WHAT IS STEGANOGRAPHY?

Steganography is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extracted at its destination. The word steganography is derived from the Greek words steganos (meaning hidden or covered) and the Greek root graph (meaning to write).

The purpose of steganography is to conceal and deceive. It is a form of covert communication and can involve the use of any medium to hide messages. It’s not a form of cryptography, because it doesn’t involve scrambling data or using a key. Instead, it is a form of data hiding and can be executed in clever ways. Where cryptography is a science that largely enables privacy, steganography is a practice that enables secrecy – and deceit.

HOW STEGANOGRAPHY IS USED NOWADAYS

Steganography has been used for centuries, but these days, hackers and IT pros have digitized it to do some pretty creative things. There are a number of apps that can be used for steganography, including Steghide, Xiao, Stegais and Concealment.

The practice of adding a watermark -- a trademark or other identifying data hidden in multimedia or other content files -- is one common use of steganography. Watermarking is a technique often used by online publishers to identify the source of media files that have been found being shared without permission.

STEGANOGRAPHY EXAMPLES INCLUDE:

- Writing with invisible ink
- Embedding text in a picture (like an artist hiding their initials in a painting they’ve done)
- Backward masking a message in an audio file (remember those stories of evil messages recorded backward on rock and roll records?)
- Concealing information in either metadata or within a file header
- Hiding an image in a video, viewable only if the video is played at a particular frame rate
- Embedding a secret message in either the green, blue, or red channels of an RRB image

TYPES OF STEGANOGRAPHY

1. Text Steganography − There is steganography in text files, which entails secretly storing information. In this method, the hidden data is encoded into the letter of each word.
2. Image Steganography − The second type of steganography is image steganography, which entails concealing data by using an image of a different object as a cover. Pixel intensities are the key to data concealment in image steganography.

3. Audio Steganography - Audio steganography is about hiding the secret message into the audio. It is a technique uses to secure the transmission of secret information or hide their existence.
4. Network Steganography - Network Steganography is a technique that uses common network protocols (the header field, the payload field or both) to hide a secret message.

TRANSFORM DOMAIN IMAGE STEGANOGRAPHY

Transform domain steganography is one of the techniques used for hidden exchange of information in frequency domain and it can be defined as the study of invisible communication that deals with the ways of hiding the existence of the communicated message.

SPATIAL DOMAIN IMAGE STEGANOGRAPHY

Spatial domain techniques, involves direct modifications on the pixel values whereas the transform domain techniques work on the transform domain coefficients that are obtained.

The spatial-domain embedding techniques are more common in comparison with the transform domain due to its simplicity in the embedding and extraction procedures, but with less strength. Nonetheless, the transform domain techniques are considered immune to the operations of image processing and are also considered less vulnerable to steganalysis attacks. The simplest method of conducting the process of data embedding through digital images is based on updating the values of cover pixels within the spatial domain. The image or spatial-domain methods apply different bit-wise techniques, which implement the noise manipulation and bit insertion by applying different simple techniques.

DIFFERENT KINDS OF STEGANOGRAPHY TECHNIQUES IN SPATIAL DOMAINS

1. LSB TECHNIQUE

LSB-Steganography is a steganography technique in which we hide messages inside an image by replacing least significant bit of image with the bits of message to be hidden. By modifying only the first most right bit of an image we can insert our secret message and it also make the picture unnoticeable, but if our message is too large it will start modifying the second right most bit and so on and an attacker can notice the changes in picture.

Embedding program:

```
input = imread('1.bmp');

message = fileread('Message.txt');

lenofmes = length(message)*8;
ascii_val = uint8(message);
```

```
binary_message = transpose(dec2bin(ascii_val,8));
binary_message = binary_message(:);
binary_message_len = length(binary_message);
binary_message = str2num(binary_message);

output = input;

[height width] = size(input);

count_embeded_bits = 1;

for i=1:height
    for j=1:width
        if(count_embeded_bits<=lenofmes)
            lsb = mod(double(input(i,j)), 2);
            changablebits = double(xor(lsb,binary_message(count_embeded_bits)));
            output(i,j) = input(i,j)+changablebits;
            count_embeded_bits = count_embeded_bits+1;
        end
    end
end

imwrite(output,'stego.bmp');
```

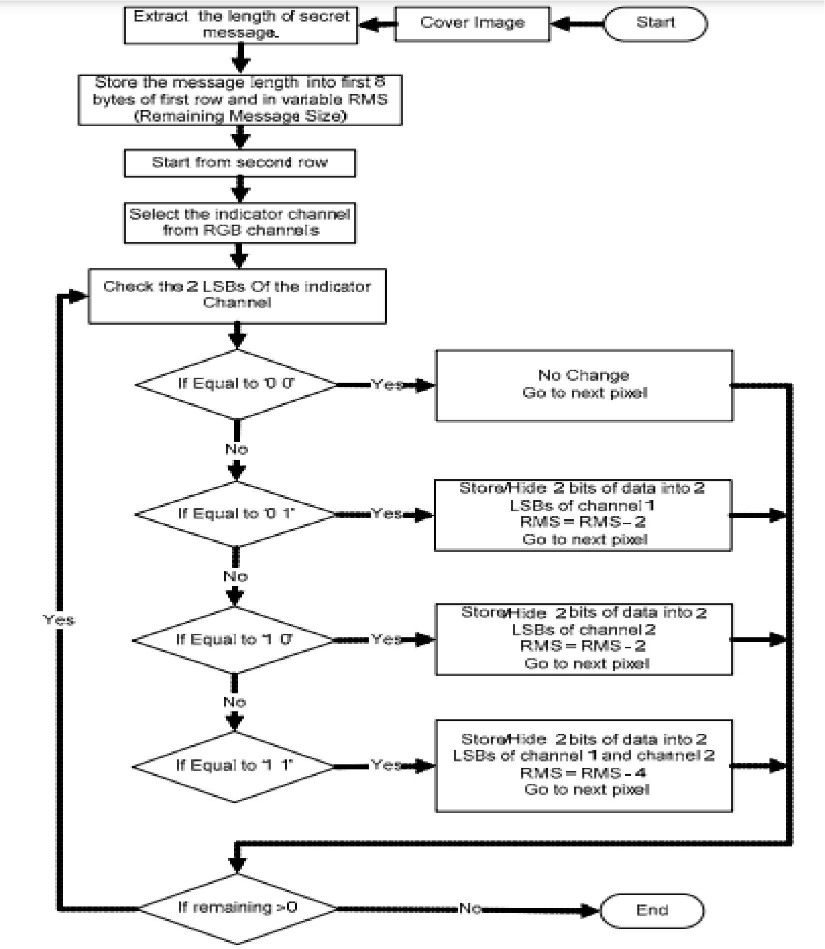
Extraction Program:

```
input = imread('stego.bmp');

[height width] = size(input);
counter=1;
for i=1:height
    for j=1:width
        lsb = mod(double(input(i,j)),2);
        extracted_bits(counter, 1) = lsb;
        counter = counter+1;
    end
end

bin_values = [ 128 64 32 16 8 4 2 1 ];
bin_matrix = reshape(extracted_bits,8,[]);
secret_text = char(bin_values*bin_matrix);

message = strsplit(secret_text,'&');
disp(message(1));
writelines(message(1),'Extracted Message.txt');
```

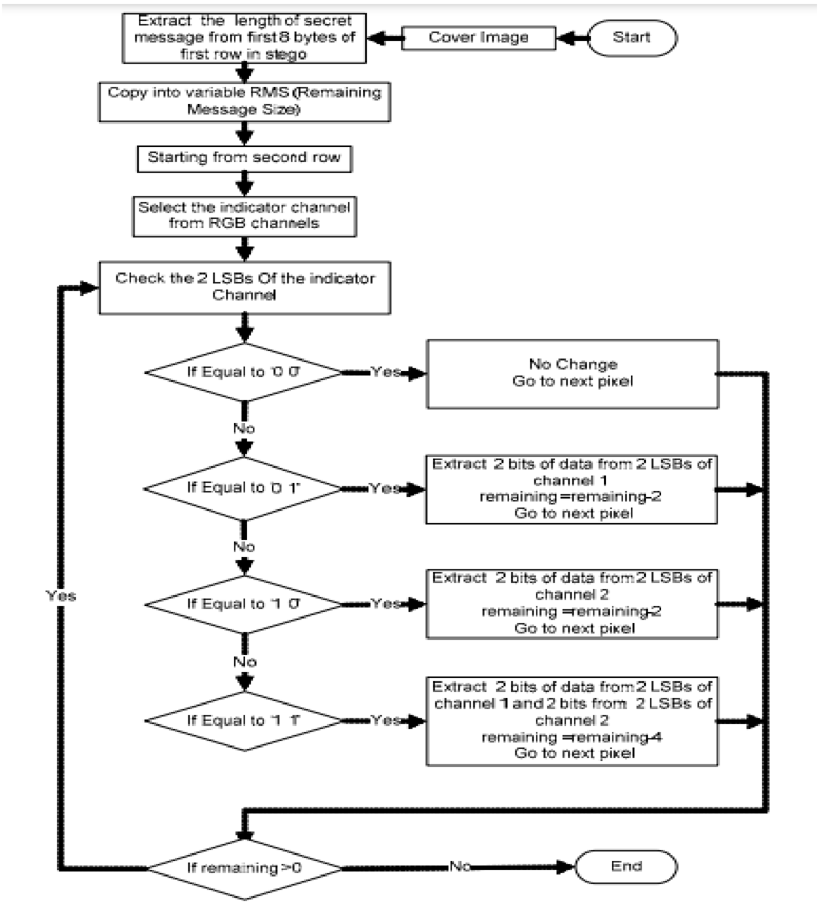
2. <u>GUTUB'S PIXEL INDICATOR TECHNIQUE (A.A.A GUTUB)</u>

Adnan Abdul-Aziz Gutub, Professor of Computer Engineering at Umm-Al-Qura University, Saudi Arabia proposed a paper of a new improved technique that takes the advantage of the 24 bits in each pixel in the RGB images using the two least significant bits of one channel to indicate existence of data in the other two channels. The stego method does not depend on a separate key to take out the key management overhead. Instead, it is using the size of the secret data as selection criteria for the first indicator channel to insert security randomness.

Construction Phase:


Start
Cover Image
Extract the length of secret message.
Store the message length into first 8 bytes of first row and in variable RMS (Remaining Message Size)
Start from second row
Select the indicator channel from RGB channels
Check the 2 LSBs Of the indicator Channel
If Equal to '0 0'
Yes
No Change Go to next pixel
No
If Equal to '0 1'
Yes
Store/Hide 2 bits of data into 2 LSBs of channel 1 RMS = RMS - 2 Go to next pixel
No
If Equal to '1 0'
Yes
Store/Hide 2 bits of data into 2 LSBs of channel 2 RMS = RMS - 2 Go to next pixel
No
If Equal to '1 1'
Yes
Store/Hide 2 bits of data into 2 LSBs of channel 1 and channel 2 RMS = RMS - 4 Go to next pixel
If remaining &gt;0
Yes
No
End

Extraction Phase:


Start
Cover Image
Extract the length of secret message from first 8 bytes of first row in stego
Copy into variable RMS (Remaining Message Size)
Starting from second row
Select the indicator channel from RGB channels
Check the 2 LSBs Of the indicator Channel
If Equal to '0 0'
Yes
No Change Go to next pixel
No
If Equal to '0 1'
Yes
Extract 2 bits of data from 2 LSBs of channel 1 remaining =remaining-2 Go to next pixel
No
If Equal to '1 0'
Yes
Extract 2 bits of data from 2 LSBs of channel 2 remaining =remaining-2 Go to next pixel
No
If Equal to '1 1'
Yes
Extract 2 bits of data from 2 LSBs of channel 1 and 2 bits from 2 LSBs of channel 2 remaining =remaining-4 Go to next pixel
Yes
If remaining>0
No
End

3. <u>GHOSAL'S TECHNIQUE(PAIR-WISE BIT BASED EMBEDDING)</u>

Sudipta Kr Ghosal, Greater Kolkata College of Engineering & Management, Kolkata, India, proposes a novel steganographic method to hide information within the spatial domain of the 24-bit color image. The proposed steganographic method works by considering the three channels (viz. red, green and blue) of each pixel of the cover image one by one up to the (maximum, if desire) last pixel and calculating the number of ones and zeroes in the red channel. Then, we calculate the absolute difference value of the number of zeroes and number of ones which is again divided by the total embedding channel numbers viz. green and blue which is 2 for a 24 bit color image. The resultant number of bits of the hidden data is embedded on the LSB part (in bit range of 0-3) of the green and blue bytes (channels) of each pixel of the cover image respectively. In the reverse way, we can extract the hidden data from the green and blue channels by checking the red channel of each pixel of the stego-image. Experimental results show that the proposed technique has improvised the hiding capacity of data (text as well as image) and at the same time retains good visual clarity of the stego-image.

Embedding Procedures:

Step 1: Load the 24-bit color image as cover.

Step 2: Load the data (usually, text or image) which is to be embedded.

Step 3: Consider the Red, Green and Blue channels of pixels starting from the first to a maximum of the end pixel to hide the secret information in the cover. That means, only the requisite number of pixels is needed from the cover which can hide the entire secret information.

Step 4: Calculate the number of 1's and number of 0's in the Red channel of each pixel.

Step 5: Calculate the absolute difference value of number of 1's and 0's in red channel.

Step 6: Divide the difference value results by the number of channels to be embedded in a pixel which is 2 for a 24 bit color image.

Step 7: Now, the resultant number of bits of the embedding size is to be embedded till a specified number of pixels and then data is to be embedded on the LSB (up to 3rd bit position) part of the Green and Blue bytes of each pixel of the cover image where the Red channel will act as an indicator.

Step 8: The final stego image is to be produced.

Extracting Procedures:

Step 1: Load the 24-bit color stego- image.

Step 2: Consider the Red, Green and Blue channel of pixels starting from the first of the stego-image to a maximum of the end pixel.

Step 3: Calculate the number of 1's and 0's in the Red channel of each pixel.

Step 4: Calculate the absolute difference value of number of 1's and number of 0's in red channel.

Step 5: Divide the difference value results by 2 in the same manner.

Step 6: Now, the resultant number of bits of the size data is to be extracted by traversing a specified number of pixels and depending on the size of the hidden data, the requisite secret data is to be extracted from the LSB part of the Green and Blue channels of each pixel of the stego image where the Red channel will act as an indicator.

Step 7: Now, the hidden data will be extracted from the stego image.

4. PVD METHOD:

Pixel-Value Difference method is mainly based on the principle of that human eyes are most sensitive to smooth area and least sensitive to the edge areas of an image i.e. the degree of distortion tolerance of an edge area is naturally higher than that of a smooth area. On the basis of this principle, more data bits are embedded in the smooth areas of an image. Actually, the determination of edge or smooth areas is dependent on the difference between two consecutive pixel values. In case of smooth areas, the values of the pixels are very close to each other. On the other hand, the pixels in the edge areas differ from their neighboring pixels by a large amount. Therefore, by checking the difference between two consecutive pixel values smooth or edge area of an image are determined in this technique. Actually, secret data bits are embedded in the image by modifying the difference between two consecutive pixel values.

5. EMD METHOD:

A novel method of steganogrphic embedding in digital images is described, in which each secret digit in a (2n+1)- ary notational system is carried by n cover pixels and, at most, only one pixel is increased or decreased by 1. In other words, the (2n + 1) different ways of modification to the cover pixels correspond to (2n + 1) possible values of a secret digit. Because the directions of modification are fully exploited, the proposed method provides high embedding efficiency that is better than previous techniques.