IJCSMS International Journal of Computer Science and Management Studies, Vol. 12, Issue 01, January 2012 7
ISSN (Online): 2231-5268
www.ijcsms.com

Implementation of IMAGE STEGANOGRAPHY Based on Random LSB

Ashish kumari[1], Shyama Sharma[2], Navdeep Bohra[3]

[1] M.Tech (Scholar), Computer Science & Engineering,
Shekhawati Engineering College, Jhunjhunu (Rajasthan)
simplyaashi@gmail.com

[2]M.Tech (Scholar), Computer Science & Engineering,
SGT Institute of Engineering & Technology, Gurgaon (Haryana)
simpysharma90@gmail.com

[3]Asist. Prof., Computer Science & Engineering,
Maharaja Surajmal Institute of Technology, Janakpuri (New Delhi)
navdeepbohra@gmail.com

ABSTRACT

Steganography is the technique of hiding a private message within a file in such a manner that third party cannot know the existence of matter or the hidden information. The purpose of Steganography is to create secrete communication between the sender and the receiver by replacing the least significant bits (LSB)of the cover image with the data bits. And in this paper we have shown that how image steganography (random and sequential LSB) works and practical understanding of what image Steganography is and how to accomplish it.

Keywords:*Cover image; Image steganography; LSB; Information hiding; Embedding; Steganalysis.*

1. INTRODUCTION

The word steganography comes from a Greek word “stegano” which means covered or hidden and the first recorded uses of steganography can be traced back to 440 B.C. to communicate in wars. From then onwards various possible permutations are done on the used applications to procure new and more ways of carrying out the information in an efficient way[4].

The idea of steganography is to keep unauthorized users away from thinking that hidden information even exists within steganographic files. Steganography is actually a method for secret communication that is about concealing the existence of a messages, where as the classical cryptography is about concealing the content of messages. It involves hiding a secret message in an appropriate file that contain irrelevant or redundant information, for example images, text, video clips, music and sounds. Steganography can be split into two main categories: [9]

1. **Statistics- aware steganography**: considers the statistical techniques that steganalysts are known to use to detect steganography.

2. **Model-based steganography**: considers the preservation of a chosen model of the cover works, rather than its statistics (components of a cover work that do not change after embedding).

In last few decades, the drastic revolution in the digital information has brought many changes in our society and our lives. Now days it is easy to access and "share" images with the internet allowing people to reach information from anywhere in the world. There has also been an increase in the number of digital images on the internet due to the fact that millions of people are taking digital photos. The excessive use of images on internet can be helpful in secure information sharing. This brings about the need for people to protect their images, their secret data or intellectual property.

Given the motivation to protect intellectual property, Steganography has been suggested as a form of data concealing in cover images copyright protection [2] and a deterrent to those wishing to obtain personal or secret data or images illegally.

In an image-based hiding system, the original image used to embed secret data is called the cover image while the resultant image, which is embedded with secret data, is referred to stego image [5].

Image steganography systems have three conflicting conditions contend with each other:

capacity, imperceptibility and robustness. Capacity refers to the amount of information that can be hidden in the cover medium, imperceptibility to an eavesdropper's inability to detect hidden information, and robustness to the amount of modification the stego medium can withstand before an adversary can destroy hidden information [1].

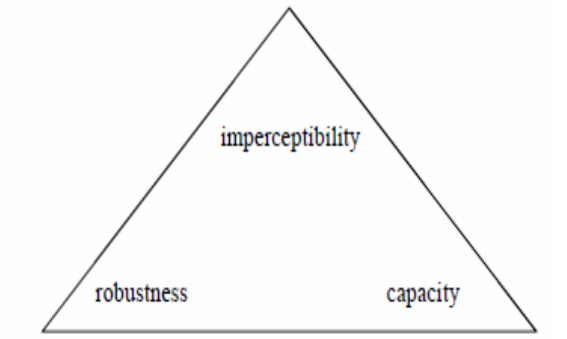

Figure 1. Three conflicting conditions of stego-image

Steganalysis is the process of detecting steganographic messages and a particular steganalysis technique is known as attack. Steganalysis is a two-stage process:
1) Classification of an image as being stego-bearing or not, and
2) Finding the location of stego-bearing pixels to extract, manipulate or sterilize the message that is applied to media such as image , audio or video clips.(stego-bearing pixels are the pixels containing the hidden message bits) [4].

2. COVER FILE (IMAGE)

Images are visual data stored in a picture frame. Steganography uses digital images which are actually an approximation of the original image (produced by camera, scanner etc) as the most common type of carrier. To produced the image, system focuses a two dimensional pattern of varying light intensity and color onto a sensor. The pattern has the co-ordinate system which can be described by a function f(x, y) and the origin is the upper left hand corner of the image. The pattern of an image can be described as an array of numbers that represent light intensities at various points. These light intensities or instances of color are called pixels.

The size of an image can be given in pixels, for example an image which is 640 x 480 pixels contains 307,200 pixels.

Each pixel is generally stored as 24-bit or 8-bit. A 24-bit pixel has a possibility of (2^{24} possible combination of colors) 16777216 colors and 8-bit pixel has (2^8 possible combination of colors)256 colors.6 The 24 bits of a 24-bit image are spread over three bytes and each byte represents red, green and blue respectively. By mixing red, green and blue light in different proportions we can obtain Colors. And image is formed by making three measurements of brightness at each pixel using the red, green and blue components. Using the RGB model the value of f(x, y) is a vector with three components corresponding to red (R), green (G) and blue (B).Each byte can have a value from 0 to 255 representing the intensity of the color. The darkest color value is 0 and the brightest is 255.

Example: A pixel could be made up of three bytes as follows: 11111111 00000000 11111111, then it shows magenta (fuchsia) color having hexadecimal value #FF00FF.

By manipulating the R, G, B values (pixel values) we can hide data in the images.

A marginal deviation in these pixel values does not alter the images as a whole but a slight shade difference occurs in the altered region that is not visible in normal conditions. Therefore the image can be serve as a cover for the information hiding (steganography). The edited image can be transmitted to the receiver along with the original image.

3. LSB BASED STEGANOGRAPHY

LSB based steganography is perhaps the most simple and straightforward approach. Here you embed the message into the least significant bit plane of the image. Since this will only affect each pixel by +l or -1, if at all it is generally assumed with good reason that the degradation caused by this embedding process would be perceptually transparent. Hence they are a number of LSB based steganography techniques available in the public domain [2].

In LSB embedding, the data is hidden in the least significant bit of each byte in the image. The size of each pixel depends on the format of the image and normally ranges from 1 byte to 3 bytes (8 bits or 24 bits). Each unique numerical pixel value corresponds to a color; thus, an 8-bit pixel is capable of displaying 256 different colors[7].The two images will look identical even if the least significant bits of the pixels in one image is changed. This is because the human eye is not sensitive enough to notice the difference in color between pixels that are different bye 1 unit.

This is why attackers do not notice anything odd or suspicious about an image if its pixel's least significant bits are modified , hence steganography applications uses LSB embedding[7].

The term Least Significant Bit (LSB) refers to the right-most bit of a binary sequence. The representation of data in binary format may only be either a 0 or a 1, often called off and on states respectively. Starting from the right most bit, the value (if on) denotes a 1. The second last bit from the right side (if on) denotes a 2, and so on up to the values double each time. If the LSB is a 1, then the total will be an odd number, and if 0, it will be an even number. However, changing the LSB value from a 0 to a 1 does not have a huge impact on the final figure; it will only affect each pixel by +1 or -1[9].

This LSB change does not have any effect on the image, change of value from a 0 to a 1 will only change the color by +1 and therefore, an altered image with slight variations in its colors will be indistinguishable from the original image by a human being, just by looking at it. The image itself is seemed unaltered after adding the least significant bits of pixels color image to store the hidden data [8].

Example: Suppose "hiding"' the alphabet N across the following eight bytes of a carrier file (the least significant bits are bold n underlined). A 'N' is represented in the American Standard Code for Information Interchange (ASCII) through binary string 01001110 and in decimal by 78. These eight binary bits can be written to the **LSB** of each of the eight carrier bytes as follows:

1001010**0** 0000110**1** 1100100**0** 1001011**0**

0000111**1** 1100101**1** 1001111**1** 0001000**0**

Only half of the least significant bits were actually changed (shown above in bold). This makes some sense when one set of 0s and 1s are being added with another set of 0s and 1s [8].

Two different embedding schemes of LSB: **sequential and randomized [9]**

- **Sequential embedding** the algorithm starts at the first pixel of the cover image (0,0) and embeds the bits of the message data in order until there is nothing left to embed.
- **Randomized embedding** scatters the locations of the values that will be modified to contain the bits of the message data. The main purpose for randomizing the approach is to make things a little trickier for the steganalysts that are looking to determine whether the image is a stegogramme or not.

4. TECHNIQUES

The three basic techniques used for Steganography are: [10]

- **Injection:** information hiding in parts of a file that are avoided by the processing application. So, ignore modifying those file bits that are relevant to an end-user leaving the cover file perfectly usable.
- **Substitution:** change of the least significant bits of information that specify the meaningful content of the main file with new addition of data in a method that causes the less amount of distortion.
- **Generation:** Unlike injection and substitution, this does not require an existing cover file but it generates a cover file for the sole purpose of substitute the least significant bits.

The steganography includes the write the text message, encode the text message, select the media for hiding the data and transmitted to recipient. At last receiver decode the received data and recover the secrete data from the image.

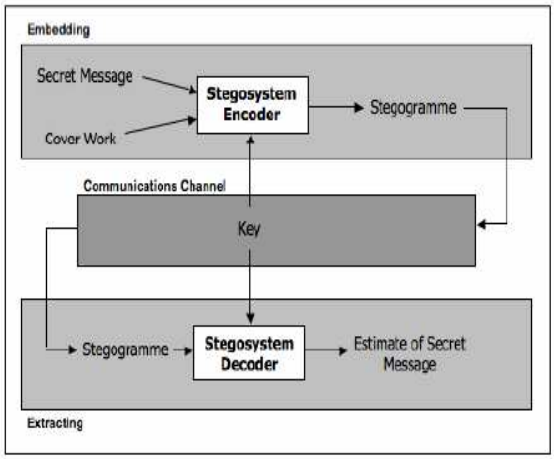

Figure2.Process of Steganography [9]

5. ANALYSIS OF LSB

Secret messages can be embedded in an image either by using sequential or random LSB replacement. It is more convenient to implement sequential LSB replacement but it has a much serious security problem in that there is an obvious statistical difference between the modified and unmodified part of the stego image. In random LSB the secret data are randomly scatter among the image data and hence improve the steganographic security. Here the sender and receiver share a secret key to generate pseudorandom number to identify where, and in what order the hidden message is laid out. This method incorporates some cryptography in that diffusion is applied to the secret message. It is much harder for the attacker to figure out the secret message because this involves changing redundant bits within the cover object and this randomness makes the embedded message seem more like noise statistically than in the sequential method [8].

6. ALGORITHM & IMPLEMENTATION DETAILS

There are some image based steganography algorithms ,Advanced Encryption Standard (AES), Data Encryption Standard (DES), Message Digest 5 (MD5), International Data Encryption Algorithm (IDEA), Bit Stream Ciphers (BSC), Secure Hash Algorithm (SHA) [4] etc. are used [12].

The use of DES, AES or IDEA algorithm is beneficial when there is a less amount of data. These algorithms involve concentrated computation and super fast processing machines and hence they are not good for massive data sets but for the little data computation these are best. But algorithms like MD5, SHA based on cryptographic hash function (uses a hash key of 16 byte)[12].

IMPLEMENTATION:

It includes the encryption and decryption steps and the media through which we send the data from sender to receiver**.**

Sender side (Encryption):

- Open the main window**.**
- Enter the data (carrier file) which we want to hide.
- Now enter the key value and if the key is not entered a pop up box is generated asking the user to enter a key.
- Open the image file (cover file) in which we want to hide the data. When an image file is not chosen a pop up box is generated asking the user to choose an image file first.
- When browse button is pressed a file chooser window is opened that allows user to choose only gif images.
- When encryption i.e. scrambling the data and hiding it behind the image is done a pop indicating that encryption has been done is generated, shown in figure 3.
- A valid IP address is entered to send the data from one system to another, shown in figure 4.
- A pop up tells the sender that the complete file has been sent.

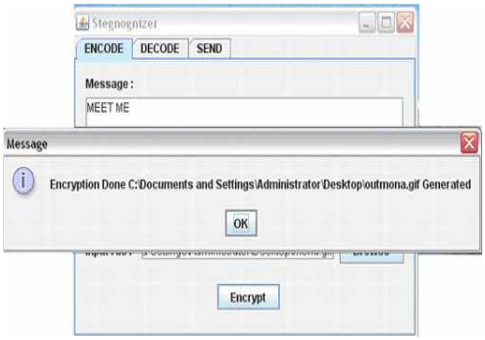

Figure 3 Encryption is done after hiding the data in image.

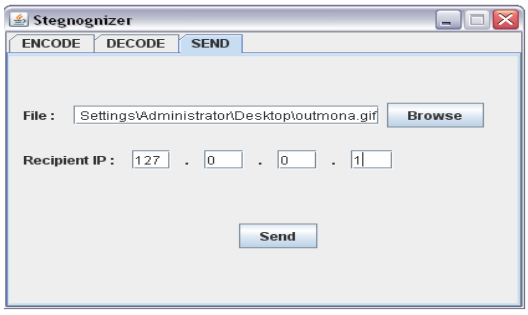

Figure 4 Sending encrypted data on the receiver's IP address.

Receiver side (Decryption):

- A pop up on the receivers side tells the receiver that a file has been received and asks if he wants to save the file.

- A file chooser is opened on the receiver's side to let the user choose the path to save the file at desired location. A pop up tells the receiver that file has been saved to his chosen path.
- When a key of less or more than 8 bytes is entered a pop up is generated asking the user to enter a valid 8 byte key.
- After the decoding and decryption process is completed a pop tells the receiver that message has been decoded.
- The decrypted message can be seen in the text area.

Figure 5 Decryption of the message

For example (Random LSB) hide a character '**N**' (carrier file) into cover file.
Bit format of alphabet 'N' - ASCII 78

Binary 0 1 0 0 1 1 1 0
Key 0 7 3 0 6 3 3 1

Each bit of the given alphabet '**N**' is hidden at the key position of byte of main file (cover file). So for hiding **8**-bits of alphabet '**N**' requires **8**-bytes of cover file. As shown in the example below:
Cover file format is:

0 7 3
00001001 10101111 00111011
0 6 3
11010101 00001111 01101011
3 1
00010101 01101010

After embedding the LSB data the encrypted file is –

0 7 3
0000100*0* *1*0101111 0011*0*011
0 6 3
1101010*0* 0*1*001111 0110*1*011
3 1
0001*1*101 01101*0*0

7. CONCLUSION

A steganography method results in a stego image which contains the hidden message and due to the various properties of image (because of its different the contents) this message is almost undetectable by the steganlysis. Steganographic process involves:-
cover medium + Some hidden data + A stego key = A stego medium
The simplest method of steganography is LSB in which we can embed the data sequentially or randomly .The key value is used in random LSB on both sides, during encryption and decryption. Steganography violates the originality of cover image for hiding the data only on the place of key bits but human eyes are not capable of identifying the difference between real and the stego image. Random LSB technique of steganography is more secure than the sequential LSB steganography because in random LSB steganography we place the message bits randomly on the given key value.

Steganography is one of the most commonly used technique for secrete communication but still there are some issues that need to be resolved. Different techniques are there for secrete communications with their own advantages and disadvantages. And hence improvements and changes are needed to be made regularly and newer versions are to be released.

8. REFERENCES

[1] Niels Provos, Peter Honeyman,"Hide and Seek: Introduction to Steganography ",Security & Privacy, IEEE (32 – 44) ,2003.

[2] R. Chandramouli, N. Memon, "Analysis of LSB based image steganography techniques," pp. 1019-1022. , Image Processing, 2001.

[3] Adel Almohammad, G. Ghinea, "Image Steganography and Chrominance Components," 996-1001,10th IEEE International Conference on Computer and Information Technology, 2010.

[4] Subba Rao, Y.V.; Brahmananda Rao, S.S.; Rukma Rekha, N.; "Secure Image Steganography based on Randomized Sequence of Cipher Bits", (332 - 335)

Information Technology: New Generations (ITNG), 2011 Eighth International Conference.

[5] Jinsuk Baek; Kim, Cheonshik; Fisher, Paul S.; Hongyang Chao; "(N, 1) secret sharing approach based onsteganography wit h gray digital images" Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference ,June 2010.

[6] Hong-Juan Zhang; Hong-Jun Tang; "A Novel Image Steganography Algorithm against Statistical Analysis", Page(s): 3884 – 3888, IEEE international conference, 2007.

[7] Juneja, M.; Sandhu, P.S.; "Designing of Robust Image SteganographyTechnique Based on LSB Inse rtion and Encryption", (302 – 305), Advances in Recent Technologies in Communication and Computing, 2009. ART Com'09. International Conference, 2009 .

[8]Sutaone, M.S. Khandare, M.V.; "Image based steganography using LSB insertiontechnique ", (146 – 151), IET International Conference on Wireless, Mobile and Multimedia Networks, 2008.

[9] Philip Bateman, "Image Steganography and Steganalysis", 4th August 2008.

[10]B.mehboob and Rashid Aziz Faruqui "A Steganography Implementation", 2008 IEEE.

[11] An Evaluation of Image Based Steganography Methods Kevin Curran, Karen Bailey, International Journal of Digital Evidence , 2003.

12] Imran Sarwar Bajwa

"A Hash-Based Approach for Colour Image Steganography", IEEE International Conference on Computer Networks and Information Technology (ICCNIT,2011).