International Journal of Electronics Engineering, 1(2), 2009, pp. 161-165

A Novel Algorithm Based Design Scheme for Embedding Secret Message onto a Steganographic Channel

Vishal Bharti & Harish Bedi

Department of Computer Science & Engineering,
BRCM College of Engineering & Technology, Bahal, Haryana, INDIA

Abstract: *Security is more about process than technology. A first step for a corporation in managing security is to consider where the security is mostly required. The majority of attacks today are aimed at the valuable data which any organization possesses. One of the secure methods to keep data secure is known as steganography. Steganographic techniques have been used with success for centuries already. The intent of steganography is to conceal even the occurrence of a message. With steganography one can send messages without anyone having knowledge of the existence of the communication. Steganography is a solution which makes it possible to send news and information without being censored and without the fear of the messages being intercepted and traced back. While sending messages can be useful, it is also possible to simply use steganography to store information on a location. This work is related with the area of steganography. The work proposes an efficient scheme for hiding and retrieving information or secret data onto a steganographic medium. Image is taken as the medium for steganography. The proposed scheme uses Least Significant Bit (LSB) techniques for hiding data in images. The altered bit is always least significant bit, which does not alter the whole image in terms of chrominance and luminance of the image. Also as we are only concerned with the LSB, the effect on the image is totally negligible in terms of the resolution and quality of the image.*

Keywords: *Steganography, Stegocarrier, Stegofile, Stegokey, Ciphertext.*

1. INTRODUCTION

Steganography hides the existence of a message by transmitting information through various carriers. Its goal is to prevent the detection of a secret message. The most common use of steganography is hiding information from one file within the information of another file. For example, cover carriers, such as images, audio, video, text, or code represented digitally, hold the hidden information. The hidden information may be plaintext, ciphertext, images, or information hidden into a bit stream [1]. The cover carrier and the hidden information create a stegocarrier. A stegokey, such as a password, is additional information to further conceal a message. An investigator who does not possess the name of the file and the password cannot know about the file's existence [2].

Steganography is a method used by individuals or organizations to secretly communicate, whereby the transmitting agent hides a message within some medium, so that only an intended recipient can detect the message's presence. The word "steganography" means "covered writing" in Greek. Steganography, if possible must be used in conjunction with cryptography to create a double layer of protection for sensitive information [2, 3]. Steganography, coming from the Greek words **stegos**, meaning roof or covered and **graphia** which means writing, is the art and science of hiding the fact that communication is taking place. A brief idea of steganography is shown in Figure 1. Using steganography, one can embed a secret message inside a piece of unsuspicious information and send it without anyone knowing of the existence of the secret message.

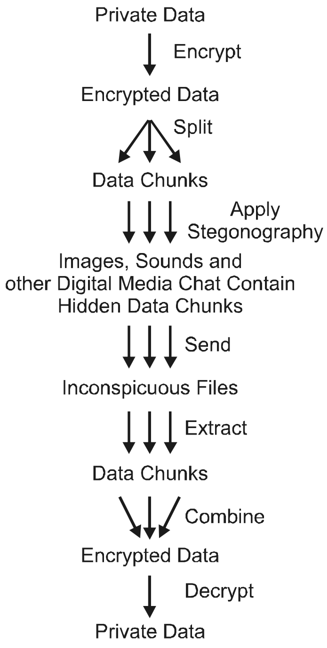

Figure 1: Steganography in Different Mediums

*Corresponding Author: mevishalbharti@yahoo.com

Steganography and cryptography are closely related. Cryptography scrambles messages so that they cannot be understood. Steganography on the other hand, will hide the message so there is no knowledge of the existence of the message in the first place [4]. In some situations, sending an encrypted message will arouse suspicion while an "invisible" message will not do so. Both sciences can be combined to produce better protection of the message. In this case, when the steganography fails and the message can be detected, it is still of no use as it is encrypted using cryptography techniques. Therefore, for good steganographic systems, knowledge of the system that is used, should not give any information about the existence of hidden messages. Finding a message should only be possible with knowledge of the key which is required to uncover it. Steganography can be treated as an added level of protection to cryptography. In layman's language one can say that Steganography is the art of hiding a small needle of information in a large hay stack of dummy information.

2. TERMS USED

The basic terms used in Steganography are:

- Cover
- Embedded Data
- Stego Data

The term "Cover" is used to describe the original, innocent message, data, audio, still, video and so on. When referring to audio signal Steganography, the cover signal is sometimes referred to as the "host" signal. The information to be hidden in the cover data is known as the "embedded" data. The "Stego" data is the data containing both the cover signal and the "embedded" information [4, 6, 8]. Logically, the processing of putting the hidden or embedded data, into the cover data, is sometimes known as embedding. Occasionally, when referring to image Steganography, the cover image is known as the container. Figure 2, depicts a basic overview of the steganographic channel, in which the message is encrypted and is embedded in a cover file which is then transmitted over an insecure channel over the network. At the receiver end, the message is removed from the cover data and is decrypted to get the original data.

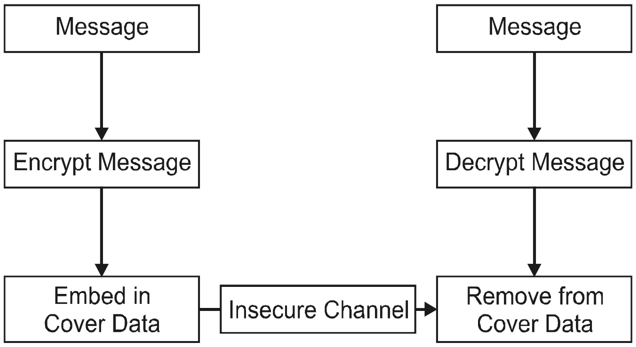

Figure 2: Basic Overview of Steganographic Channel

3. WHY STEGANOGRAPHY?

Steganography although, related to cryptography, they are not the same. The aim of Cryptography is to transform the message so that it cannot be understood by any one except the sender and the receiver who use special method to encrypt and decrypt the messages respectively. Steganography intention is to hide the existence of the message itself. More precisely, "the goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any adversary to even detect that there is a second secret message present." The advantage of Steganography is that it can be used to secretly transmit messages without the fact of the transmission being discovered. Often encryption techniques are clubbed with Steganography techniques to achieve higher levels of secrecy. However, it should be noted that the hidden message does not need to be encrypted to qualify as Steganography. The message itself can be put in plain English and still be a hidden message. However, most steganographers like the extra layer of protection that encryption provides.

3.1. Steganographic Techniques

Steganographic techniques [5, 8] can be grouped into three broad categories:

- Injection.
- Substitution.
- Generation of new files.

3.1.1. Injection

Injection refers to the insertion of a message into an existing medium. The simplest example is the use of the hidden attribute in Microsoft Word, which allows for hiding text with a special, hidden font. This very simple technique was used to store notes and references during the creation of this document. A casual observer can view the report and not be aware of rough notes that are easily revealed by going to Word's tools/options and clicking on "hidden text".

3.1.2. Substitution

This technique replaces data in the original file with a coded representation of the original message. The colors of "pixels", tiny elements of digital images are often represented by the value of a number contained in an eight-bit byte of data.

3.1.3. Generation of a New File

Both injection and substitution require a host file, sometimes called a container, in reference to images, and a host signal in reference to audio signals. Host files, contain embedded message but may also exhibit characteristics that reveal a pattern that can be used by steganalysis tools to detect the presence of the message. To eliminate this potential weakness, a coded message can be generated as part of an original computer generated text, audio or image file.

4. IMAGES AS STEGANOGRAPHIC CARRIER

The color and intensity of each pixel in a digital image is controlled by a binary code. It is impossible for the human eye to detect small changes in this code and accordingly secret information can be hidden inside it [11]. It is important to choose the right cover image when hiding information. The noisier the picture, the less detectable the imbedded information will be.

The binary code for a pixel is derived from the presentation of the 3 primary colors: red, green and blue. In a 24-bit image each primary color will be represented by 1 byte, i.e. 3-bytes are needed to represent a pixel. In 8-bit images each pixel is represented by a single byte. The number associated with the pixel points to a specific color in a color index table (palette) that contains the 256 possible colors. The image software paints the specified color at the associated pixel position on the screen. After doing this for all pixels in a 640 × 480 pixels image, the image is displayed on the screen.

A common image size is 640 × 480 pixels and 8 bits per pixel. This gives 256 colors per pixel and contains about 2500 kilobits of data. If a 24-bit image of 1024 × 768 pixels were to be stored, the amount of space required would exceed 2 Mbytes. These images might draw attention, because of their size. File compression techniques are used to make them more manageable.

5. STEGANOGRAPHY IN DIGITAL IMAGES

Insertions may hide information in every byte of information in the image or selectively hide information in “noisy” areas that’s more difficult to detect. Random scattering of information or redundant pattern encoding is also a techniques used to embed data in images. Most of these methods are classified under one of the following categories:

- Least significant bit insertion.
- Masking and filtering.
- Algorithms and transformations.

The success in the hiding of information depends on a combination of the type of image file and the technique used [6, 7].

Least Significant Bit (LSB) insertion is the most common and simplest approach to information hiding, unfortunately it is also very vulnerable to image processing and manipulation. Only lossless compressed images (BMP, GIF) can be used as cover images for this technique. Lossy compressed images (JPEG) do not reconstruct images exactly and will result in the destruction of hidden information.

LSB insertion involves the swapping of the secret bits with the least significant bits in every byte of the cover image. A 24-bit image will be able to hide 3 bits in every pixel (1 bit per byte, 3 bytes per pixel). This means that a 1024 × 768 image has the potential to hide a total of 300 Kbytes. Figure 3, graphically shows how much hidden information an image can carry without being detected.

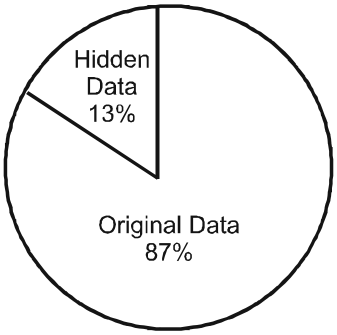

Figure 3: The Proportion of a 24-bit Cover Image that can be used to Hide Data.

Color limitations exist in 8-bit images and therefore an appropriate cover image must be chosen and used in combination with other techniques like reorganizing the palette, to successfully hide information.

On average only half of the least significant bits will be changed. This is illustrated with the following example:

The binary value for A is 1000001. We want to hide the binary value for A in the following 3 pixels, each represented by 3 bytes (24-bits):

00100111 11101001 11001000
00100111 11001000 11101001
11001000 00100111 11101001

After hiding the information the bit representation would look like this:

00100111 1110100**0** 11001000
0010011**0** 11001000 1110100**0**
11001000 00100111 11101001

Only 3 bits needed to be changed. Data can most probably be hidden in the least significant and second least significant bits, without the human eye being able to detect it. The color arrangement of the palette must be considered when LSB-insertion is used to embed the information.

6. PROPOSED SCHEME

We propose a method based on LSB insertion, where the data is hidden. This scheme uses image as our cover medium in which secret data is to be embedded. How data flows in the system is expressed in the form of flow charts which are depicted in Figure 4 and 5. This section also focuses on the algorithms used for encryption and decryption along with the various assumptions on which the above two algorithms are based.

6.1. Assumptions

There are certain constraints on which the proposed algorithms are based, these are listed as follows:

- The algorithms are based on the assumption that network is ideal i.e. what is sent is what is received **(WISIWIR).**
- Support for only few image and audio formats.

- Cannot hide binary files.
- The password **(key)** should be known to both sender and receiver (wherever required).

6.2. Flowchart and Algorithm for LSB Data Insertion

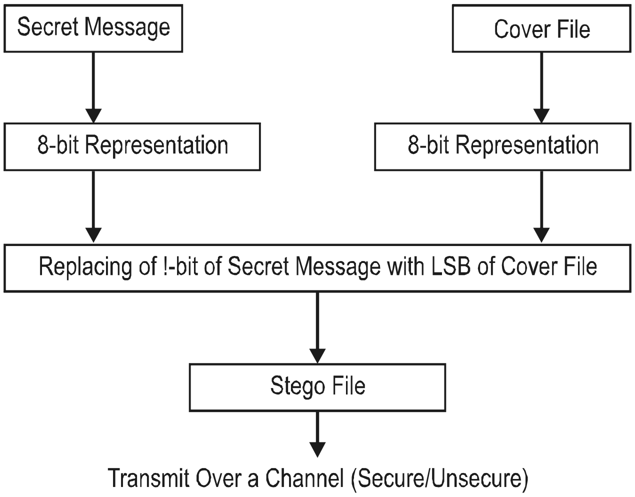
Secret Message
Cover File
8-bit Representation
8-bit Representation
Replacing of !-bit of Secret Message with LSB of Cover File
Stego File
Transmit Over a Channel (Secure/Unsecure)

Figure 4: Data Hiding at Transmitter End

Algorithm data_hide()

```
    // Input: A Cover file into which data is to be hidden.
    // A secret file which is the data to be hidden.
    //Output: A stego file which is the file obtained after hiding.
    // Function that implements LSB insertion technique
      while
        data is to be hidden
        do
            Get a character from the secret file
            Split the character into 8-bit format
        while
            all the bits are not hidden
            do
                Get a character from the cover file.
                Split the char into 8-bit format.
                Replace the LSB of the character with the bit to be hidden.
                Put the character back in the cover file (which is now stego file)
                Copy the remaining characters from the cover file to the stego file as it is.
```

6.3. Flowchart and Algorithm for LSB Data Extraction

Algorithm data_hide()

```
    // Input: A Stego file which contains hidden data
    //Output: An output file with the hidden data that has been extracted
    // Function that implements LSB extraction technique
while
    data is to be extracted
    do
        bits_extracted < 0
    while
        bits_extracted < 8
do
        Get a character from the stego file.
        Retrieve the LSB of the character.
        Bits_extracted = bits_extracted +1.
        Join the 8 bits obtained to form a character.
        Place the character at its appropriate position in the output file.
```

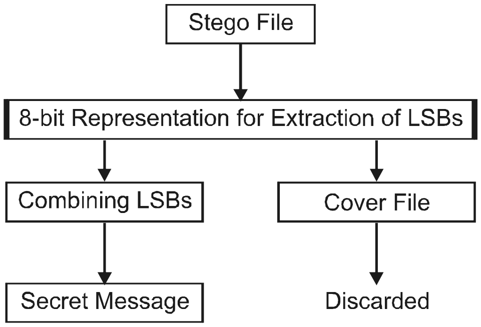

Figure 5: Data Extraction at Receiver End

7. CONCLUSION AND FUTURE ASPECTS

This paper gives a brief idea of the previously developed schemes of data hiding in images by replacing Least Significant Bit (LSB) insertion. The scheme of LSB insertion is also explained with an example. A new and efficient method for hiding data in images using LSB insertion is proposed for both the transmitter and the receiver end. Thus providing more security as well as the effect on the image, which we are using as a steganographic medium should be very less, which in turn is almost negligible for the human eye.

As the number of used LSBs during LSB coding increases or, equivalently depth of the modified LSB layer becomes larger, probability of making the embedded message statistically detectable increases and perceptual transparency of stego objects is decreased. Therefore, there is a limit for the depth of the used LSB layer in each sample of host image that can be used for data hiding.

Our future work includes working out for hiding data in images in frequency domain. Future work also includes working out on some of the anomalies in the proposed scheme like improving the algorithm to support other file formats which are not covered in this paper, also making algorithm to support for hiding binary files.

REFERENCES

[1] Vishal Bharti and Itu Snigdh, “Practical Development and Deployment of Covert Communication in IPv4, *Journal of Theoretical and Applied Information Technology*, (2008).

[2] Wootten, David R., “A Graphic User Interface for Rapid Integration of Steganography Software”, Naval Postgraduate School Monterey CA, (1996).

[3] Alkhraisat Habes, “Information Hiding in BMP Image Implementation, Analysis and Evaluation”, *Information Transmission in Computer Networks*, (2006).

[4] Whitepaper, *www.technicalinfo.net*, *www.technicalinfo.net/papers/CSS.html.*

[5] Neil F. Johnson, “Exploring Steganography: Seeing the Unseen”, George Mason University, **31**, (2), (1998).

[6] Donovan Artz, “Digital Steganography: Hiding Data within Data”, *IEEE Internet Computing*, (2001), *http://computer.org*.

[7] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for Data Hiding, *IBM Systems Journal*, **35**, (3-4), (1996), 313-336.

[8] Goyal, V. K., “Multiple Description Coding: Compression Meets the Network”, *IEEE Signal Processing Magazine*, **18**, (5), (2001), 74-93.

[9] Ahmidi, N. and Safabakhsh, R., “A Novel DCT-based Approach for Secure Color Image Watermarking”, Proceedings Information Technology, Coding and Computing, **2**, (2004), 709.

[10] Bas, P., Le Bihan, N., Chassery, J. M., “Color Image Watermarking Using Quaternion Fourier Transform”, IEEE International Conference on Acoustics, Speech, and Signal Processing, **3**, (2003), 521-524.

[11] Parisis, A. Carre, P., Fernandez, M. C., Laurent, N., “Color Image Watermarking with Adaptive Strength of Insertion”, IEEE International Conference on Acoustics, Speech, and Signal Processing, **3**, (2004), 85-88.

[12] Vishal Bharti, Kamna Solanki, “A Frequency Domain Manipulation based Approach towards Steganographic Embedding in Digital Images for Covert Communication”, Publication in *International Journal of Applied Engineering Research (IJAER)*, **4** (7).