CHAPTER 4
PERFORMANCE ANALYSIS OF STEGANOGRAPHY
FOR PROTECTION OF DATA

4.1 Introduction

In section 1.5.2 of Chapter 1 introduction on steganography is put forth. In this chapter different types of image steganography will be discussed in detail. Steganography has been a technique for secret communication used from ancient times. In ancient days’ people used wax tablets for hidden communication. In fact the first recorded occurrence of a form of steganography goes as back as the 5th century BC when in Greece Histaiacus shaved a slave’s head, tattooed a message on his skull and the slave was dispatched with the message after his hair grew back [21].

After paper came as the medium of communication, a paper mask with holes is shared among two parties, this mask is placed over a blank paper and the sender of the text will write his secret message through the holes then takes the mask off and fills the blanks so that the message appears as an innocent text, this hole can be done without any specific pattern. After that micro dots came in the World War II and had reused invisible ink and null ciphers. A Nazi spy used a text like this: “Apparently neutral’s protest is thoroughly discounted and ignored. Blockade issue affects pretext for embargo on by-products, ejecting suet and vegetable oils.” Using the 2nd letter from each word the secret message is revealed: “Pershing sails from NY June 1” [22] [23]. In 1945, Morse code was concealed in a drawing. The hidden information was encoded onto the stretch of grass alongside the river. It was done with a technique by drawing the long grass to denote a line and the short grass denoted a point. The decoded message read: “Compliments of CPSA MA to our chief Col Harold R. Shaw on his visit to San Antonio May 11, 1945” [24].
In the digital era steganography techniques have changed a lot became powerful with the support of computing power. After the development in internet, information theory, coding theory digital steganography also developed a lot. The digital Image steganography domains are mainly classification is shown below:

![Image Steganography Diagram]

**Fig. 4.1 Types of image steganography**

### 4.1.1 Spatial Domain Steganography

Spatial domain processing is executed directly on the pixel values of the image. These techniques use the pixel gray levels and the corresponding pixel color values directly for encoding the message bits. The spatial methods are most frequently employed by steganography tools because of fine concealment, great capability of hidden information and easy realization. It is also easy to implement. The most common algorithm belonging to this class of techniques is the Least Significant Bit (LSB) steganography method in which the least significant bit of the binary representation of the pixel is modified to represent the message bit. The other spatial
domain techniques are MSB (Most Significant Bit), RGB steganography – it is based on the pixel intensities etc. These will be explained detail in the coming sections of this thesis focuses the spatial domain steganography method.

4.1.2 Transform Domain Steganography

In frequency domain method pixel values are transformed and again processing is applied on the transformed coefficients. The secret data is hidden in significant areas of the covered image, which makes data non-changeable against attacks such as compression, image processing methods than LSB approach. This method provides a better security level to steganography method and has led to the development of several algorithms. This method transforms mainly categories into DCT, DWT and DFT. A lossless and reversible scheme have been introduced that use each block of quantized DCT coefficient in JPEG image for secret data. The method results in high stego image quality and achieves reversibility.

A matrix method is used for encoding in order to reduce the embedded noise in the signal. Wavelet Transform (WT) converts the spatial domain information to the corresponding frequency domain information. Wavelets are used in image specifically because it separately partitions the high frequency as well as the low frequency information pixel by pixel. This scheme mainly addresses the capacity as well as the robustness of the data hiding system.

The Discrete Fourier Transform is used to get the frequency component of each of the pixel values. The Discrete Fourier Transform (DFT) of spatial value \( f(x, y) \) for the image of size \( M \times N \) is defined in equation for frequency domain transformation.

\[
F(\chi, \psi) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi \frac{m\chi}{M} + \frac{n\psi}{N}}
\] (4.1)
\( \hat{f}(m,n) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(x,y) e^{j2\pi \left( \frac{mx}{M} + \frac{ny}{N} \right)} \)  

(4.2)

Where \( u = 0 \) to \( M-1 \) and \( v = 0 \) to \( N-1 \). Similarly, inverse discrete Fourier transform (IDFT) is used to convert frequency component to the spatial-domain value, and is defined in equation for transformation from frequency to spatial-domain. The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the real and imaginary part or with magnitude and phase [29].

### 4.2 Spatial Domain Steganography Techniques

An introduction to steganography and basic ideas are shared in chapter 1.4.2 and the definition of spatial domain steganography and its types are described in the previous section 4.1 & 4.1.1. Spatial domain techniques are well suited for protection of information and it is very easy to implement. A lot of spatial domain techniques are defined and it used for many applications. In that, some of the frequently used techniques are analyzing in the coming sections.

#### 4.2.1 LSB - Least Significant Bit Steganography

LSB steganography includes two schemes. One is the sequential embedding and the other is the scattered embedding. Taking images as example, sequential embedding will replace the pixels least significant bits with the message one by one sequentially. Scattered embedding makes the messages randomly scatter over the whole image by a random sequence to control the places where data can be embedded. The representative tools of LSB steganography include S-Tools, Hide and Seek, Hide4PGP and Secure Engine Professional. Image formats such as BMP, GIF, PNG images and WAV audio are deposited in lossless format and are taken as carriers.
by these tools. LSB embedding will however affect the relativity of the image’s LSB plane. Due to this some visual marks will be left in the stego-image [30]. This can be checked from the images that are shown below.

![Images](image1.png)

(a) (b) (c)

Fig. 4.2 (a) Cover image; (b) LSB plane of cover image; (c) LSB plane of stego-image.

LSB algorithm consecutively replaces the least significant bit of cover image with the message bits. This method makes use of the natural weakness of Human eyes visual capabilities in recognizing the slight difference of colors. The LSB method alters some or all the 8th bit of image’s data in such a way that the image’s changes are not at all perceptible for any human eyes. In such a manner, when using a color image, the LSB of each of the RGB components can be utilized. Therefore, the overall capacity for hiding secret data in a color image is almost triple of the same image in the grayscale mode. Hence in 8 bit or 24 bit images LSB algorithm acts the same way in the LSB bit and embeds information inside the stego image [31], [32], [27].

The steps in the algorithm of LSB are given below:

Step 1: The data bits are changed from decimal to binary.

Step 2: The Cover image is read.

Step 3: The cover image is converted from decimal to binary format.

Step 4: The text to be disclosed inside the image and is broken into bits.
Step 5: Take the first 8 bytes of data from the cover Image. (This step will be looped)

Step 6: Embed the data bit to be hide in the least significant bit of the cover image one by one.

Let us took the first byte of original data from the Cover image be:

Eg.:- 1 1 0 1 1 0 0 0

First bit of the data to be hidden: 1

Replace the least significant bit

```
1 1 0 1 1 0 0 0
```

This process will be continued for first 8 byte of data and conceal the first byte of data.

step 7: Continue the step 6 for all pixels in the file.

Images after embedding data using LSB Steganography.

![Fig. 4.3 Cover image](image1)

![Fig. 4.4 Stego image](image2)

There has been lot of improvements as well as modifications proposed in order to strengthen the LSB steganography technique. The above-mentioned algorithm is LSB-1 changing algorithm. LSB-2 changing algorithm also is defined in [31]. In this method the 2nd least bit is changed. The remaining process remains the same as
above. By changing the LSB of a pixel which causes in small changes in the intensity of the colors. These changes cannot be identified by the human eye; thus, the message is successfully hidden inside the image.

In [34] the secret data is firstly encoded by using the Vigenere encryption method to guarantee the protection of the hidden message. Afterward, the Lempel Ziv Welch (LZW) technique compresses the data to reduce the occupational capacity of the confidential data. Then, by utilizing the extended knight tour algorithm, on each bit stream of the data is spread out on the image to increase the robustness of the method. Like this different version of LSB algorithms are implemented.

**General advantages of spatial domain LSB technique are** [ from 33]:

- There is less chance for degradation of the original image.
- More information can be stored in an image.

**Disadvantages of LSB technique are:**

- Less robust, the hidden data can be lost with image manipulation.
- Hidden data can be easily destroyed by simple attacks.

### 4.2.2 MSB - Most Significant Bit Steganography

In image processing, the most significant bit or MSB also called the high-order bit is the bit position in a binary number having the greatest value. The MSB is referred to as the left-most bit because in positional notation of writing more of the significant digits further to the left. The MSB can also correspond to the sign of a signed binary number in one or two's complement notation where "1" means negative and "0" means positive. Then assign each bit a position number which ranges from zero to N-1 and here N is the number of bits in the binary representation used. Usually this is simply the exponent value for the corresponding bit weight in base-2 (such as in 2^31..2^0). Although a few CPU manufacturers assign bit numbers the opposite way,
the MSB will remain the most significant bit. This may be the main reason why the term MSB is quite often used instead of a bit number, although the primary reason is different number representations use different numbers of bits. By extension, the most significant bits (plural) are the bits closest to, and including, the MSB. For example, if 240 can be hidden in the first eight bytes of three pixels in a 24bit image [36].

```
PIXELS: 00100111 11101001 11001000
00100111 11001000 11101001
11001000 00100111 11101001
240: 01111000
RESULT: 00100111 11101001 11001000
10100111 11001000 01101001
01001000 00100111 01101001
```

Algorithm of MSB based steganography from [35], [36] is given below:

Step 1: Read the cover image and text message, which is to be hidden in the cover image.

Step 2: Convert text message in binary.

Step 3: Calculate MSB of each pixel of cover image.

Step 4: Replace MSB of the cover image with each bit of secret message one by one.

Step 5: Write stego image.

Algorithm to Retrieve Text Message

Step 1: Read the stego image.

Step 2: Calculate MSB of each pixel of stego image.

Step 3: Retrieve bits and convert each 8 bits into character.
Several variations of MSB algorithms have been discussed in various papers. MSB steganography is applicable for both grayscale and colour images. It will not affect the quality of the image much and human eye couldn’t find out the difference. Hence to an extent noise can be added into the image securely. The images are shown below after and before embedding the image using MSB steganography algorithm.

Fig. 4.5 Cover image                           Fig. 4.6 Stego image

Several studies show that while comparing LSB and MSB steganographic methods - LSB method is the better one according to [35], [36], [37].

4.2.3 RGB – RGB based Steganography

According to a computer an image is a collection of data. It is basically an array of integer numbers that represent light intensities at various points or pixels which makeup the image’s raster data. Digital images which are used by computers are typically stored as either 24-bit (RGB) or 8-bit (Grayscale) files. A 24-bit image will provide more space for hiding information as it has more pixels. However, 24-bit image bitmaps can be quite large also. All color variations for the pixels are derived from the 3 primary colors which are red, green, and blue. Each primary color is represented by one byte in the image. The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce different arrays of colors. The main purpose of the RGB color model is for display of still / video images in electronic systems, such as televisions and conventional photography as well as in computer.
Example RGB algorithm [41] is explained below. There is a key-number generated which indicates the gap value between two pixels. It contains the message which needs to be hidden. This key-number K2 is computed based on Message Length and image dimension i.e. Image Width and Image Height of the cover image. K2 is an integer which indicates the gap between pixels. It might contain the actual information which is hidden i.e. the information is inserted after every K2 pixel. The potential drawback of this is: the message distribution within the image is fixed in order. i.e., information is inserted after every K2 no. of pixels.

In this method a randomization based approach based on hashing with respect to the MSBs of the channels, to skip R no. of bytes, where R is a generated random no. based on the seed value S given to a random no. algorithm. To illustrate the procedure, a XOR operation between the least significant bits of the cover image and the image with hidden text will indicate the pixels which have changed. The message is seen clustered towards the upper portion of the stego image, where the message is seen to be distributed with skipping K2 bytes after storing information once. The result for the proposed approach, where the message has two approaches. distributed randomly (with respect to MSBs) after storing information once are better than the above.

The encoding algorithm is:

Input: Cover Image, Secret Message File

Step 1: Take as input the cover image, message/file to be hidden.

Step 2: Compress the original secret file. (Output: .zip file)

Step 3: Store the compressed file within the cover image

Step 4: Calculate the random number R and skip R no. of bytes.

Step 5: Repeat steps 3 and 4 till the message embedding is not over.
Output: StegoImage

Decoding process is:

Input: StegoImage

Step 1: Take the stego image as an input.

Step 2: Retrieve the message from the image by generating random number R and skipping R bytes every time, using the table indicator values.

Step 3: The output would be a compressed text file (.zip). By uncompressing the file, we would get the original text message file.

Output: Secret Message

Image after embedding the data using modulus RGB steganography technique.

Fig. 4.7 Cover image          Fig. 4.8 Stego image

This algorithm uses the concept of LSB and RGB steganography. Also, other variations of RGB algorithms are discussed in [40] variable bit steganography. In [42] MSB, LSB and RGB concepts are together used and embed data in smooth areas 1-3-4. LSBs Insertion technique has been utilized which hides data in 1-bit in 1 least significant bit of Red component (Most significant byte), 3-bits in 3 least significant bits of Green component and 4 bits in 4 least significant bits of Blue component (Least significant byte) of each selected pixel. This ratio 1:3:4 has been taken depending on their respective contribution of each red, green and blue component to the colors of RGB image.

**The main advantages of RGB steganography is**

- RGB is the most common and simplest model.
• The technique is more secure; third party cannot easily detect the presence of hidden data.

• One of the main advantages is its capacity, because it embeds large amount of data as compared to previous techniques.

4.2.4 PVD – Pixel Value Differencing based Steganography

In PVD-based schemes, to determine the number of embedded bits the difference between the consecutive pixels is taken. The larger the difference between the pixels is, the greater the number of secret bits that can be embedded. Usually, PVD-based approaches can achieve better imperceptible results compared with those typical LSB-based approaches with the same embedding capacity.

PVD based steganography uses the ideas of pixel intensity values. Several algorithms implemented in this concept. Pixel Value Differencing using secret key algorithm is described below [43].

Data Embedding Procedure: In this method capacity of each pixel is determined as in PVD method. Then password is appended with 5th bit of that pixel to create the hiding properties of data. If password matches, then the data is inserted directly. Else the inverted data is inserted into the pixel. Thus, somewhere data is inserted inverted and at the other places it is hidden directly.

If someone extracts the data without knowledge of secret key, then it is not of any use. Size of the embedded bit is also unknown without the key. So, with wrong key, the message can’t reshape. For a message as the image, extracted file will have some random dimensions which make it useless.

Step 1: Read cover image and create RGB matrixes. For gray scale image create gray matrix.

Step 2: Insert secret key and convert it into bit stream.
Step 3: Determine gray level variation near target pixel and thus determine capacity of that pixel say “n”.

Step 4: XOR $5^{th}$ bit of cover pixel with one bit of secret key.

Step 5: If password matches, “n” bits of message data are embedded in cover pixel.

Step 6: Otherwise data is inverted before embedding.

Step 7: Steps 3-6 are repeated till whole data is embedded.

Step 8: Then Size of message file is embedded into last 100 bytes using same procedure.

Step 9: Data is written in form of image file to obtain stegoimage.

**B. Data Extraction Process:**

For reconstructing of message file two things will be required. First is embedded data and the second one is the size of file.

Data is protected with password but it is not the same as combining cryptography and steganography. In that case data is first encrypted and after this it is then embedded. Here data is embedded and encrypted simultaneously. Thus, even if someone is able to collect all the embedded data, he needs cover file to reconstruct it. So, any modification in image will make it unrecoverable. Extraction of data can be done as follow:

Step 1: Open stego file and convert it in RGB matrixes.

Step 2: Enter password and convert it into 1D bit stream.

Step 3: Extract last 100 bytes of stego image to get message file sizes.

Step 4: Calculate capacity of stego pixels (as in data hiding process).

Step 5: Now extract last “n” bits of stego pixel as message data.

Step 6: XOR $5^{th}$ bit of stego image with secret key.
Step 7: If key matches insert data to message bit stream directly, else invert data first and then insert to bit stream.

Step 8: Repeat steps 4-7 till full size of message data is obtained.

Step 9: Now reshape message data and write to the file.

LSB steganography method gives a very low embedding capability, which is 1-bit per byte. On other hand PVD method gives very good embedding capability, in between 2-4 bits per byte. This is much better than LSB method. PVD steganography can be combined with encryption techniques to create a secure as well as high capacity steganography system can be developed.

While human vision perception is less sensitive to subtle changes in edge areas of a pixel, it is more sensitive to change in the smooth areas. Even when we modify last 1-2 bits of all image pixels in smooth area, it isn’t distinguishable to human eyes.

Fig. 4.9 (a) Lena.jpg                            Fig: (b) Lena_00.jpg & Lena_11.jpg

The standard image used for testing, Lena.jpg as in fig. 1(a) is modified. Last two bits of all image pixels are set to 00 in fig. 1(b) Lena_00.jpg and to 11 in fig. 1(c) Lena_11.jpg. There is no visible difference in between these images adapted from [43].

Different PVD algorithms are discussed [44], [45], [46]. The disadvantages of this method are,

In the existing PVD-based methods, typically a raster scanning order is used for dividing the embedding units, which allows only the vertical edges to be used. The
disadvantage here is that; such fixed division will leave some revealing clues for detectors. Most of the PVD-based steganography methods employ fixed continuous ranges to classify the differences between the pixel values in the embedding unit, which will lead to steps which are undesired and appearing at the PVD histogram.

Based on the properties of HVS and several experiments, embedding secret data into the edge regions cannot also produce more visual indistinguishable results and also enhances the capabilities against the statistical analysis. Therefore, those edge regions should be used as far as possible. Typical PVD-based approaches can embed more secret bits into busy areas.

4.3 Experimental Results and Comparison

The above-mentioned steganography techniques are analyzed using the image’s noise calculating techniques like PSNR (Peak Signal to Noise Ratio), MSE (Mean Squared Error), RMSD (Root Mean Square Error), SSIM (Structural Similarity Index) etc. Different sizes of images (24 bit) are tested as samples and find the values for comparison.

Test 1:

The sample image used for testing is image of penguins given below. The size of this image is 1.5 MB and the format of the image is .bmp. As part of our tests a standard text message of 445 bytes was inserted into the image and then the PSNR, MSE and SSIM value was calculated.
Table 4.1 Specification of Penguin

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the Cover image</td>
<td>1.5 MB</td>
</tr>
<tr>
<td>Format</td>
<td>BMP</td>
</tr>
<tr>
<td>Number of Pixels</td>
<td>1614600</td>
</tr>
</tbody>
</table>

Below table depicts the results for each of the steganography methods after embedding the message into the cover image Fig. 4.10 by comparing the original image and the stego image.

Table 4.2 Image quality of Penguin in various techniques

<table>
<thead>
<tr>
<th></th>
<th>LSB</th>
<th>MSB</th>
<th>RGB</th>
<th>PVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>93.07</td>
<td>92.9</td>
<td>94.2</td>
<td>92.1</td>
</tr>
<tr>
<td>MSE</td>
<td>0.00003</td>
<td>0.00003</td>
<td>0.00003</td>
<td>0.00029</td>
</tr>
<tr>
<td>SSIM</td>
<td>1.00</td>
<td>1.00</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Once the message has been embedded in cover image the change in the pixel value will not be noticed to human eyes in the stego image. It is quite hard to find the very minor colour changes in the cover and stego image by human eye. To identify this a histogram comparison is used by the steganalyst.

The histogram analysis for cover and stego image is given below.

Fig. 4.11 (a) Histogram – Cover image (Penguin)
(b) Histogram of Penguin image after embedding message [from 82]
Mean: 2014 Median: 223 Standard deviations: 56.2

From the histogram analysis it is clearly visible that the changes are minor hence the naked eye couldn’t find out the changes.

**Test 2:**

The sample image used for test 2 is image of squirrel given below.

The size of this image is 2.5 MB and format is BMP. As part of the tests a standard text message of 500 bytes was inserted into the image and then the PSNR, MSE and SSIM value was calculated.

![Fig. 4.12 Squirrel Cover image](image)

<table>
<thead>
<tr>
<th>Table 4.3 Specification of squirrel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the Cover image</td>
</tr>
<tr>
<td>Format</td>
</tr>
<tr>
<td>Number of pixels</td>
</tr>
</tbody>
</table>

Below table 4.4 depicts the results for each of the steganography methods after embedding the message into the cover image (Fig. 4.12) by comparing the original image and the stego image.

<table>
<thead>
<tr>
<th>Table 4.4 Image quality of squirrel in various techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LSB</strong></td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>PSNR</td>
</tr>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>SSIM</td>
</tr>
</tbody>
</table>

Mean: 100.2 Median: 87 Standard deviations: 48.5
Test 3:

The sample image used for testing is image of Taj mahal given below. The size of this image is 5.5 MB and format is BMP. As part of the tests a standard text message of 1000 byte was inserted into the image and then the PSNR, MSE and SSIM value was calculated.

Mean: 96.3 Median: 100 Standard deviation: 67.2

<table>
<thead>
<tr>
<th>Size of the Cover image</th>
<th>5.5 MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>BMP</td>
</tr>
<tr>
<td>Number of pixels</td>
<td>5760000</td>
</tr>
</tbody>
</table>

Below table 4.6 depicts the results for each of the steganography method’s PSNR, MSE and SSIM values after embedding the message into the cover image by comparing the original image and the stego image.

Table 4.6 Image quality of Taj mahal in various techniques

<table>
<thead>
<tr>
<th></th>
<th>LSB</th>
<th>MSB</th>
<th>RGB</th>
<th>PVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>90.11</td>
<td>89.9</td>
<td>92.24</td>
<td>90.02</td>
</tr>
<tr>
<td>MSE</td>
<td>0.00005</td>
<td>0.00004</td>
<td>0.00004</td>
<td>0.00033</td>
</tr>
<tr>
<td>SSIM</td>
<td>1.00</td>
<td>1.00</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>
From the three test cases it has been detected that the RGB method has good PSNR value and SSIM values compared to LSB, MSB and PVD techniques. All the techniques have minor changes between the values. In all these test cases .bmp file is used, but all types of files (.jpeg, .gif, etc.) can be used for these tests.

4.4 Summary

Steganography method is used to hide data into images without evoking suspicion to hackers as well as other attackers. There is different classification of steganography based on the algorithm used internally. In [82] compared LSB, MSB, RGB and PVD methods of steganography and also tested each one of them. The tests were done to identify which one of these methods is better from the other in terms of noise produced and changes made to the pixels. From our tests we could find out that the RGB based steganography method out performs the other methods in terms of PSNR and MSE calculation while SSIM calculation remained the same for all methods. It is also more difficult to identify by hackers when compared with LSB and MSB methods. Hence it is more advantageous over the other two methods for use of steganography.