Volume 1, No. 5, December 2010

Journal of Global Research in Computer Science

ISSN-2229-371X

RESEARCH PAPER

Available Online at www.jgrcs.info

IMPLEMENTATION OF IMAGE STEGANOGRAPHY

Prof. Samir Kr. Bandyopadhyay*[1] and Rana Chakraborty[2],
[1]Department of Computer Science & Engineering
University of Calcutta, India,
Email: skb1@vsnl.com
[2]Dept. of Computer Science & Engineering
Institute of Science and Technology, Chandrakona, Medinipur
Email: rana_jui@yahoo.com

Abstract: Image steganography is a technique in which we can embed a message in a picture in such a way so that the looks of the picture will remain same. There are many ways to implement this image steganography. We have followed the Lest Significant Bit (LSB) approach. We have use MAT LAB software in our lab to practically implement LSB Steganography technique taking .bmp, .jpg, .gif, .tif, .png type picture.

Key word: LSB, HVS, and BMP

INTRODUCTION

Types of Steganography

Steganography can be classified into various types[6], depending upon the cover medium used. Cover medium may be text, image or audio or video file. Hence steganography can be said to occur in three major types:

i)Text Steganography
ii)Image Steganography
iii)Audio/video Steganography

Image Steganography

Image steganography is about exploiting the limited powers of the human visual system (HVS). Within reason, any plain text, cipher text, other mages, or anything that can be embedded in a bit stream can be hidden in an image[6].
Image steganography has come quite far in recent years with the development of fast, powerful graphical computers. Digital image is the most common type of carrier used for steganography. A digital image is composed of finite number of elements each of which has a particular location and value (gray scale). The processing of these digital images by means of a digital Computer is referred as digital image processing. Images are used for steganography in following ways.
The message in encrypted form or in the original form is embedded as the secret message to be sent into a graphic file. 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. The recipient 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. A stegoanalyst or attacker may try to intercept the stego-image. Computer based stenography allows changes to be made to what are known as digital carriers such as images or sounds. The changes represent the hidden message, but result if successful in no discernible change to the carrier. The information has nothing to do with the carrier sound or image. Information might be about the carrier such as the author or a digital watermark or fingerprint.

LSB Steganography: Least significant bit (LSB) insertion [1] is a common, simple approach to embedding information in a cover image [2]. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue colour components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data [3] . For example a grid for 3 pixels of a 24-bit image can be as follows:
(00101101 00011100 11011100)
(10100110 11000100 00001100)
(11010010 10101101 01100011)
When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:
(00101101 00011101 11011100)
(10100110 11000101 00001100)
(11010010 10101100 01100011)
Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size [3] . Since there are 256 possible intensities of each primary colour, changing the

LSB of a pixel results in small changes in the intensity of the colours. These changes cannot be perceived by the human eye - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference [2].

In the above example, consecutive bytes of the image data – from the first byte to the end of the message – are used to embed the information. This approach is very easy to detect [4]. A slightly more secure system is for the sender and receiver to share a secret key that specifies only certain pixels to be changed. Should an adversary suspect that LSB steganography has been used, he has no way of knowing which pixels to target without the secret key [5].

In its simplest form, LSB makes use of BMP images, since they use lossless compression. Unfortunately to be able to hide a secret message inside a BMP file, one would require a very large cover image. Nowadays, BMP images of 800 × 600 pixels are not often used on the Internet and might arouse suspicion[2]. For this reason, LSB steganography has also been developed for use with other image file formats.

LSB IN GIF

Since GIF images only have a bit depth of 8, the amount of information that can be hidden is less [6] than with BMP. Embedding information in GIF images using LSB results in almost the same results as those of using LSB with BMP. LSB in GIF is a very efficient algorithm to use when embedding a Reasonable amount of data in a gray scale image. GIF images are indexed images where the colors used in the image are stored in a palette. It is sometimes referred to as a color lookup table. Each pixel is represented as a single byte and the pixel data is an index to the color palette. The colors of the palette are typically ordered from the most used color to the least used colors to reduce lookup time.

Some extra care is to be taken if the GIF images are to be used for Steganography. This is because of the problem with the palette approach. If the LSB of a GIF image is changed using the palette approach, it may result in a completely different color. This is because the index to the color palette is changed. The change in the resulting image is noticeable if the adjacent palette entries are not similar. but the change is not noticeable if the adjacent palette entries are similar. some applications that use LSB methods on GIF images have low security because it is possible to detect even moderate change in the image

DIFFERENT FILE FORMAT

Image formats can be separated into three broad categories: lossy and lossless compression formats, and uncompressed formats Uncompressed formats take up the most amounts of data, but they are exact representations of the image. Bitmap formats such as BMP generally are uncompressed, although there also are compressed BMP files as well Lossy compression formats are generally suited for photographs. It is not suited for illustrations, drawings and text, as compression artifacts from compressing the image will standout. Lossy compression, as its name implies, does not encode all the information of the file, so when it is recovered into an image, it will not be an exact representation of the original. However, it is able to compress images very effectively compared to lossless formats, as it discards certain information. A prime example of a lossy compression format is JPEG.

Lossless compression formats are suited for illustrations, drawings, text and other material that would not look good when compressed with lossy compression. As the name implies, lossless compression will encode all the information from the original, so when the image is decompressed, it will be an exact representation of the original. As there is no loss of information in lossless compression, it is not able to achieve as high a compression as lossy compression, in most cases. Examples of lossless image compression are PNG and GIF. (GIF only allows 8-bit images.)

TIFF and BMP are both "wrapper" formats, as the data inside can depend upon the compression technique that is used. It can contain both compressed and uncompressed images.

Algorithm

To implement lsb steganography we have followed these steps.

```
Step 1.Read the cover image and we will get one array of three dimension

Step 2. Each pixcell of the image is represented by three values. One for R
one for G and the last one is B. Convert each value to its equivalent binary.

Step 3. Read the text which we want to keep hide and store in a variable.

Step 4. Now get the ASCII value of each character.

Step 5. Now convert the decimal ASCII value to its equivalent binary digits.

Step 6. FOR each pixel of the input image
        a. Convert the intensity value of the current pixel into the binary form
        b. Assign one bit of the text in to the last bit.

Step 7.Goto the next bit of the text
        Got to the next pixel.
        Goto step 5

Step 8. END
```

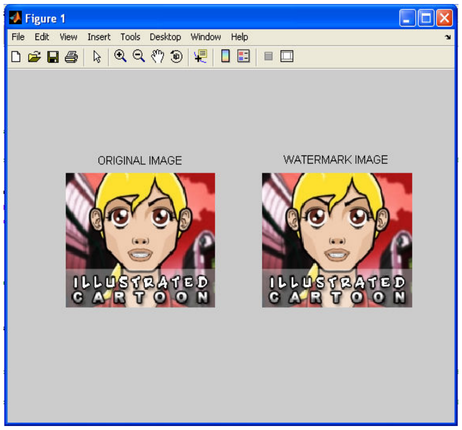
OUTPUT

This is the source image.

ILLUSTRATED
CARTOON

The text that will be embedded within this image is stored within a file named sample.txt. The contain of this file is “My name is Rana Chakraborty.This is a example of image steganography”.

This is the watermark image containing the text also.



ILLUSTRATED
CARTOON

CONCLUSIONS

With this project I have come to know a lot, especially about bit operations and bit-masking, something that I never understood before. I feel my self very interesting as I went on developing it. I became more interested in the subject the more I researched it. I have learned that while implementing Image Steganography is important, thinking of how to detect and attack it and the methods to do so are far more complex than actually doing the Steganography itself. There is a lot of research that is beginning to discover new ways to detect Steganography, most of which involves some variation of statistical analysis. It is interesting to see what other methods will be developed and how accurate they will be at detecting Steganography.

REFERENCES

[1] T. Morkel, J.H.P. Eloff, M.S. Olivier **“**An overview of image steganography”, University of Pretoria, 0002, Pretoria, South Africa.

[2] Johnson, N.F. & Jajodia, S., “Exploring Steganography: Seeing the Unseen”, Computer Journal,February 1998.

[3] Krenn, R., “Steganography and Steganalysis”,http://www.krenn.nl/univ/cry/steg/article.pdf.

[4] Wang, H & Wang, S, “Cyberwarfare: Steganography vs. Steganalysis”, Communications of the ACM, 47:10, October 2004.

[5] Anderson, R.J. & Petitcolas, F.A.P., “On the limits of steganography”, IEEE Journal of selected Areas in Communications, May 1998.

[6] Namita Tiwari, Dr.Madhu Shandilya, “Evaluation of Various LSB based Methods of Image Steganography on GIF File Format”, International Journal of Computer Applications (0975 – 8887)Volume 6– No.2, September 2010.