International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.4 (2014), pp.363-374
http://dx.doi.org/10.14257/ijmue.2014.9.4.37

High Capacity Signature Hiding Technique in Higher Depth of LSB Layer

Biswajita Datta[1], Debnath Bhattacharyya[2] and Samir Kumar Bandyopadhay[3]

[1]*Department of Computer Science and Engineering, St. Thomas College of Engineering, Kolkata, India*
[2]*Department of Computer Science and Engineering, Vignan University, Guntur-522213, India*
[3]*Department of Computer Science and Engineering, University of Calcutta, Kolkata-700009, India*

debnathb@gmail.com, skb1@vsnl.com

Abstract

Today Signature is very popular authentication information. Here we try to hide this confidential information by Steganography. This security technique prevents discovery of the very existence of communication through digital media. In our proposed work the LSB replacement technique of Steganography is used. Here we able to increase imperceptibility as well as capacity of stego image by considering higher LSB layer for hiding the target data and replacing multiple bits.

Keywords: *Steganography, Signature, Cover image, Target data, Stego image, Human visual system*

1. Introduction

Communication is a very important word with respect to civilisation. Nations who cannot communicate with other basically lost their identity. With the advancement of technology today communication are mostly done through internet which is open and public in nature. So information protection has become most vital issue and ongoing topic of research. During the transformation if information are intentionally or unintentionally modified it lost its meaning. The information may be protected against these adversaries if we can hide the existence of the message [1].

Steganography can be an effective for this secure communication through these digital media. Its aim is to hide the very existence of the message in the cover medium [2]. The term steganography was only coined at the end of the 15th century but data hidden on the back of wax writing tables, written on the stomachs of rabbits, or tattooed on the scalp of slaves are the early use of it [3]. Recently at the time of World War II people also use invisible writing by the help of invisible ink. Modern steganography is generally understood to deal with electronic media rather than physical objects. Steganography is the combination of two Greek words "stegos" (means "cover") and "grafia" (means) "writing" defining it as "covered writing" [4, 10]. The basic requirements of Steganography are:

ISSN: 1975-0080 IJMUE

- Cover media that can hold the hidden information.
- The secret message (target message) that basically embed within the cover media.
- A steganography function and its inverse.

Based on the different media we have text, image, audio, video and protocol Steganography where the cover medium are text, image, audio, video, and IP packet respectively [5]. Both the spatial and transform domain embedding scheme as well as for each of the cover media has to maintain the challenges of Steganography which are imperceptibility, robustness and capacity. Imperceptibility is concerned with the fact that human being should not become suspicious of the existence of the covert data within the medium. Robustness ensures that once a secret message is inserted, it becomes impossible to delete or manipulate that message. Moreover, the capacity of the digital media deals with the fact that how much target we can embed within the cover file without less distortion of it [6]. But these factors are depend on each other and a balance must be maintained between them as increase in one of the factors leads to the decrease in other.

In our proposed method we try to we concentrate on how much target data we can embed into the cover file so that human sense cannot follow its existence. Here a binary image is hidden within another which is basically a colour image. And we also try to increase the robustness of stego image by increasing the depth of LSB layer. Basically image Steganography technique try to hide the existence of the so that it can cheat the HVS [7, 8].

2. Proposed method

As target data we consider a very important biometric authentication data – signature. In our proposed technique we consider that the signature image is basically a binary image and we try to hide this binary image within a 24 bit Color image. In case of a binary image each pixel is 1 bit long and can have values as either 0 or 1, whereas in case of a color image, each pixel is 24 bit long. These 24 bits can be thought of as a collection of 3 bytes where the first byte (first 8 bits) signify the intensity of red component, the next byte signifies green component and the last byte signify blue component.

In our method first we have replaced the 5th bit (from the LSB) of each of the R, G and B component of the cover image with the pixel value of the target image, which is either 0 or 1. if I want to store 1 in a binary string S='11101011'(235), then S(5)=0 is replaced by 1 and if I want to store 0 in a binary string S='10111111' (191), then S(5)=1 is replaced by 0. Thus the modified string becomes S='11111011' (251) and S= '10101111' (175) respectively. In both cases the difference between original and modified pixel value is 16. So replacing the 5th bit may give rise to a problem. From the example stated above it is observed that a changing only in the 5th bit can change the entire value of the string up to a maximum of 16. In our proposed technique we try to adjust the bits of the original string after embedding the data in the 5th LSB layer to reduce this difference [9].

Technique for bit adjustment to minimize the change in pixel value due to replacement of 5th LSB layer we consider 3 possible cases.

Case 1: 5th bit changes from 1 to 0

Again it has 4 sub cases:

Table 1(a)

Sub-case no.	Specifications	Action taken
1.1	when the 4th and 6th bit (from LSB) are 0 and 0 respectively	Set all the bits to the right of the 5th bit (i.e., towards LSB) to 1.
1.2	when the 4th and 6th bit (from LSB) are 0 and 1 respectively	Set all the bits to the right of the 5th bit (i.e., towards LSB) to 1
1.3	when the 4th and 6th bit (from LSB) are 1 and 0 respectively	Set all the bits to the right of the 5th bit (i.e., towards LSB) to 0 and set the 6th bit (first bit to the left of the 5th bit) to 1.
1.4	When the 4th and 6th bit are 1 and 1 respectively.	Set all the bits to the right of the 5th bit (i.e., towards LSB) to 0 and set all the bits to the left of the 5th bit (i.e., towards the MSB) to 0 until a 0 is encountered. When a 0 is encountered, set it to 1.

Case 2: 3rd bit changes from 0 to 1.

Again it also has 4 sub cases

Table 1(b)

Sub-case no.	Specifications	Action taken
2.1	when the 4th and 6th bit are 1 and 1 (from LSB) respectively	Set all the bits to the right of the 5th bit (i.e towards LSB) to 0.
2.2	when the 4th and 6th bit are 1 and 0 (from LSB) respectively	Set all the bits to the right of the 5th bit (i.e towards LSB) to 0
2.3	when the 4th and 6th bit are 0 and 1 (from LSB) respectively	Set all the bits to the right of the 5th bit (i.e towards LSB) to 1 and set the 6th bit(first bit to the left of the 5th bit) to 0
2.4	when the 4th and 6th bit are 0 and 0 (from LSB) respectively	Set all the bits to the right of the 5th bit (i.e towards LSB) to 1 and set all the bits to the left of the 5th bit (i.e towards the MSB) to 1 until a 1 is encountered. When a 1 is encountered, set it to 0.

Case 3: No change at all. If the bit which I want to place at the 3rd LSB position of the pixel value is same with the 3rd LSB of the original pixel value then the original pixel value become unchanged. Here I also may think of two sub-cases: 0 and replace with 0 and 1 and replace with 1.

Now we demonstrate the first two cases and their sub cases of our proposed method with some examples.

Case 1: 5th bit changes from 1 to 0

Case 1.1:

when the 4th and 6th bit (from LSB) are 0 and 0 respectively		8	7	6	5	4	3	2	1
Original intensity value	146	1	0	0	1	0	0	1	0
After Replacement of 5th bit with 0	130	1	0	0	0	0	0	1	0
Modified intensity value with shaded bits	143	1	0	0	0	1	1	1	1

Case 1.2:

when the 4th and 6th bit (from LSB) are 0 and 1 respectively		8	7	6	5	4	3	2	1
Original intensity value	178	1	0	1	1	0	0	1	0
After Replacement of 5th bit with 0	162	1	0	1	0	0	0	1	0
Modified intensity value with shaded bits	175	1	0	1	0	1	1	1	1

Case 1.3:

when the 4th and 6th bit (from LSB) are 1 and 0 respectively		8	7	6	5	4	3	2	1
Original intensity value	154	1	0	0	1	1	0	1	0
After Replacement of 5th bit with 0	138	1	0	0	0	1	0	1	0
Modified intensity value with shaded bits	160	1	0	1	0	0	0	0	0

Case 1.4:

When the 4th and 6th bit are 1 and 1 respectively		8	7	6	5	4	3	2	1
Original intensity value	186	1	0	1	1	1	0	1	0
After Replacement of 5th bit with 0	170	1	0	1	0	1	0	1	0
Modified intensity value with shaded bits	192	1	1	0	0	0	0	0	0

Case 2: 5th bit changes from 0 to 1

Case 2.1:

when the 4th and 6th bit are 1 and 1 (from LSB) respectively		8	7	6	5	4	3	2	1
Original intensity value	170	1	0	1	0	1	0	1	0
After Replacement of 5th bit with 0	186	1	0	1	1	1	0	1	0
Modified intensity value with shaded bits	176	1	0	1	1	0	0	0	0

Case 2.2:

when the 4th and 6th bit are 1 and 0 (from LSB) respectively		8	7	6	5	4	3	2	1
Original intensity value	138	1	0	0	0	1	0	1	0
After Replacement of 5th bit with 0	154	1	0	0	1	1	0	1	0
Modified intensity value with shaded bits	144	1	0	0	1	0	0	0	0

Case 2.3:

when the 4th and 6th bit are 0 and 1 (from LSB) respectively		8	7	6	5	4	3	2	1
Original intensity value	162	1	0	1	0	0	0	1	0
After Replacement of 5th bit with 0	178	1	0	1	1	0	0	1	0
Modified intensity value with shaded bits	159	1	0	0	1	1	1	1	1

Case 2.4:

when the 4th and 6th bit are 0 and 0 (from LSB) respectively		8	7	6	5	4	3	2	1
Original intensity value	130	1	0	0	0	0	0	1	0
After Replacement of 5th bit with 0	146	1	0	0	1	0	0	1	0
Modified intensity value with shaded bits	127	0	1	1	1	1	1	1	1

We can see from the above example that the maximum change in the modified pixel is 8 instead of 16. After the adjustment of the bits due to enhancement of perceptual transparency we replace LSB (1st bit) for increasing the capacity of the stego file and for this case the max change is 9.

In case of binary image each pixel can have only 2 values, either 0 or 1. Thus when I apply this algorithm to hide a binary image within a 24 bit RGB image, I can embed 2 pixels of the target image in each of the R, G and B component of the cover image- one in the 5th LSB and another in the 1st LSB position. Thus a total of 6 pixels of the target image can be stored in one pixel of the cover image.

Since 6 pixel of the target image can be saved in 1 pixel of the cover image, instead of embedding the target pixels in consecutive pixels of the cover image from the 2nd row. For hiding the data we do not consider the consecutive pixel we choose the pixels one after the other.

The first row is used for embedding image size. We store this value based on digits of row and column value. Now we explain this by an example let the binary image size be 87×123. So row value is 87 and column value is 123. The number of digits in row value is 2 in binary it is 0010 and stores it in 1st and 2nd LSB of R and G plane of 1st pixel. The corresponding number of digits in column size is stored in 1st and 2nd LSB of R and G plane of 2nd pixel.. Then the individual digits (like 8,7,1,2,3) of row and column size are stored in consecutive 1st and 2nd LSB of R and G plane pixel of first row starting from 5th pixel, row and then column basis.

At the receiver side we first collect the size of the target image and then try to form the target image by extracting data from it. Here we take a pixel of stego image and pick the 5th and 1st LSB of each of the three planes R, G and B. These six bits form six pixels of target signature image. Like this the other pixels of target image are formed.

3. Algorithm of Proposed Technique

Encoding Algorithm

Step 1: Start
Step 2: Read cover image and target image.
Step 3: Store the size of the target image in t_row(row size) and t_col(column size) variable.
Step 4: To store the length of the target image call function st_len(cover image, t_row, t_col)
Step 5: [start hiding from second row of the cover image]
Repeat Step 5 to Step 8 until all pixel of target image is not embedded
Take a pixel of target image and embed it at the 5^{th} bit of R plane of a pixel of cover image after that call the algorithm adjust(cover image). [Consider LSB as index 1 and MSB as index 8]
Step 6: After adjusting the intensity value insert next pixel of target image into the LSB of adjusted pixel
Cover image.
Step 7: Next two pixel of target image are inserted into the G plane of same pixel of cover image by
considering same procedure.
Step 8: Then next two pixel of target image are inserted into the B plane of same pixel of cover image by
considering same procedure.

[So we see that six pixel of target image are inserted into a single pixel of cover image.]
Step 9: Send the stego image to the receiver.
Step 10: End.

Algorithm for length storing

Function st_len(cover image, t_row, t_col)
[t_row and t_col be the row and column size of target image,respectively.]

Step 1: Count the number of digit in t_row and t_col value.
Step 2: Convert these two numbers into 4 bit binary and store these in r_bin and c_bin.
Step 3: Replace 1st and 2nd LSB of Red and Green component of first pixel of first row of the cover image
with 3rd - 4th and 1st - 2nd bits of r_bin.
Step 4: Replace 1st and 2nd LSB of Red and Green component of first pixel of first row of the cover image
with 3rd - 4th and 1st - 2nd bits of c_bin.
Step 5: Cut the digit of t_row value.
Step 6: Convert each digit into 4 bit binary.
Step 7: Replace 1st and 2nd LSB of Red and Green component of each pixel of first row of the cover image
with 3rd - 4th and 1st - 2nd bits from LSB of each 4 bit binary.
Step 8: Restore the bits of Red, Green component of modified pixel in the cover image.
Step 9: Apply Step 5, 6, 7 and 8 for column size t_col also.
Step 10: end

Algorithm for bit adjustment

Function adjust (cover image, t_row, t_col)
[t_row and t_col be the row and column size of target image,respectively.]

Step 1: Start
Step 2: Convert the R, G, B values of the selected pixel of the cover image to its corresponding binary value.
Step 3: For each of the R, G, B components of the selected pixel (P) of the cover image repeat the following steps
Step 4: Replace the 5th bit (from the LSB) with the pixel value of the Target image (say S(i), where i =1,2,…….S).

Step 5: if the change in the 5th bit is from 0 to 1 follow the following steps

Step 5.1: let the a_i = 5th bit, if a_{i-1} =1 and a_{i+1} =1 then set $a_{i-1}, a_{i-2}, \ldots a_0 = 0$

Step 5.2: else if a_{i-1} =1 and a_{i+1} =0 then set $a_{i-1}, a_{i-2}, \ldots a_0 = 0$

Step 5.3: else if a_{i-1} =0 then set $a_{i-1}, a_{i-2}, \ldots a_0 = 1$

Step 5.3.1: if a_{i+1} =1 then set a_{i+1} = 0

Step 5.4: else j= i+1, while (a_j! = 1) set a_j = 1
Step 5.4.1: increment j
Step 5.4.2: set a_j = 0

Step 6: if the change in the 5th bit is from 1 to 0 follow the following step

Step 6.1: let the a_i = 5th bit, if a_{i-1} =0 and a_{i+1} =0 then set
$a_{i-1}, a_{i-2}, \ldots a_0 = 1$

Step 6.2: else if a_{i-1} =0 and a_{i+1} =1 then set
$a_{i-1}, a_{i-2}, \ldots a_0 = 1$

Step 6.3: else if a_{i-1} =1 then set $a_{i-1}, a_{i-2}, \ldots a_0 = 0$

Step 6.3.1: if a_{i+1} =0 then set a_{i+1} = 1

Step 6.4: else j= i+1, while (a_j! = 0) set a_j = 0
Step 6.4.1: increment j
Step 6.4.2: set a_j = 1

Step 7: Set a(0), i.e 1st bit from the LSB = S(i+1), i.e the next pixel of target image.
Step 8: End

Decoding Algorithm

Step 1: Take the stego image as input at the receiver side.
Step 2: The row and column size first extracted from first row of the stego image.
Step 3: Iterate the decoding algorithm according to the size of the target image.
Step 3.1: Take a pixel of cover image.
Step 3.2: Pick 5th and 1st LSB of R plane of target image.
Step 3.3: Pick 5th and 1st LSB of G plane of target image.
Step 3.4: Pick 5th and 1st LSB of B plane of target image.
Step 4: Restore them for forming the target image.
Step 5: End

4. Result

Signature is very secure as well as important information for authentication. We should transmit this instruction secretly and for hide its existence during transmission here in our proposed technique we embed signature within a RGB cover image. After applying our proposed method we can hide this and the results are shown Figure 1 and Figure 2.

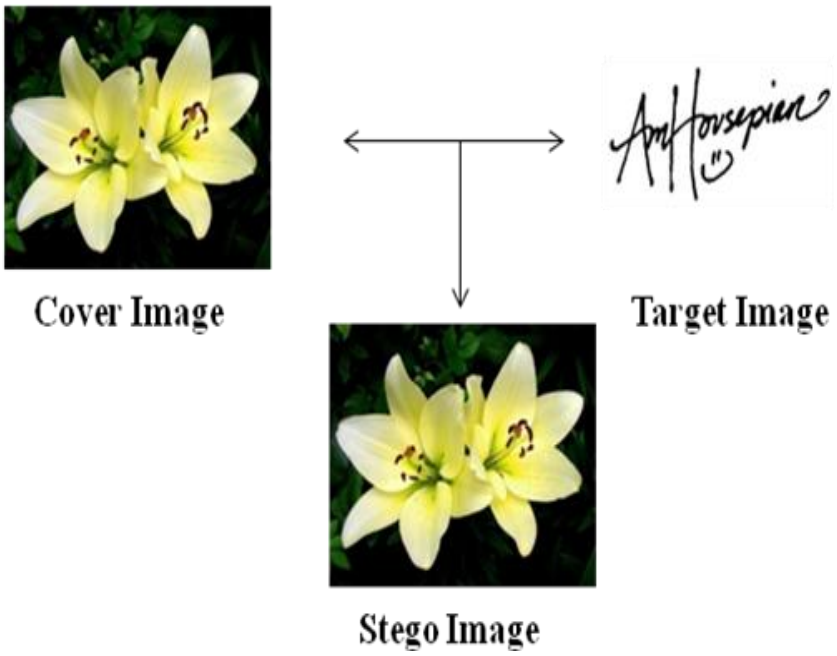

Figure 1. Test Case 1

Figure 2. Test Case 2

For analyzing the result we consider the test case 1.

Target image: (binary image)

Cover image: (RGB color image)

Suppose the size of the target image be 100 × 72. We store the size in the first row according to the technique discussed earlier. In the following table we show how the target pixel intensity value is stored within the cover image by an example in Figure 3 and Table 2a.

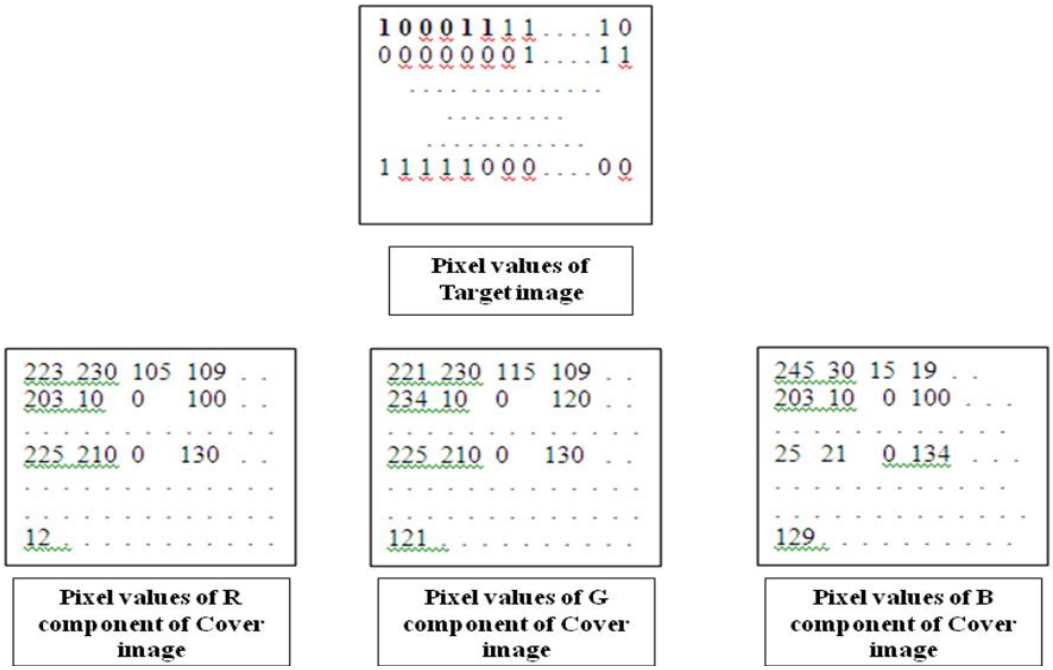

Figure 3.

Table 2(a)

Target pixel (insert at 5th)	Original intensity value (Dec)	Original Intensity value (Bin)	Sub-case	Modified intensity value (Bin)	Modified Intensity value (Dec)	Next Target pixel (insert at 1st)	Modified intensity value (Bin)	Modified Intensity value (Dec)
1st pixel of 2nd row								
Red Component								
0	177	101**1**0001	1.2	101**0**1111	175	0	101**0**111**0**	174
Green Component								
1	255	111**1**1111	3	111**1**1111	255	0	111**1**111**0**	254
Blue Component								
1	10	000**0**1010	2.2	000**1**0000	16	1	000**1**000**1**	17
2nd pixel of 2nd row								
Red Component								
0	186	101**1**1010	1.4	110**0**0000	192	1	110**0**000**1**	193
Green Component								
1	110	011**0**1110	2.1	011**1**0000	112	0	011**1**000**0**	112

In case of decoding, we read the 5th and 1st bit (from LSB) of the selected pixels of the stego image. We arrange these pixels in a 2-D matrix having size equal to that of the size of the target image. The process is explained in the Table 2B.

Table 2(b)

Binary value of selected pixel	Value of 5th bit (from LSB)	Value of 1st bit from LSB
101**0**111**0**	0	0
111**1**111**0**	1	0
000**1**000**1**	1	1
110**0**000**1**	0	1
011**1**000**0**	1	0

At receiver side the target image be:

0 0 1 0 1 1 0 1 1 0 ……
……..
……
…..
……
…….
……

Here in this method we consider 5th LSB for inserting data one target data. By increasing the depth of LSB layer we try to remove the main disadvantage of standard LSB coding techniques which is low robustness. But when we increase the depth of LSB layer during LSB coding the probability of making the embedded message statistically detectable increases and perceptual transparency of stego objects is decreased. In our proposed method we try to minimize this limitation by using an adjustment technique. Without applying this adjustment technique if we embed a target data at 5th LSB position the modified pixel's intensity value become 16 more than the original (24). But in our proposed technique it reduces to at most 8.

Then we embed next target data at LSB for increasing the capacity of the stego image and for this the difference between original and modified pixel's intensity value becomes at most 9.

Since 6 pixels of target image is hidden in one pixel of RGB colour image so I can hide a binary image of size 5 times more than the cover image into that particular cover image. By this we can meet one of the challenges of Steganography.

The main title (on the first page) should begin 1 3/16 inches (7 picas) from the top edge of the page, centered, and in Times New Roman 14-point, boldface type. Capitalize the first letter of nouns, pronouns, verbs, adjectives, and adverbs; do not capitalize articles, coordinate conjunctions, or prepositions (unless the title begins with such a word). Please initially capitalize only the first word in other titles, including section titles and first, second, and third-order headings (for example, "Titles and headings" — as in these guidelines). Leave two blank lines after the title.

5. Conclusion

In this modern era where security becomes an important issue, Steganography plays a vital role for secure communication. Authentication, data integrity as well as confidentiality issues are maintained in our work because we hide the signature of a person within an image in such a way so that no one can understand its existence. In our proposed method we meet the three challenges of Steganography mainly capacity by hiding a binary image within an RGB image.

In our future work we try to increase the capacity of stego image by compressing signature image.

References

[1] M. Kharrazi, H. T. Sencar and N. Memon, "Image Steganography: Concepts and Practice", Lecture Notes Series", Institute for Mathematical Sciences, National University of Singapore, Singapore, **(2004)**.
[2] B. Pfitzmann, "Information Hiding Terminology", Proc. of First Int. Workshop on Information Hiding, Cambridge, UK, **(1996)** May 30-June1, Lecture notes in Computer Science, vol. 1174, Ross Anderson (Ed.), pp. 347-350.
[3] J. Fridich and R. Du, "Secure Steganographics Methods for Palette Images", In Information Hiding, 3rd International Workshop, Springer, **(1999)**, pp. 47-60.
[4] N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen", Computer, vol. 31, no. 2, **(1998)**, pp. 26–34.
[5] R. B. Wolfgang and E. J. Delp, "Watermark for digital images", Proceeding of the IEEE International Conference on Image Processing, IEEE Computer Society, Washington DC, USA, **(1996)** September 16-19, pp. 219-222.
[6] C. C. Chang, T. D. Kieu and Y. C. Chou, "High capacity data hiding for grayscale images", In Proceedings of the First International Conference on Ubiquitous Information Management and Communication, Seoul, Korea, **(2007)** February, pp. 139–148.
[7] C. Parthasarathy and S. K. Srivatsa, "Increased Robustness of LSB Audio Steganography by Reduced Distortion LSB Coding", Journal of Theoretical and Applied Information Technology, vol. 7, **(2005 – 2009)**, pp. 080 - 086.
[8] S. K. Bandyopadhay, B. Datta and K. Dutta, "Information Hiding in Higher LSB Layer in an Audio Image", International Journal of Advanced Research in Computer Science, vol. 2, no. 3, **(2011)**.
[9] S. K. Bandyopadhay and B. Datta, "Higher LSB Layer Based Audio Steganography Technique", International Journal on Electronics & Communication Technology, vol. 2, Issue 4, **(2011)** October - December, pp. 129-135.
[10] G. Paul, I. Davidson, I. Mukherjee and S. S. Ravi, "Keyless Steganography in Spatial Domain using Energetic Pixels", In Proceedings of the 8th International Conference on Information Systems Security (ICISS), **(2012)** December 15-19, Guwahati, India, vol. 7671, LNCS, Springer, pp. 134-148.

