IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

An Image Steganography Technique with High Hiding Capacity Based On 24 Bit Color Image

Manoj Kumar Sharma[*1], **Noor Mohd**[2], **Avnish Kumar Sharma**[3]
[*1] M. Tech Scholar, Department of Comp.Sci.and Engg., Graphic Era University, Dehradoon, India
[2] Assistant Professor, Department of Comp.Sci.and Engg., Graphic Era University, Dehradoon, India
[3]Assistant Professor, Department of MCA., Marathwara Institute of Technology, Bulandshahr, India
manojcs2005@rediffmail.com

Abstract

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the internet. For hiding secret information in images, there exist a large variety of Steganography techniques some are more complex than others and all of them have respective strong and weak points. Different applications may require absolute invisibility of the secret information, while others require a large secret message to be hidden. This paper intends to give an overview of image Steganography, its uses and techniques. It also attempts to identify the requirements of a good Steganography and briefly reflects on which steganographic techniques are more suitable for which applications.

Keywords: Steganography, Cryptography, Data Hiding, Encryption, LSB

Introduction

One of the reasons that intruders can be successful is the most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may reveal the information to others, modify it to misrepresent an individual or organization, or use it to launch an attack. One solution to this problem is, through the use of Steganography. Steganography is a technique of hiding information in digital media. In contrast to cryptography, it is not to keep others from knowing the hidden information but it is to keep others from thinking that the information even exists. Digital steganography, or information-hiding schemes, can be characterized utilizing the theories of communication [6]

Steganography become more important as more people join the cyberspace revolution. Steganography is the art of concealing information in ways that prevents the detection of hidden messages. Steganography include an array of secret communication methods that hide the message from being seen or discovered.

The growing possibilities of modern communications need the special means of security especially on computer network. The network security is becoming more important as the number of data being exchanged on the internet increases. Therefore, the confidentiality and data integrity are requires to protect against unauthorized access and use. This has resulted in an explosive growth of the field of information hiding

Information hiding is an emerging research area, which encompasses applications such as copyright protection for digital media, watermarking, fingerprinting, and Steganography. In watermarking applications, the message contains information such as owner identification and a digital time stamp, which usually applied for copyright protection.

The most popular one is embedding a message into a colored image using LSB [8]. In this method the data is being hidden in the least significant bit of each pixel in the cover image. In this method, we can take the binary representation of the hidden data and overwrite the LSB of each byte within the cover image. If we are using 24-bit color image, the amount of change will be minimal and indiscernible to the human eye. But this techniques hiding capacity is not so good and the attackers can easily destroy the hidden data by changing the least significant bit with less degradation of image quality. Therefore, Wang et al. [5] proposed a method using the genetic algorithm to embed secret data into each host pixel. However, using the genetic consumes huge computational time and the solution of a mapping function is not optimal. In 2002, Chang et al. [9] offered their dynamic programming strategy to pick out the best solution from all possible conditions that can significantly reduce the computation time.

ISSN: 2277-9655
Impact Factor: 1.852

In Steganography, the possible cover carriers are innocent looking carriers (images, audio, video, text, or some other digitally representative code) which will hold the hidden information. A message is the information hidden and may be plaintext, cipher text, images, or anything that can be embedded into a bit stream. Together the cover carrier and the embedded message create a stego-carrier.

Hiding information may require a stego key which is additional secret information, such as a password, required for embedding the information. For example, when a secret message is hidden within a cover image, the resulting product is a stego-image. A possible formula of the process may be represented as:

cover medium + embedded message + stego key = stego-medium

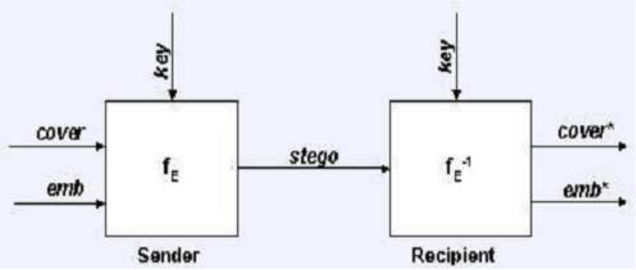

Figure 1: Graphical Version of the Steganographic System

FE: steganographic function "embedding"
FE (-1): steganographic function "extracting"
Cover: cover data in which *emb* will be hidden
Emb: message to be hidden
Stego: cover data with the hidden message

The advantage of Steganography is that it can be used to secretly transmit messages without the fact of the transmission being discovered. Often, using encryption might identify the sender or receiver as somebody with something to hide.

Related Work and Methods of Data Hiding

There are many steganography tools which are capable of hiding data within an image. These tools can be classified into five categories based on their algorithms: (1) spatial domain based tools; (2) transform domain based tools; (3) document based tools; (4) file structure based tools; and (5) other categories such as video compress encoding and spread spectrum technique based tools [4].

The spatial domain based steganography tools use either the LSB or Bit Plane Complexity Segmentation (BPCS) algorithm. The LSB algorithm uses either a sequential or scattered embedding schemes for hiding the message bits in the image. In the sequential embedding scheme, the LSBs of the image are replaced by the message bit sequentially (i.e. one by one in order, as mentioned in the introduction). In the scattered embedding scheme, the message bits are randomly scattered throughout the whole image using a random sequence to control the embedding sequence.

Two basic types of LSB modifications can be used for the embedding schemes described above. They are LSB replacement and LSB matching. In LSB replacement, the LSB of the carrier is replaced by the message bit directly. On the other hand, in LSB matching if the LSB of the cover pixel is the same as the message bit, then it remains unchanged; otherwise, it is randomly incremented or decremented by one. This technique, however, requires both the sender and the receiver to have the same original image, which makes LSB matching very inconvenient [4].

LSB is a simple approach to embedding information in an image. Applying LSB technique to each byte os a 24-bit image, three bit can be encoded into each pixel, as each pixel is represented by three bytes. Applying LSB technique to each byte of an 8-bit image, only one bit can be encoded into each pixel, as each pixel is represented by one byte. For example, if we use 8-bit image to hide the letter A (has the binary value 01000001), we need eight pixels. Suppose the original eight pixels are:

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

Inserting the letter A (as a binary value) into these eight pixels will give following (starting from left to right):

(0010011**0**) (11101001) (11001000) (0010011**0**) (1100100**0**) (11101000) (11001000) (00100111)

Only the emphasized bits are changed. Care needs to be taken in the selection of the cover image, so that changes to the data will not be visible in the stego-image.

The current Steganography tools based on the LSB algorithms include S-Tools, Hide and Seek, Hide4PGP and Secure Engine Professional. These tools support BMP, GIF, PNG images and WAV audio files as the carriers [4]. Each of these tools has unique features. S-Tools reduce the number of colors in the image to only 32 colors. Hide and Seek makes all the palette entries divisible by four. In addition, it forces the images sizes to be 320x200, 320x400, 320x480, 640x400 or 1024x768 pixels.Hide4PGP embeds the message in every LSB of an 8-bit BMP images, and in every fourth LSB of a 24-

bit BMP image. These applications are flawed because they do not analyze the image file after it has been embedded with data to see how vulnerable it is to steganalysis.

The transform domain based steganography tools embed the message in the transform coefficients of the image. The main transform domain algorithm is JSteg [4].These applications can only work with JPGs because most other image formats do not perform transforms on their data.

The document based steganography tools embed the secret message in document files by adding tabs or spaces to .txt or .doc files [4]. These applications are limited because they only work with document files. They also cannot hide much data because there are a very limited number of tabs or spaces they can reasonably be added to a document. In addition, they are vulnerable to steganalysis because it is easy for an attacker to notice a document file that has been embedded with additional tabs or spaces.

The file structure based steganography tools embed the secret message in the redundant bits of a cover file such as the reserved bits in the file header or the marker segments in the file format [4]. These applications cannot hide very large data files because there are a very limited number of header or marker segments available for embedding hidden data.

There are also steganography tools based on video Compression and spread spectrum techniques. The large size Of video files provides more usable space for hiding of the message. The spread spectrum technique spreads the energy of embedded message to a wide frequency band, making the hidden message difficult to detect [4]. These steganography tools are inconvenient because they require the users to send an entire video file every time they want to send hidden data.

Concept of Steganography

IT is the art and science of **hiding information** by embedding secure and confidential messages (such as Identity, Password etc.) within other harmless digital file format (such as JPEG, BMP, PNG, WAV, MP3, ECT.). In one sentence, Steganography is a technique of hiding information in digital media

The purpose of Steganography is covert communication to hide a message from a third party. In contrast to cryptography, it is not to keep others from knowing the hidden information but it is to keep others from thinking that the information even exists.

The Steganography process generally involves placing a hidden message in some transport medium, called the carrier. The secret message is embedded in the carrier to form the Steganography medium. The use of a Steganography key may be employed for encryption of the hidden message and/or for randomization in the Steganography scheme. Message is the data that the sender wishes to remain it confidential. It can be plain text, cipher text, other image, or anything such as a copyright mark, a covert communication, or a serial number. Encryption key is known as *stego-key*, which ensures that only recipient who knows the corresponding decoding key will be able to extract the message from a *cover-object*. The *cover-object* with the secretly embedded message is then called the *Stego-object.*

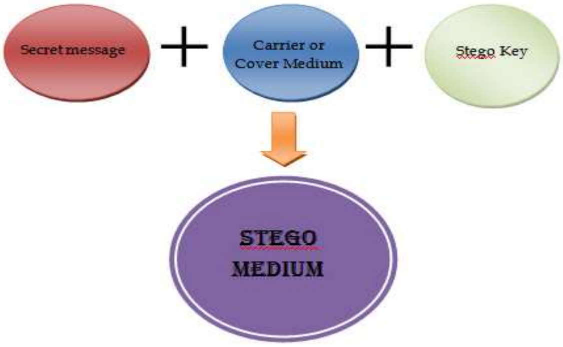

Figure 2: Steganography scheme

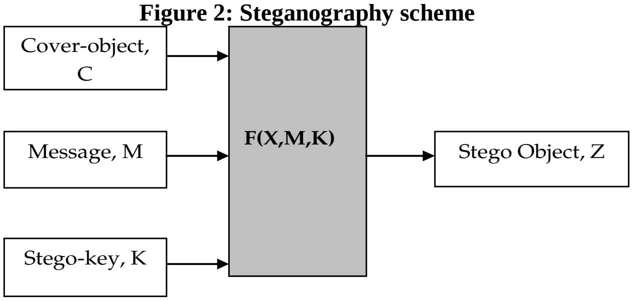

Figure 3: Use the function (F)

Recovering message from a *Stego-object* requires the *cover-object* itself and a Corresponding decoding key if a *Stego-key* was used during the encoding process. The original image may or may not be required in most applications to extract the message.

We shall perform steganography on image files and we shall hide the encrypted message into image files in an encrypted format thus achieving a multiple cryptographic system. The most commonly used technique for image steganography is bit insertion where the LSB of a pixel can be modified. Ref [5] explains various other techniques involve spread spectrum, patch work, JPEG compression etc. Instead of traditional LSB encoding, we will use a modified bit encoding technique to achieve image steganography in which each pixel will store one byte of data.

ISSN: 2277-9655
Impact Factor: 1.852

Algorithms and Techniques

Data Hiding Algorithm

The key difference between our application and the other programs that implement LSB embedding is that our application ranks images based on their suitability as cover images for some data. This allows a user to pick an image suited for hiding particular data, which reduces the threat of steganalysis attacks. No other application we are aware of currently offers this functionality of matching an image to the data to be hidden.

In the application the user first specifies the data that they would like to hide, which can be in any file format. The application then encrypts this data using the recipient's RSA public key [7]. Once the encrypted data is obtained, the procedure described in the following paragraph is repeated for each image in a user's image library.

Each bit of the encrypted data is compared to the least significant bit of the pixel bytes in an image. The comparisons are made starting from the first byte in the image until the last byte that permits all the data to be hidden in that image. The application cycles through the pixels of the image looking for the block of bytes that result in the least number of LSB changes. The image is then given a rank based on the percentage of least significant bits that match the encrypted data bits. Consider, for example 10 bits of encrypted data that need to be hidden in an image with a bit pattern of 10000000001. If some block of bytes in the image has least significant bits with a pattern 1000000011, this would result in the image receiving a ranking of 90%, because nine of the ten bits are an exact match.

This approach takes 24 bit color images as cover and the embedding data may be text of image.

In this approach we have considered each pixel one by one and embedded the secret data into it depending upon the number of color channel information's in each pixel. To embed the secret data, we have checked the number of 1's and 0's in red channel of consecutive pixels starting from the first till the last to hide the entire data. Then, we have calculate the absolute difference value of number of 1's and 0's in each red channel which is again divided by the number of channel to be embedded in a pixel which is 2 for a 24 bit color image as there are three channel namely red, green and blue whereas embedded channel is only green and blue in our approach.

Now we are giving an example of how the above method works. Let we assume, the bit pattern (R, G and B) for two consecutive pixels of a 24-bit color image is as shown below:

11011011	00010110	10000011
01001100	00110110	10101011

Now, if we want to embed a character A (has the binary value 01000001), we need to follow the above method. So, as per our method:

The number of 1's in Red byte is 6
The number of 0's in Red byte is 2
So the absolute difference value is (6-2) =4
Dividing the above result by 2 yields=4/2=2.

So, bit embedded on the LSB part of the green and blue byte is 2. Also, for the second R byte the bit embedded on the LSB part of the blue bite is 1.

Now, the bit stream of the stego image will be as follow shown in below:

11011011	000101**01**	100000**00**
01001100	001101**0**0	101010**0**1

So, by replacing only 6 bits in 4 numbers of selected bytes, we can hide the binary string 01000001.

Encryption Algorithm

As mentioned previously, the data to be hidden is first encrypted using the RSA public key algorithm. Encrypting the data before hiding it provides defense in depth, and makes the job of the attacker more difficult if their goal is to recover the secret data [7].

The application uses the RSA algorithm for two reasons. First, by using a public key algorithm the need for a private shared key between the sender and recipient of the data is eliminated. Shared keys are impractical because they require a secure way of distributing the key to every person who you may want to communicate with. A public key for a person can be distributed fairly easily by publishing it on a website, or by emailing it to people you expect would need to send you secret information. Second, the RSA algorithm is also widely known and demonstrably secure if large enough prime numbers are used to generate the keys. Using an algorithm such as RSA which is public knowledge is in keeping with the principle of open design of secure software systems. Adhering to this principle was also the reason we chose not to use our own encryption algorithm

C. *Injection (or insertion)*

Using this technique, you store the data you want to hide in sections of a file that are ignored by the processing application. By doing this you avoid modifying those file bits that are relevant to an end-user—leaving the cover file perfectly usable. For example, you can add additional *harmless* bytes in an executable or binary file. Because those bytes don't affect the process, the end-user may not even realize that the file contains additional hidden information. However, using an insertion technique changes file size according to the amount of data hidden and therefore, if the file looks unusually large, it may arouse suspicion.

D. Generation

Unlike injection and substitution, this technique doesn't require an existing cover file—this technique generates a cover file for the sole purpose of hiding the message. The main flaw of the insertion and substitution techniques is that people can compare the stego file with any pre-existing copy of the cover file (which is supposed to be the *same* file) and discover differences between the two. You won't have that problem when using a generation approach, because the result is an *original* file, and is therefore immune to comparison tests Among the substitution techniques, a very popular methodology is the LSB (Least Significant Bit) algorithm**,** which replaces the least significant bit in some bytes of the cover file to hide a sequence of bytes containing the hidden data.

The encrypt module is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in destination.

The decrypt module is used to get the hidden information in an image file. It take the image file as an output, and give two file at destination folder, one is the same image file and another is the message file that is hidden it that.

Classification and Techniques used in Steganography

There are many approaches in classifying steganographic systems. They can be classified according to the type of covers used for secret communication. Here, we classify Steganography according to the cover modifications applied in the embedding process. The six classifications are:

***Substitution systems* –** Substitute redundant parts of a cover with a secret message.

Transform domain techniques – Embed secret information in a transform space of the signal (e.g. Frequency domain)

Spread spectrum techniques– Adopt ideas from spread spectrum communication

Statistical methods – Encode the information by modifying many statistical properties of a cover and then use of hypothesis testing in the extraction process.

Distortion techniques – Storing of information by signal distortion and then measure the deviation from the original cover in the decoding step

Cover generation method – Encode information in the way that the cover for secret communication is created.

Conclusion

This paper introduced the concept of steganography and

Steganalysis as well as the methods for carrying these out. We believe that steganography when combined with encryption provides a secure means of secret communication between two parties. Our application, with its image analysis and ranking capability is a significant improvement on current steganography tools.

Steganography can be used for hidden communication. We have explored the limits of Steganography theory and practice. We printed out the enhancement of the image Steganography system using LSB approach to provide a means of secure communication. A Stego-key has been applied to the system during embedment of the message into the cover image.

References

[1] Steganography – A Data Hiding Technique [International Journal of Computer Applications (0975 – 8887) Volume 9– No.7, November 2010]

[2] Text Steganography : Novel Approach [International Journal of Advanced Science and Technology Vol. 3, February, 2009]

[3] Sutaone, M.S., Khandare, M.V, "Image based Steganography using LSB insertion technique", IEEE WMMN, pp. 146-151, January 2008 http://www.nicetext.com/doc/isc01.pdf

[4] Ming, Chen, Z. Ru, N. Xinxin, and Y. Yixian, "Analysis of Current Steganography Tools: Classifications & Features", Information Security Centre, Beijing University of Posts & Telecommunication, Beijing, December 2006.

[5] T. Morkel, "An Overview of Image Steganography", Department of Computer Science, University of Pretoria, South Africa

[6] J. R. Smith and B.O. Comisky. Modulation and information hiding in images. In R. Anderson, editor, *Information Hiding, First International Workshop*, volume 1174 of *Lecture Notes in Computer Science*, pages 207–226. Springer-Verlag, Berlin, 1996.

[7] Mamta Juneja , Parvinder Singh Sandhu, "Designing of Robust Image Steganography

Technique Based on LSB Insertion and Encryption" 2009 International Conference on Advances in Recent Technologies in Communication and Computing.

[8] R.Chandramouli and Nasir Memon, " Analysis of LSB Based Image Steganography Techniques" , 2001 International Conference on Image Processing. October 7-10,2001. Thessaloniki, Greece, Vol. 3, pp. 1019-1022.

[9] C-C Chang, J-Y , Hsiao, C-S, " Finding Optimal least -significant-bit substitution in image hiding by dynamic programming strategy," Pattern Recognition Vol. 36,Issue 7, pp. 1583-1595, 2003.

[10] R-Z Wang, C-F, Lin, and J-C, Lin, "Image hiding by optimal LSB substitution and genetic algorithm," Pattern Recognition Vol. 34, Issue, pp. 671-683, 2001.