Customized and Secure Image Steganography through Least Significant Bit Replacement

Kyaw Zin Latt, Lin Min Ko
kyawzinlatt55@gmail.com, linminko@gmail.com
Computer University (Maubin)

Abstract

Steganography is the process of hiding a secret message within a larger one in such a way that someone can not know the presence or contents of hidden message. The purpose of Steganography is to maintain secret communication between two parties. The basic structure of Steganography is made up of three components: the "carrier", the message, and the key. The carrier can be a painting, a digital image, an mp3 and even a TCP/IP packet among other things. It is the object that will 'carry' hidden message. A key is used to decode/decipher/discover hidden message. This can be anything from a password, a pattern, a black-light, or even lemon juice. This system presents how Steganography is used in a modern context while a practical understanding of what Steganography is providing how to accomplish it. This paper describes the design of a data hiding structure using Steganography. In this paper, we focus on the use of Steganography within digital images using Least Significant Bit (LSB) substitution.

Keywords: Steganography, LSB, stego, stegano

1. Introduction

Nowadays, communications are moving more and more towards electronic means, such as email, faxes and mobile phones. Every day hundreds or thousands of people interact electronically, where it is through e-mail, e-commerce (business conducted over the Internet). Secure communication is more profound than ever, recognizing that the conduct of much of our business messages and personal matters is being carried out through the medium of computers. Steganographic techniques can be used to protect the privacy of information.

Steganography is one of the important research subjects in the field of information security [7]. It enables secret communication by embedding messages in the texts, images, audio, video files or other digital carriers. Among all the image information hiding methods, LSB embedding is widely used for its high hiding capacity, and simple to realize. Much public Steganographical software, such as S-Tools and Steganos apply this technique. Therefore, it's with great significance to detect the images with hidden messages produced by LSB embedding effectively, accurately and reliably. And many experts made efforts on the LSB steganography and steganalysis research over the years [3].

Also, Stegnography is the art and science of hiding information in ways that prevent the detection of hidden messages. It includes a vast array of secret communication methods that conceal the message's very existence. Stegnography hides the message, so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an "invisible" message created with stegnographic methods will not. Modern steganograph's goal is to keep hidden message's mere presence undetectable traces in the cover medium.

In this paper, we focus the Least Significant Bit (LSB) that can help hiding messages on the digital image. Present work concentrates upon using Least Significant Bit conversion but is not limited to it. Binary numbers based stenographic study is implemented at small scale. This paper is an effort to explore the real power of binary numbers to hide the messages in secure and customized way.

This paper intends to use for image Steganography to illustrate the security potential of stenography for business and media use. The rest of this paper is organized as follows. In Section 2, we give a description of related works. In Section 3 we discuss the overview Steganography's theoretical background, followed by detailed discussion together with image processing. In Section 4, we describe the proposed system design for Steganography based on least significant bits and in Section 5 we describe the system implementation. Finally, we conclude this paper in Section 6.

2. Related Work

Steganography's is to hide very presence of communication as opposed to cryptography, which aims to make communication unintelligible to those who do not posses the right keys Andersen et al. [1]. The traditional approach to image encoding consists in the source coding, encryption and channel coding. The source coding is used to compress data and match it with the band-width of communication channel. However, the obtained data are sensitive to the communication noise and not protected against unauthorized use. To protect data against unauthorized access the encryption is accomplished. The encryption stage is performed separately from source coding. To reduce nuisance of the communication channel noise the channel coding is used which is based on the specialized

error correction codes able to detect and correct errors directly during data transmission. Both encryption and channel coding require the introduction of the redundant information in initial data that leads to the increase of data size and corresponded time of transmission.

Moreover, images, video, sound files and other computer files contain perceptually irrelevant or redundant information as "covers" or carriers to hide secret messages. After embedding a secret message into cover-image, stego-image is obtained. It is important that the stego image does not contain any detectable artifacts due to message embedding. A third party can use reliable identify which images contain secret messages, the Steanographic tool becomes useless. Obviously, the less information is embedding into cover-image, the smaller the probability of introducing detectable artifacts by the embedding process.

Fridrich et al. [6] developed a steganalysis method for detection of LSB embedding in 24-bit color images, which is based on analysis of close pairs of colors created by LSB embedding. Westfeld [13] performed the blind steganalysis based on statistical analysis of PoVs (pairs of values). This method, so-called χ_2-statistical test, gave a successful result to a sequential LSB steganography. Manchanda [10] pointed that this idea can also detect random embedding if applied to smaller part of images. Fridrich et al. [5] introduced hypothesis test to judge the existence of secret messages by LSB embedding, and the general framework was given in [12].

3. Background Theory

Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret. Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. Once the presence of hidden information is revealed or even suspected, the purpose of steganography is partly defeated [4]. The strength of steganography can thus be amplified by combining it with cryptography. Therefore, we discuss various forms of Stegnography for using secure and customized data.

3.1 Categories of Stegnography

There are many forms of Steganography including audio, video and image media. Almost all digital file formats can be used for Steganography, but the formats are more suitable those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display. The redundant bits of an object are those bits that can be altered without the alternation being detected easily. Image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding. Figure 1 shows the four main categories of file formats that can be used for Steganography.

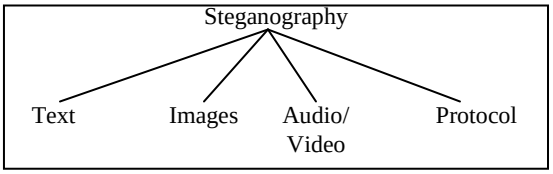

Figure 1: Different Kinds of Stegnography

Hiding information in text is historically the most important method of Steganography. An obvious method was to hide a secret message in every n^{th} letter of every word of a text message. It is only since the beginning of the Internet and all the different digital file formats that is has decreased in importance. Text Steganography using digital files is not used very often since text files have a very small amount of redundant data.

Given the proliferation of digital images, especially on the Internet, and given the large amount of redundant bits present in the digital representation of an image, images are the most popular cover objects for Steganography.

To hide information in audio files similar techniques are used as for image files. One different technique unique to audio Steganography is masking, which exploits the properties of the human ear to hide information unnoticeably. A faint, but audible, sound becomes naudible in the presence of another louder audible sound. This property creates a channel in which to hide information. Although nearly equal to images in Steganographic potential, the larger size of meaningful audio files makes them less popular to use than images.

The term protocol Steganography refers to the technique of embedding information within messages and network control protocols used in network transmission. In the layers of the OSI network model, there exist covert channels where Steganography can be used. An example of where information can be hidden is in the header of a TCP/IP packet in some fields that are either optional or are never used.

These forms of steganography often are used in conjunction with cryptography, so that the information is doubly protected; first it is encrypted and then hidden so that an adversary has to first find the information (an often difficult task in and of itself) and then decrypt it. The following formula provides a very generic description of the pieces of the Steganographic process:

Cover Medium + Hidden Data + Stego Key = Stegano Medium

In this system, the cover medium is the file in which we will hide the hidden data, which may also be encrypted using the stegano key. The resultant file is the stegano medium (which will, of course be the same type of file as the cover medium). The cover medium(and thus, the stegano medium) are typically image or audio files. In this system, we have focused on image files and therefore, refer to the cover image and stegano image. In this system, we utilize Image Steganography Method.

3.2 Image Steganography Techniques

Image Steganography has been widely studied by researchers. There are a variety of techniques used in which information can be hidden in images.

(i)Replacing Least Significant Bit

In image Steganography almost all data hiding techniques try to alter insignificant information in the cover image. For instance, this is a simple method that it is to place the embedding data at the least significant bit (LSB) of each pixel in the cover image. The altered image is called stego-image. Altering LSB doesn't change the quality of image to human perception but this method is sensitive a variety of image processing attacks like compression, cropping etc.

(ii)Replacing Moderate Significant Bit

This method is showed how to use the moderate significant bits of each pixel in the cover image to embed the secret message. It improves sensitivity to modification, but it degrades the quality of stego-image.

(iii)Transformation Domain Techniques

Other familiar data hiding techniques use the transformation domain of digital media to hide information. Functions such as the discrete cosine transform (DCT) and discrete wavelets transform (DWT). These techniques hide the messages in the significant areas of the cover image, which makes them robust against compression, cropping and other image processing attacks.

In this system, we make use of LSB method. An image file is merely a binary file containing a binary representation of the color or light intensity of each picture element (pixel) comprising the image. The simplest approach to hiding data within an image file is called Least Significant Bit (LSB) insertion. [11] Figure 2 takes the binary representation of the hidden data and overwrites the LSB of each byte within the cover image.

The Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. 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 color components can be used, since they are each represented by a byte.

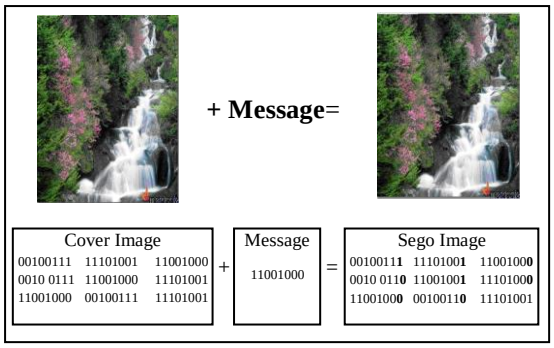

Figure 2: Model of Steganography Process with LSB

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. For example a grid for 3 pixels of a 24-bit image can be as follows:

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

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:

(00100111 11101001 11001000)
(0010011**0** 11001001 1110100**0**)
(11001000 0010011**0** 11101001)

Although the number was embedded into the first 8 bytes of the grid, only the 4 bolds 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. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. 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. [10] Consequently, how to hidden the embedded data in the image are discussed in the next subsection.

3.3 Embedding data

Embedding data, which is to be hidden, into an image requires two files. The first is the innocent-looking image that will hold the hidden information, called the *cover image.* The second file is the *message*—the information to be hidden. A message may be plaintext, ciphertext, other images, or anything that can be embedded in a bit stream [8].

When combined, the cover mage and the embedded message make a *stego image* as illustrate as Figure 3. A stego-key (a type of password) may also be used to hide, then later decode, the message. Most steganography software neither supports nor recommends using JPEG images, but recommends instead the use of lossless 24-bit images such as BMP. The next-best alternative to 24-bit images is 256-color or gray-scale images [9]

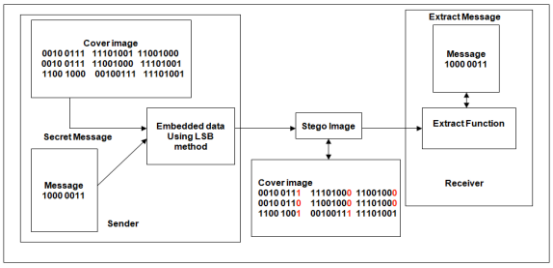

Figure 3: Internal Process of Steganography

4. Proposed System Design

Steganography is the process of hiding a secret message within a larger one in such a way that someone cannot know the presence or contents of the hidden message. Image Steganography allows for two parties to communicate secretly and covertly. In this system, the user makes up two procedures, that is, sending and receiving as shown in Figure 4.

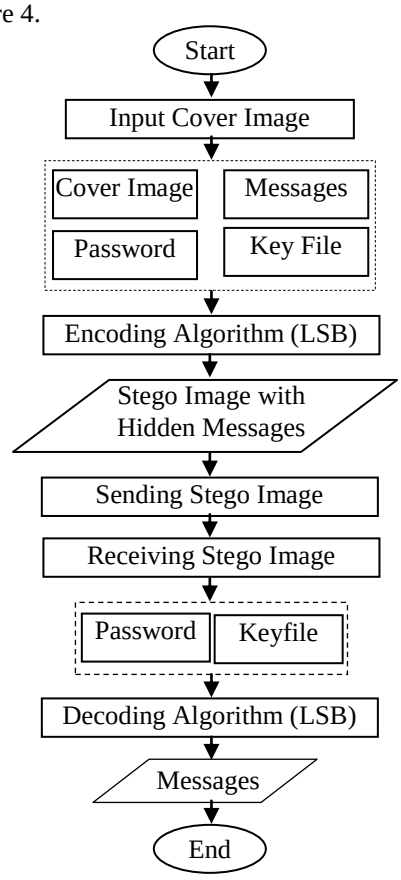

Figure 4: System Architecture by using Image Steganography

In sending procedure, there are four steps, acquiring cover image, accepting cover image in the system and putting up password and key files, merging cover image and require messages by using encoding algorithm (LSB) to form stego image and sending stego image to the receiver. In receiving procedure, two steps, namely, putting in typing password and keyfile and extracting the original or secret message by using decoding algorithm (LSB) are employed in this system.

(i) Sending Procedure

Before placing hidden data in cover image, this cover image is acquired from the scanner or digital camera in the system. The types of acquiring image are .jpeg, .gif, and gray-scale for this system. Subsequently, the user inscribes require password and enters the keyfile (like password to send effective and secure message) in the system. In addition, the receiving user must recognize sending user's password and keyfile. Then, the sending user writes the sending messages to hide with cover image file. And then, this system carries to embed these sending messages on the cover image as the stego image by using encoding algorithm along with LSB in Figure 5. The sender transmits the stego image to the receiver site.

Algorithms: Coding
Input: Cover-image, secret message, keys K1;
Output: Stego-image.
Step1: Read key K1 based on gray-Level or RGB ranges.
Step2: Read cover image (8-bit gray Image or 8-bit color image RGB Channel)
Step3: Decide No. of bits insertion into each range .
Step4: Read the secret message and Convert it into bit stream form.
Step5: Append with the message bits.
Step6: For each Pixel
6.1: Find image value g.
6.2: Decide the K-bits insertion based on gray or RGB ranges.
6.3: Find K-message bits and insert using method.
6.4: Decide new image Value g' using method.
6.5: Go to step6.
Step7: End

Figure 5: Encoding Algorithm (LSB)

(ii)Receiving Procedure

After the user sends the stego image to the receiver site, at the receiver, for extracting the message, this system accepts key file and password from the receiving user. Afterward, the receiver extracts the message using decoding algorithm with LSB in Figure 6.

Algorithm: Decoding
Input: Stego-image, keys K1;
Output: Secret information;
Step1: Read key K1 based on gray or RGB ranges.
Step2: Read the stego image.
Step3: Decide No. of bits extraction into each range.
Step4: For each pixel, extract the K-bits and save into file.
Step5: Find the bit stream
Step6: End

Figure 6: Decoding Algorithm (LSB)

Therefore, receiver site decodes the Stego image to identify secret messages from the sender. This system displays the original message to the user at the receiver site.

5. System Implementation

In this system, there have two sections that hide message and extract message. At hide section, the user selects image file and key file, writes password. And then, this system takes selected image file to hide the messages by using LSB method as demonstrate as Figure 7. The cover image and stego image are appeared in the system.

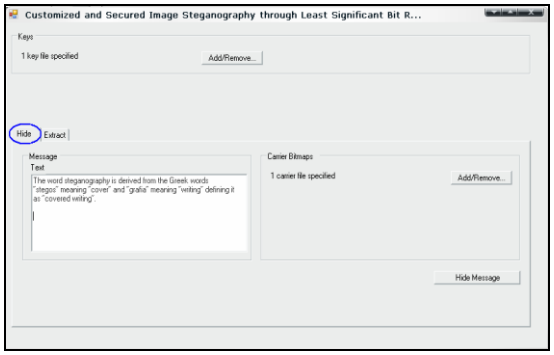

Figure 7: User Interface of Sender Site

This form is the comparison of two images files. Image from left side is plain image file or cover image and image from right side is image file with embedded message data or Stego image in Figure 8. But the user accepts as true as the two images are the same and cannot see image's changes with human eyes. Therefore, the hacker cannot care the messages in the image file and doesn't think to present the embedded data in the image file.

Figure 8: Images Comparison

These embedded messages within image file or cover image are converted to form stego image. The sender sent the stego image to the receiver site. At extract section, the receiver will receives this Stego image. The user at the receiver site extracts the stego image by using key file and password file. In this process, the decoding algorithm with LSB Method is used. if the key and password are true, the embedded messages in the images file are extracted and appeared in the text box.

At the receiver side, the user clicks the Extract Hidden Text button to extract the hidden message inside the stego image . The system shows the hidden messages in textbox as shown in Figure 9.

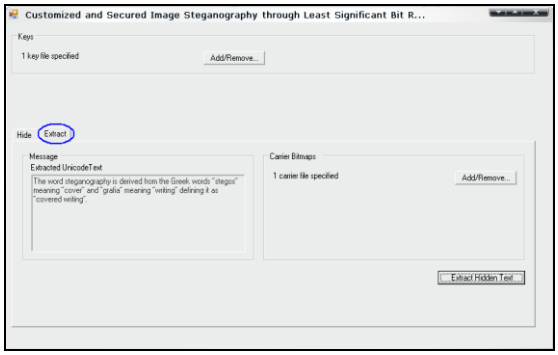

Figure 9: User Interface of Receiver Site

6. Conclusion

Image Steganography is important, thinking of how to detect and attack it. 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. Steganography will continue to increase in popularity over cryptography. As it gets more and more advanced, Steganalysis tools are detecting it. More encrypted data is being hidden using Steganography as the combination of the two provides an even hard target to crack.

References:

[1] Andersen R.J. and Petitcolas F.A.P. "On the Limits of Steganography". IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection 16 No.4, 474-481, 1998.

[2] Avcibas I., Memon N. and Sankur B. “Steganalysis using Image Quality Metrics”. IEEE Transactions on Image Processing, vol. 12, pp. 221–229, Feb. 2003.

[3] Bao F. and Wang X. “Steganography of Short Messages through Accessories” Pacific Rim Workshop on Digital Steganography, Japan, 2002.

[4] Cachin C. “An Information-theoretic Model for Steganography”, D. Aucsmith (Ed.): Information Hiding, 2nd International Workshop, vol. 1525 of Lectures Notes in Computer Science, pp. 306-318, Springer, 1998.

[5] Fridrich J., Du R., and Long M., “Steganalysis of LSB Encoding in Color Images”, Binghamton, 2007.

[6] Fridrich J. and Goljan M., “Practical Steganalysis of Digital Images-state of the Art” Proc. SPIE Photonics

West, Vol. 4675, pp. 1-13, San Jose, California, January, 2002.

[7] Fisk G., Fisk M., Papadopoulos C. and Joshua N. "Eliminating Steganography in Internet Traffic with Active Wardens", Available: Nov 1, 2002 http://citeseer.nj.nec.com/fisk02eliminating.html.

[8] Gonzalez R.C. and Woods R.E. "Digital Image Processing". Addison Wesley, Reading, 1992

[9]Johnson N. F. and Jajodia S. "Steganography: Seeing the Unseen". IEEE Computer, pp.26-34, February 1998.

[10]Manchanda S., Dave M., and Singh S. B., "Customized and Secure Image Steganography through Random Numbers Logic", Signal Processing,

[11] Morket T., Eloff J. H. P. and Olivier M. S., "An Overview of Image Steganography", Information of Computer Security Architecture (ICSA) Research Goup Department of Computer Science, University of Pretoria, 0002, Pretoria, South Africa.

[12]Revathi M., Bhattacharjee, Vijayalakshmi S., "Framework of LSB, Adaptive Steganalysis with IQM and Stegnography of Digital Media.

[13]Wesfeld A. and Pfitzmann A., "Attacks on Steganographic Systems", Lecture Notes in Computer Science, Vol. 1768, Springer-Verlag, Berlin, pp. 61-75, 2000.