International Journal of Engineering Inventions
ISSN: 2278-7461, ISBN: 2319-6491
Volume 2, Issue 1 (January 2013) PP: 37-45

Different Steganography Methods and Performance Analysis

Shantala .C.P[1], K.V Viswanatha[2]
[1]Research Scholar, Dr MGR Educational and research Institute. Chennai, India,
[2]Professor, Dept of Computer Science, CIT, Gubbi, Tumkur, Karnataka, India

Abstract:- Steganography is a process that involves hiding a message in an appropriate carrier for example an image, audio or any data file. The carrier can then be sent to a receiver without anyone else knowing that it contains a hidden message. The objective is to hide the existence of the message in the carrier and make the method easy to read the message. The aim of this study is to investigate steganography and how it is implemented. Based on this work a number of methods of steganography are implemented and evaluated. The strengths and weaknesses of the chosen methods have been analyzed. Different steganography methods are implemented. The methods are chosen for their different strengths in terms of resistance to different types of steganalysis or their ability to maximize the size of the message they could store. All of the methods used are based on the manipulation of the single least significant bit of pixel values, which correspond to the message being hidden.

Keywords:- File folder steganography, Image Processing, image steganography, Steganography, TCP/IP-Covert channel.

I. INTRODUCTION

The word steganography means "covered or hidden writing" [1]. The object of steganography is to send a message through some innocuous carrier to a receiver while preventing anyone else from knowing that a message is being sent at all. Computer based stenogaraphy allows changes to be made to what are known as digital carriers such as images, sounds or any data files. If the process is successful, the changes represent the hidden message, resulting in no discernible change to the carrier.

Cryptography and steganography are different. Cryptographic techniques can be used to scramble a message so that it cannot be read, even if it is accessed. If a cryptographic message is discovered it is generally known to be a piece of hidden information (anyone intercepting it will be suspicious) but it is scrambled so that it is difficult or impossible to understand and de-code. Steganography hides the very existence of a message so that, if successful, it generally attracts no suspicion at all. Using steganography, information can be hidden in carriers such as images, audio files, text files, videos and data transmissions [1]. When the message is hidden in the carrier a stego-carrier is formed for example a stego-image.

Hopefully it will be perceived to be as close as possible to the original carrier or cover image by the human senses. Images are the most widespread carrier medium [2].

They are used for steganography in the following way. The message may firstly be encrypted. The sender (or embedder [4]) embeds the secret message to be sent into a graphic file [3] (the cover image [4] or the carrier). This results in the production of what is called a stego-image. Additional secret data may be needed in the hiding process e.g. a stegokey. The stego-image is then transmitted to the recipient [3]. The recipient (or extractor [4]) extracts the message from the carrier image. The message can only be extracted if there is a shared secret between the sender and the recipient. This could be the algorithm for extraction or a special parameter such as a key [3] (the stegokey). A stego analyst or attacker may try to intercept the stego-image. Figure 1 shows the steganographic system.

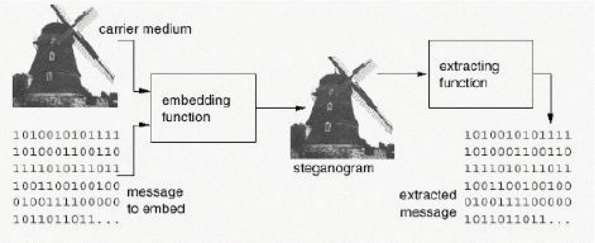

Figure 1: The Steganographic System [2].

1.1 Types Of Digital Carriers

There are a variety of digital carriers or places where data can be hidden. Data may be embedded in different types of images (lossless and lossy compressed images). Properties of images can be manipulated including luminescence, contrast, colors [1], edges of images at different tolerance.

In audio files small echoes or slight delays can be included or subtle signals can be masked with sounds of higher amplitude by converting audio in to 16 bit binary file and changing the LSB values with data to be hidden.

Unused or reserved space on a disc can be used to hide information. Information can be hidden in reserved field of file header attributes which is of 10 bytes per file. The disk directory is 7 sectors long and contains all of the information about a file except for the information stored in the FAT. Each entry in the directory is 32 bytes long and has 8 fields. Here the 10 bytes of reserved field can be used to hide the data.

Data may be hidden in unused space in file headers. Information can be hidden in TCP/IP header. Within each header there are multitudes of areas that are not used for normal transmission. An analysis of the areas of a typical IP header that are either unused or optional reveals many possibilities where data can be stored and transmitted, because these fields are not more likely to be altered in transit (TCP/IP optional fields). Therefore the following may be encoded and decoded.

-The IP packet identification fields
-The TCP initial sequence number fields
-The TCP acknowledge sequence number fields

1.2 Image Structure And Image Processing

Each pixel is generally stored as 24-bit or 8-bit. A 24-bit pixel has a possibility of 224 color combinations [1]. The 24 bits of a 24-bit image are spread over three bytes and each byte represents red, green and blue respectively. Colors are obtained by mixing red, green and blue in different proportions. An image can be formed by making three measurements of brightness at each pixel using the red, green and blue components of the detected light. Using the RGB model the value of f(x, y) is a vector with three components corresponding to red (R), green (G) and blue (B). They can be regarded as orthogonal axes defining a three dimensional color space. Every value of f(x, y) is a point in the color cube shown in Figure 2 [5]. The three components are normally quantized using 8 bits. An image made of these components is described as a 24-bit color image [Efford00]. Each byte can have a value from 0 to 255 representing the intensity of the color. The darkest color value is 0 and the brightest is 255. For example a pixel could be made up of three bytes as follows: 11111111 00000000 00000000. The first 8 bits represent red, the second 8 bits represent green and the third 8 bits represent blue. The bit values in this example result in a red pixel. Its red byte is at a maximum value (11111111) and its green (00000000) and blue (00000000) bytes have the lowest possible value.

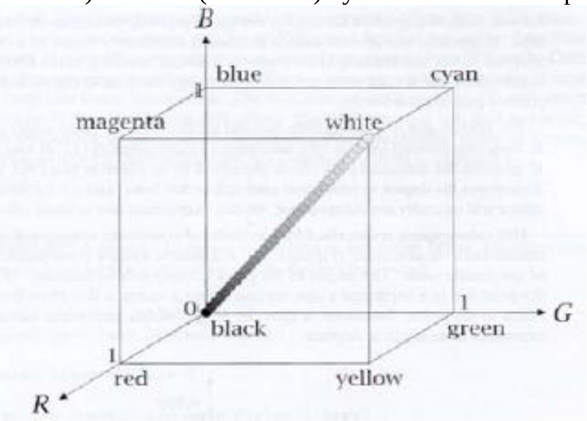

Figure 2: The RGB Color Cube (Efford, 2000)

Some images are 8-bit. Each pixel is represented by one byte only. This one byte can have any value ranging from 0 to 255 generating 256 possible colors or 256 grayscale values for black and white images. The colors are taken from a color index or palette, also called a color map or color table. This palette contains up to 256 colors representing the colors in the image. The value of the pixel in an image points to a color in the palette [1].

1.3 Image Edges and Its Tolerance

Edges are places in the image with strong intensity contrast. Since edges consist of mainly high frequencies, we can, in theory, detect edges by applying a high pass frequency filter in the Fourier domain or by

convolving the image with an appropriate kernel in the spatial domain. In practice, edge detection is performed in the spatial domain, because it is computationally less expensive and often yields better results.
We can see that the position of the edge can be estimated with the maximum of the 1st derivative or with the zero-crossing of the 2nd derivative. Therefore we want to find a technique to calculate the derivative of a two-dimensional image. For a discrete one-dimensional function f(i), the first derivative can be approximated by equation as shown in fig 3.

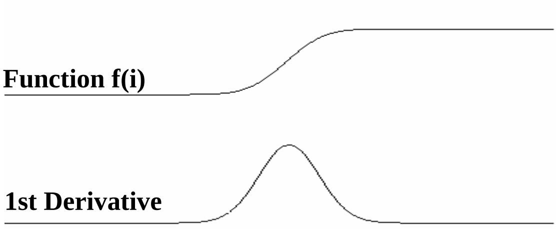

Figure 3: 1st order derivative of an edge illustrated in one dimension.

After having calculated the magnitude of the 1st derivative, we now have to identify those pixels corresponding to an edge. The easiest way is to threshold the gradient image, assuming that all pixels having a local gradient above the threshold must represent an edge. An alternative technique is to look for local maxima in the gradient image, thus producing one pixel wide edges. A more sophisticated technique is used by the Canny edge detector. It first applies a gradient edge detector to the image and then finds the edge pixels using non-maximal suppression and hysteresis tracking.

1.4 Tolerance

Digital images are typically stored in either 24-bit or 8-bit per pixel files. 24-bit images are sometimes known as true color images. Obviously, a 24-bit image provides more space for hiding information. However 24-bit images are generally large and not that common. 24-bit image, 1024 pixels wide by 768 pixels high would have a size in excess of 2 Mbytes. As a result this Tolerance Algorithm provides a combination of 224 combinations of values to the Tolerance which is nearly impossible to decode the given tolerance.

1.5 Audio Formats

It is important to distinguish between a file format and a codec. A codec performs the encoding and decoding of the raw audio data while the data itself is stored in a file with a specific audio file format. Though most audio file formats support only one audio codec, a file format may support multiple codecs, as AVI does.
The following are three major groups of audio file formats.

- Uncompressed audio formats, such as WAV, AIFF and AU.
- Formats with lossless compression, such as FLAC, Monkey’s Audio (filename extension APE), WavPack, Shorten, TTA, Apple Lossless and lossless Windows Media Audio (WMA).
- Formats with lossy compression, such as MP3, Vorbis, lossy Windows Media Audio (WMA) and AAC.

II. STEGANALYSIS

The two stages involved in breaking a steganographic system are detecting that steganography has been used and reading the embedded message [3]. Steganalysis methods should be used by the steganographer in order to determine whether a message is secure and consequently whether a steganographic process has been successful. The goal of a stegoanalyst is to detect stego-messages, read the embedded message and prove that the message has been embedded to third parties [4]. Detection involves observing relationships between combinations of cover, message, stego-media and steganography tools [1]. This can be achieved by passive observation. Active interference by the stegoanalyst involves removing the message without changing the stego-image too much (the stegoanalyst might want to conceal his existence), or removing the message without consideration to the stego-image appearance or structure [4]. Whether a message has been hidden in an image or not, the image could be manipulated to destroy a possible hidden message [1].

There are two necessary conditions to be fulfilled for a secure steganographic process. The key must remain unknown to the attacker and the attacker should not be familiar with the cover image [3]. If the cover image is known, the message could then be embedded in a random way so that it is secure. However it is preferable that the image is unknown. Attacks on steganography can involve detection and/or destruction of the embedded message. A stego-only attack is when only the stego-image is available to be analyzed [1]. A known cover attack is when the original cover image is also available. It involves comparing the original cover image with the stego-image. As explained above hiding information results in alterations to the properties of a carrier which may result in some sort of degradation to the carrier [1]. Original images and stego-images can be analyzed by looking at color composition, luminance and pixel relationships and unusual characteristics that can be detected. If a hidden message is revealed at some later date, the attacker could analyze the stego-image for future attacks. This is called 'known message attack'. The chosen stego attack is used when the steganography algorithm and the image are known. A chosen message attack is when the stegoanalyst generates stego-images using a given steganography algorithm using a known message [1]. The purpose is to examine the patterns produced in the stego-images that may point to the use of certain steganography algorithms. Most steganographic algorithms embed messages by replacing carefully selected pixels bits with message bits [2]. Any changes to the data associated with the image through embedding will change the properties of the image in some way. This process may create patterns or unusual exaggerated noise [1].

The patterns visible to the human eye could broadcast the existence of a message and point to signatures of certain methods or tools used [1]. Human sight is trained to recognize known things. This process of analysis depends on the ability of humans to discern between normal noise and visual corruption and patterns created by steganography [2]. It can be difficult to distinguish randomness and image contents and also distinguish LSBs and random bits by machine.

If numerous comparisons can be made between the cover images and the stego-images, patterns can begin to emerge [1]. At a later stage if the cover is not available the known signature will be sufficient to indicate a message and the tool used to embed it [1]. Some of the methods of carrying out steganography produce characteristics that act as signatures for that steganography method [1]. The image may not give away the existence of stenography but the palette could. Therefore steganography can be detected by examining the palette itself. In color palettes the colors are ordered from most used to least used. The changes between color values rarely change in one-bit increments in an unstegoed image. But this feature would be created by embedding in the LSBs during steganography.

In order to prevent detection, steganographic and cryptographic keys can be used. A steganographic key controls embedding and extracting of the message. The key could scatter the message randomly over the carrier. A cryptographic key is used to encrypt a message before embedding. Therefore even when the message is detected it can't be read. In the case of bitwise methods destruction of the embedded message is fairly easy because the LSBs of the images can be changed with compression. The image may be converted to lossy compression format such as JPEG. JPEG images which have been processed with Jpeg-Jsteg can be recompressed and this will destroy the message embedded in the DCT coefficients because they will be recalculated [1].

III. STEGANOGRAPHIC METHODS AND ANALYSIS

The aim of the study is to produce a system, containing different stenographic methods and to examine the strengths and weakness of those methods. The system to be produced will also contain an option to encrypt the message before it is embedded (image steganography).

Steganography can be carried out on any digital media. The chosen media for different steganographic methods are Images (BMP/JPEG, Edges of an image and Tolerance).

IV. EVALUATION AND RESULTS

When images are used as the carrier in steganography they are generally manipulated by changing one or more of the bits of the byte or bytes that make up the pixels of an image. The message can be stored in the LSB of one color of the RGB value or in the parity bit of the entire RGB value. Changing the LSB will only change the integer value of the byte by one. Usually three bits from each pixel can be stored to hide an image in the LSBs of each byte of a 24 bit image. Consequently, LSB requires that only half of the bits in an image be changed when data can be hidden in least and second least significant bits and yet the resulting stego image which will be displayed is indistinguishable to the cover image to the human visual system [6]. This will not noticeably alter the visual appearance of a color and hence the image itself. Changing a more significant bit would cause a proportionately greater change in the visual appearance of a color. The main objective of steganography is to pass a message to a receiver without an intruder even knowing that a message is being passed which means that there should be no discernable change to the carrier. This is the first method to be tested and will involve encoding some of the basic processes required for later steganographic methods to be tested also. It will involve changing the LSB of one of the colors making up the RGB value of the pixel. This should have very little effect on the appearance of the image. This process will most likely result in the formation of new colors for the palette. It may be found that if the palette is ordered by luminance, there will be pairs of very similar colors. How noticeable that is, depends on the color profile used in the image to start with.

The common image with particular size is taken for testing the performance of all image Steganography methods and its size may vary according to type of format it is saved in.

The different parameters studied are Mean Square Error(MSE), Peak Signal to Noise Ratio(PSNR), Maximum change (MAX change). The variation of these parameters with hidden data size and also the internal variation of PSNR with MSE are displayed in tables and graphs for various techniques like Stego LSB 1BIT BMP, JPEG, BMP EDGES in figures 4(a),4(b), 4(c), Fig 5(a), 5(b), 5(c) , fig 6(a), 6(b)., Fig 7(a),7(b) and Fig 8.

4.1 Stego Lsb 1 Bit Bmp

Practical methods should allow for the use of the full image size, thus the amount of data that can be hidden is proportionate to the number of pixels in the image rather than to the colors in the palette. The only restriction is then the size of the image.

Single original BMP image is used to hide the data of six different sizes in it and comparison is made between steged images with the original image and the results are discussed in the figures 4(a), 4(b), 4(c).

Fig 4(a): Picture Size

SL. NO	Hidden Data Size	MSE	PSNR	MAX Chang e
1	0.868kb	2.32	44.46	28
2	2.54kb	2.41	44.30	28
3	3.38kb	3.55	42.62	35
4	5.52kb	4.03	42.07	29
5	8.73kb	4.20	41.89	33
6	11.8kb	5.69	40.57	30

Fig 4(b): Table of MSE, PSNR and MAXChanges for different Hidden Data Size.

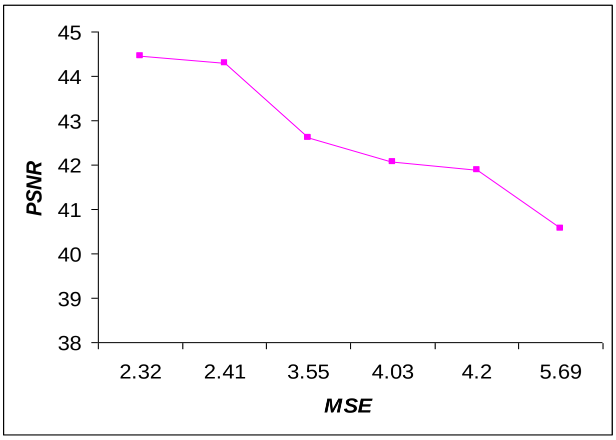

Fig 4(c): variation of PSNR with MSE

Hiding different data size in Same size BMP will bring noticeable changes in MSE. As the data size increases the MSE increases. There is a possibility that we can make out much change in the original image and hence steganography can be detected.

4.2 Stego Lsb 1 Bit Jpeg

Fig 5(a): Image Size: 86.8 KB

SL. NO	Data Size	MSE	PSNR	MAX Change
1	0.868kb	2.33	44.44	28
2	2.54kb	2.44	44.25	28
3	3.38kb	3.59	42.57	28
4	5.52kb	4.02	42.08	28
5	8.73kb	4.19	41.90	30
6	11.8kb	5.84	40.46	33

Fig 5(b): MSE, PSNR and MAX Changes

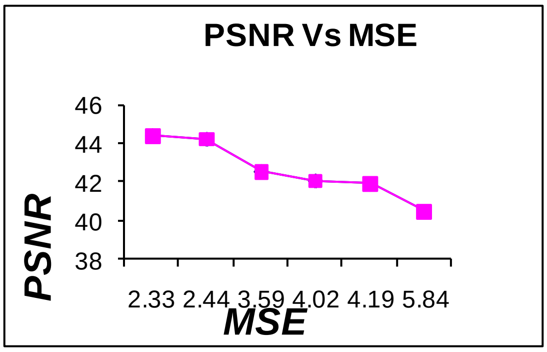

Fig 5(c): PSNR Vs MSE

Hiding different data size in

Same size JPEG image will bring noticeable changes in MSE. As the data size increases the MSE increases.there is a possibility that we can make out much change in the original image and hence steganography can be detected

4.3 Stego Lsb 1 Bit Bmp Edges Different Data Same tolerance:

Tolerance is the minimum value of a pixel chosen for hiding the message
Tolerance: 90
Available: 158043(22kb)

SL NO	Hidden Data size	PSNR	MSE	MAX
1	0.868kb	**76.78**	**0.0014**	1
2	2.54kb	**79.85**	**6.722e-004**	1
3	3.38kb	**79.85**	**6.722e-004**	1
4	5.52kb	**79.85**	**6.722e-004**	1
5	8.73kb	**79.85**	**6.722e-004**	1
6	11.8kb	**79.85**	**6.722e-004**	1

Fig 6(a): MSE, PSNR and MAX Changes

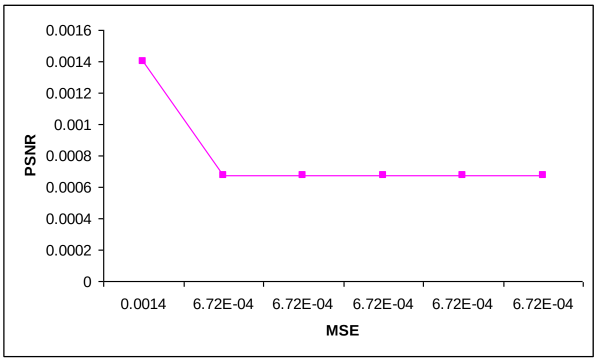

Fig 6(b): PSNR Vs MSE

In case of Edge steganography,for the constant tolerance and different data size there not much change in MSE and PSNR is almost constant and hence it shows this method is quite efficient and avoid finding stegnography easilySame data size and different tolerance value:

Needed: 0.868 kb

SL No.	Tolerance	Available pixels	PSNR	MSE(e-004)	Max
1	90	158043	**73.70**	**0.0028**	1
2	100	107781	**73.70**	**0.0028**	1
3	110	74214	**73.70**	**0.0028**	1
4	115	61749	**72.71**	**0.0035**	1
5	120	52341	**71.89**	**0.0042**	1
6	130	36276	**71.22**	**0.0049**	1

Fig 7(a): MSE, PSNR and MAX Changes

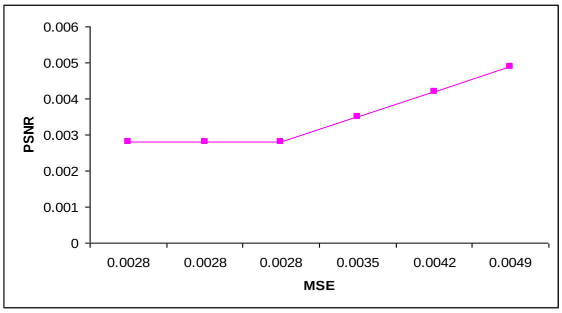

Fig 7(b): PSNR Vs MSE

Same proved in case of difference tolerance levels and same data size that not much change in the original image.

3.4 BMP, JPEG and Edge image steganography comparison

Image size:- **JPEG: 800*600** (*86.8 kb)*
BMP: 800*600 (*1.37 mb)*
Tolerance (for EDGE stego):*90*
HIDDEN DATA:-*868 bytes*

	BMP	**JPEG**	**EDGE**
MSE	2.32	2.33	0.0014
PSNR	44.46	44.44	**76.78**

Fig 8: JPEG Vs EDGE

Fig 8 shows the comparison of BMP, JPEG and EDGE steganography and it clearly indicates that the edge steganogrphy having less MSE and hence the best methods among the 3 to hide information of secure communication**.**

4.5 STEGO LSB 1 BIT AUDIO

Audio Sample Size: **16 bit**
Audio Sample Rate: **44 KHz**

SL NO	**Input Data (bytes)**	**Time Domain avg. Error (%)**	**Frequency Domain avg. Error (dB)**
1	40	0.00022	0.001306
2	60	0.00149	0.016906
3	80	0.00167	0.016809
4	100	0.00181	0.016744
5	120	0.00197	0.017440

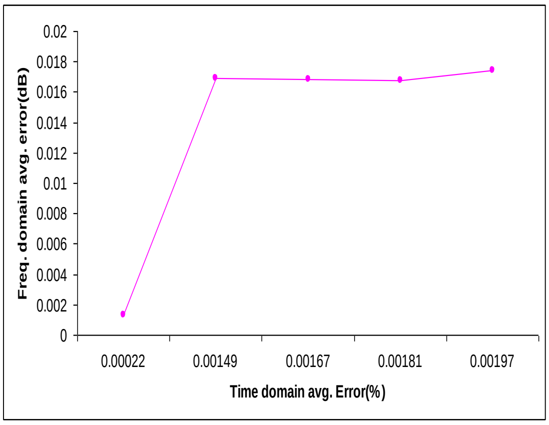

Fig 9: Error Rate in time and frequency domain

Audio Steganography can also be used to hide data for secure communication. Since the audio signal bit patterns are different than Image bit patterns this method cannot be compared with image based steganography. This method is considered with respect to time domain and frequency domain error. The error rate is very minimum for the above experimental results and can be used for secure data transfer.

V. CONCLUSION

The aim of this study is to investigate steganography and how it is implemented. Based on this work a number of methods of steganography are implemented and evaluated. The strengths and weaknesses of the chosen methods have been analyzed. Six steganography methods are implemented. The methods are chosen for their different strengths in terms of resistance to different types of steganalysis or their ability to maximize the size of the message they could store. All of the methods used are based on the manipulation of the single least significant bit of pixel values, which correspond to the message being hidden.

Hiding different data size in same size BMP image will bring noticeable changes in MSE. As the data size increases the MSE increases. There is a possibility that we can make out much change in the original image and hence steganography can be detected.

Hiding different data size in same size JPEG image will bring noticeable changes in MSE. As the data size increases the MSE increases, there is a possibility that we can make out much change in the original image and hence steganography can be detected.

In case of Edge steganography, for the constant tolerance and different data size there is no much change in MSE and PSNR is almost constant and hence it shows this method is quite efficient and avoid finding stegnography easily

Same proved in case of difference tolerance levels and same data size that not much change in the original image. The comparison of BMP, JPEG and EDGE steganography clearly indicates that the edge steganogrphy having less MSE and hence the best methods among the 3 to hide information of secure communication**.**

REFERENCES

1) Johnson Neil F., ZoranDuric, SushilJajodia, Information Hiding, and Watermarking - Attacks & Countermeasures, Kluwer 2001.
2) Westfield Andreas and Andreas Pfitzmann, Attacks on Steganographic Systems. Third International Workshop, IH'99 Dresden Germany, October Proceedings, Coputer Science 1768.pp. 61- 76, 1999.
3) Zollner J., H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, G. Wolf, Modelling the Security of Steganographic Systems, Information Hiding, 2nd International Workshop, IH'98 Portland, Oregon, USA, Computer Science 1525. pp. 344-354, April 1998.
4) Pfitzmann Birgit. Information Hiding Terminology.First International Workshop, Cambridge, UK, Proceedings, Computer Science 1174. pp. 347-350, May -June 1996
5) NabarunBagchi. Secure BMP image Steganography using Dual Security Model (I.D.E.A Image Intensity and Bit Randomization)and Max-Bit Algorithm, International journal of Computer Applications (0975-8887), Vol 1-No. 21, pp. 18, 2010
6) Ali Javed., AsimShahZad, RomanaShahzadi, Fahad Khan, Comprehensive Analysis and Enhancement of SteganographicStrtegies for Multimedia Data Hiding and Authentication, International Journal of Computer and Network Security, Vol 2, No.3, pp 75, March 2010.

.