$$

Table 7: Result of NAE after Applying on Cover and Stego image

Serial NO	Size of Secret Message	NAE(Red channel)	NAE (Green channel)	NAE (Blue channel)
1.	1KB (1024 bytes)	0.000015	0.000018	0.000018
2.	2KB (2048 bytes)	0.000030	0.000036	0.000036
3.	3KB (3072 bytes)	0.000044	0.000054	0.000054
4.	4KB (4094 bytes)	0.000060	0.000071	0.000072

As we can see the table shows low NAE after embedding the secret message using the proposed algorithm.

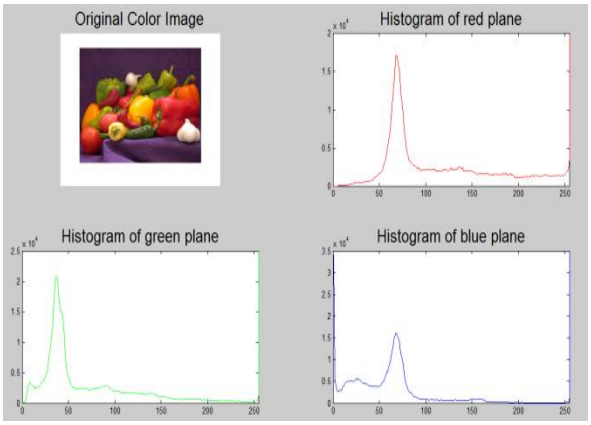

Figure 11: Histogram of Cover image

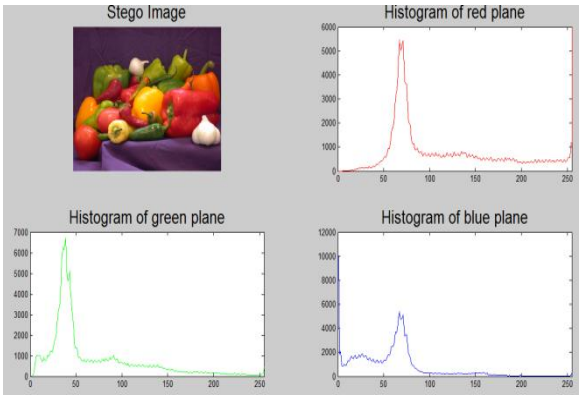

Figure 12: Histogram of Stego image with 4kb hidden data

6. Conclusion and Future work

Proposed method doesn't directly stores the secret message still the result shows it gives better PSNR ratio implies low error rate, low SC, low AD and low NAE .The error rate is minimized by changing 8^{th} LSB according to the secret message and cover image value which gives maximum of $\pm 2^{i-1}$ and minimum of 0 as error. From Table-6 we can conclude that our proposed algorithm give fixed MD which is very low. And because of randomization the histograms analysis also gives better result. In future, we will apply this algorithm in frequency domain so that the secret data can be spread over the entire image.

References

[1] Arvind Kumar, Km. Pooja, "Steganography- A Data Hiding Technique" International Journal of Computer Applications, Vol. 9– No.7, pp, 19-23, November 2010.

[2] Li B., He, J. Huang, J.She, Y.Q,"A Survey on Image Steganography and Steganalysis" Journal of Information hiding and Multimedia Signal Processing 2(2), pp,142–172 2011.

[3] Jarno Mielikaine, "LSB Matching Revisited", IEEE Signal Processing Letters, Vol. 13, No.5,pp,285-287, 2006.

[4] Juan José Roque, Jesús María Minguet, "SLSB: Improving the Steganographic Algorithm LSB", WOSIS, INSTICC press, pp, 57-66, 2009.

[5] Amanpreet Kaur, Renu Dhir, and Geeta Sikka , "A New Image Steganography Based On First Component Alteration Technique", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 6, No. 3,pp,53-56 ,2009.

[6] Gandharba Swain1 and Saroj Kumar Lenka2, "LSB Array Based Image Steganography Technique by Exploring the Four Least Significant Bits", Springer-Verlag Berlin Heidelberg, Part II, CCIS 270, pp, 479–488, 2012.

[7] Ibrahim, Rosziati, and Teoh Suk Kuan. "Steganography algorithm to hide secret message inside an image." arXiv preprint arXiv:1112.2809 (2011).

[8] M. Al-Shatnawi, "A New Method in Image Steganography with Improved Image Quality" Applied Mathematical Sciences, Vol. 6, no. 79,pp, 3907 – 3915, 2012.

Tanusree Podder is pursuing M.Tech in Computer Science and Engineering at National Institute of Technology, Agartala, Tripura, INDIA. She is presently working on cryptography and Steganography.

Lalita Kumari is working as Assistant Professor in Department of Computer Science & Engineering, NIT Agartala, India. Her area of interest is Image Processing, Pattern recognition, etc.

Abhishek Majumder is pursuing M.Tech in Computer Science & Engineering at SSCET, Badhani, Punjab, India under Punjab Technical University, Jalandhar. His ongoing research area is on DNA based Cryptography and Steganography and his areas of interest are network security, image processing, and bioinformatics.