The Process of Encoding and Decoding of Image Steganography using LSB Algorithm

Ravinder Reddy Ch[1] Roja Ramani A[2]
[1] *Department of Master of Computer Applications, Teegala Krishna Reddy Engineering College, Medbowli, Meerpet, Hyderabad 500 097*
churukantireddy@gmal.com

[2]*Department of Master of Computer Applications, Teegala Krishna Reddy Engineering College, Medbowli, Meerpet, Hyderabad 500 097*
rojadabbulu@gmail.com

Abstract: Steganography is a technique that allows one to hide binary data within an image while adding few noticeable changes. Steganography is the dark cousin of cryptography, the use of codes. While cryptography provides privacy, steganography is intended to provide secrecy. Privacy is what you need when you use your credit card on the Internet -- you don't want your number revealed to the public. For this, you use cryptography, and send a coded pile of gibberish that only the web site can decipher. Though your code may be unbreakable, any hacker can look and see you've sent a message. For true secrecy, you don't want anyone to know you're sending a message at all.

In the present world, the data transfers using internet is rapidly growing because it is so easier as well as faster to transfer the data to destination. So, many individuals and business people use to transfer business documents, important information using internet. Security is an important issue while transferring the data using internet because any unauthorized individual can hack the data and make it useless or obtain information un- intended to him. The main intention of the paper is to analyze the various steganography algorithms and stenographic application such that it provides good security. The proposed approach provides higher security and can protect the message from stego attacks. The image resolution doesn't change much and is negligible when we embed the message into the image and the image is protected with the personal password. So, it is not possible to damage the data by unauthorized personnel.

Keywords:
Encryption: is used to hide information into the image-Algorithm
Decryption: is used to get the hidden information in an image file-Algorithm
Least Significant Bit Substitution Techniques:
- ***Image definition***
- ***Image Compression***
- ***Image Processing***

Transforming values to frequencies

INTRODUCTION

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 exists 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 project report intends to give an overview of image steganography, its uses and techniques. It also attempts to identify the requirements of a good steganography algorithm and briefly reflects onwhich steganographic techniques are more suitable for which applications.

Graphical Representation

The graphical representation of Steganography system is as follows:

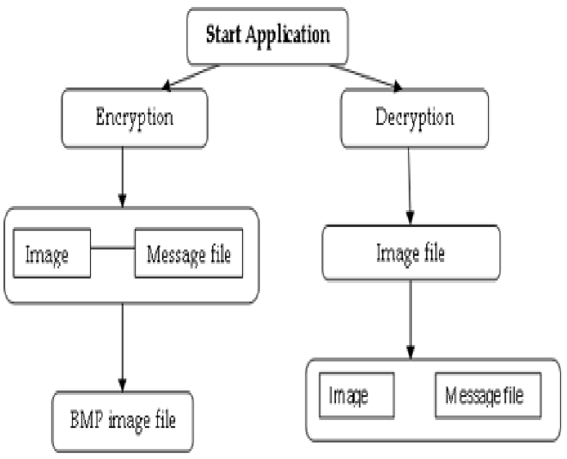

Fig-1

The two methods are – Encrypt and Decrypt.
In encryption the secret information is hiding in with any type of image file.Decryption is getting the secret information from image file.

ENCRYPTION PROCESS:

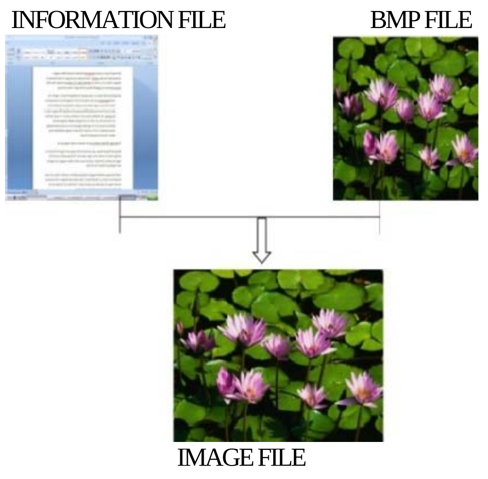

Fig-2

DECRYPTION PROCESS:

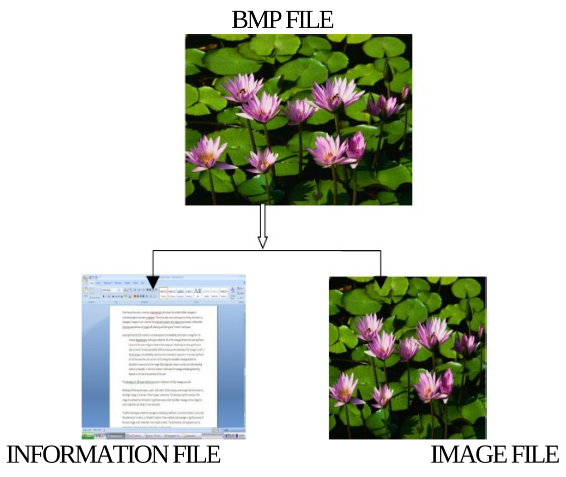

Fig-3

To send a message, a source text, an image in which the text should be embedded, and a key are needed. The key is used to aid in encryption and to decide where the information should be hidden in the image. Either another image or a short text can be used as a key. To receive a message, a source image containing the information and the corresponding key are both required. The result will appear in the text tab after decoding. Images should contain as many colors as possible. It is also best if many of the colors are similar. At least 4 colors are required.

Microsoft .Net framework prepares a huge amount of tool and options for programmers that they simples programming. One of .Net tools for pictures and images is auto-converting most types of pictures to BMP format. I used this tool in this software called "Steganography" that is written in C#.Net language and you can use this software to hide your information in any type of pictures without any converting its format to BMP (software converts inside it).

LSB algorithm:

The algorithm used for Encryption and Decryption in this application provides using several layers lieu of using only LSB layer of image. Writing data starts from last layer (8st or LSB layer); because significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.

The encryption 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 decryption 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 in that.

LSB (Least Significant Bit) substitution is the process of adjusting the least significant bit pixels of the carrier image. It is a simple approach for embedding message into the image. The Least Significant Bit insertion varies according to number of bits in an image. For an 8 bit image, the least significant bit i.e., the 8th bit of each byte of the image is changed to the bit of secret message. For 24 bit image, the colors of each component like RGB (red, green and blue) are changed. LSB is effective in using BMP images since the compression in BMP is lossless. But for hiding the secret message inside an image of BMP file using LSB algorithm it requires a large image which is used as a cover. LSB substitution is also possible for GIF formats, but the problem with the GIF image is whenever the least significant bit is changed the whole colour palette will be changed. The problem can be avoided by only using the gray scale GIF images since the gray scale image contains 256 shades and the changes will be done gradually so that it will be very hard to detect.JPEG, the direct substitution of steganographic techniques is not possible since it will use lossy compression. So it uses LSB substitution for embedding the data into images. There are many approaches available for hiding the data within an image: one of the simple least significant bit submission approaches is 'Optimum Pixel Adjustment Procedure'.

The simple steps for **OPA** explain the procedure of hiding the sample text in an image.

Step1: A few least significant bits (LSB) are substituted with in data to be hidden.

Step2: The pixels are arranged in a manner of placing the hidden bits before the pixel of each cover image to minimize the errors.

Step3: Let n LSBs be substituted in each pixel.

Step4: Let d= decimal value of the pixel after the substitution.d1 = decimal value of last n bits of the pixel.d2 = decimal value of n bits hidden in that pixel.

Step5: If (d1~d2)<=(2^n)/2then no adjustment is made in that pixel. Else

Step6: If(d1<d2)d = d – 2^n.If(d1>d2)d = d + 2^n.This 'd' is converted to binary and written back to pixel. This method of substitution is simple and easy to retrieve the data and the image quality better so that it provides good security.

The encoder algorithm is as given below:

```
1: for i = 1, ..., len (msg) do
2: p = LSB(pixel of the image)
3: if p != message bit then
4: pixel of the image = message bit
5: end if
6: end for
```

The encoding process shows that the entire algorithm can be implemented by writing just a few lines of code. The algorithm works by taking the first pixel of the image and obtaining its LSB value (as per line 2 of the Algorithm). This is typically achieved by calculating the modulus 2 of the pixel value. This will return a 0 if then number is even, and a 1 if the number is odd, which effectively tells us the LSB value. We then compare this value with the message bit that we are trying to embed. If they are already the same, then we do nothing, but if they are different then were place the pixel value with the message bit. This process continues whilst there are still values in the message that need to be encoded The decoder algorithm is:1: for i = 1, ..., len(image string) do2:message string = LSB (pixel string of the image)3: end for The decoding phase is even simpler. As the encoder replaced the LSBs of the pixel values in c in sequence, we already know the order that should be used to retrieve the data. Therefore all we need to do is calculate the modulus 2 of all the pixel values in the stegogramme, and we are able to reconstruct m as m0 .The above Algorithms how the pseudo code of the decoding process.

Note that this time we run the loop for length of message instead of length of string. This is because the decoding process is completely separate from the encoding process and therefore has no means of knowing the length of the message. If a key were used, it would probably reveal this information, but instead we simply retrieve the LSB value of every pixel. When we convert this to ASCII, the message will be readable up to the point that the message was encoded, and will then appear as gibberish when we are reading the LSBs of the image data.

Least Significant Bit Substitution Techniques

One of the earliest stego-systems to surface were those referred to as Least Significant Bit Substitution techniques, so called because of how the message data m is embedded within a cover image c. In computer science, the term Least Significant Bit (LSB) refers to the smallest (right-most) bit of a binary sequence. The structure of binary is such that each integer may only be either a 0 or a 1, often thought of as off and on respectively.

Starting from the right, the value (if on) denotes a 1. The value to its left (if on) denotes a 2, and so on where the values double each time. Now let us consider the following 8-bit binary sequence:

1 0 1 1 0 0 1 1

Summing all the values equal to 1 yields a result of 179. The right-most value (denoted in bold text) is the LSB of this sequence. This value essentially determines whether the total sum is odd or even. 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 ever change by +1 at most. If we now think of each 8-bit binary sequence as a means of expressing the colour of a pixel for an image, it should be clear to see that changing the LSB value from a 0 to a 1 will only change the colour by +1 - a change that is unlikely to be noticed with the naked eye. In fact, the LSBs of each pixel value could potentially be modified, and the changes would still not be visible. This highlights a huge amount of redundancy in the image data, and means that we can effectively substitute the LSBs of the image data, with each bit of the message data until the entire message has been embedded. This is what is meant by Least Significant Bit Substitution. Finally, when we talk of Least Significant Bit Substitution algorithms, we should mention that this encompasses two different embedding schemes: sequential and randomised. Sequential embedding often means that the algorithm starts at the first pixel of the cover image c0;0 and embeds the bits of the message data in order until there is nothing left to embed. Randomised embedding however scatters the locations of the values that will be modified to contain the bits of the message data. The main reason 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.

Image definition

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image [14]. This numeric representation forms a grid and the individual points are referred to as pixels.Most images on the Internet consists of a rectangular map of the image's pixels (represented as bits) where each pixel is located and its colour [15]. These pixels are displayed horizontally row by row. The number of bits in a colour scheme, called the bit depth, refers to the number of bits used for each pixel [16]. The smallest bit depth in current colour schemes is 8, meaning that there are 8 bits used to describe the colour of each pixel [16]. Monochrome and greyscale images use 8 bits for each pixel and are able to display 256 different colours or shades of grey. Digital colour images are typically stored in 24-bit files and use the RGB colour model, also known as true colour [16]. All colour variations for the pixels of a 24-bit image are derived from three primary colours: red, green and blue, and each primary colour is represented by 8 bits [14]. Thus in one given pixel, there can be 256 different quantities of red, green and blue, adding up to more than 16-million combinations, resulting in more than 16-million colours [16]. Not surprisingly the larger amount of colors that can be displayed, the larger the file size [15]. 3.2 Image Compression when working with larger images of greater bit depth, the images tend to become too large to transmit over a standard Internet connection. In order to display an image in a reasonable amount of time, techniques must be incorporated to reduce the image's file size. These techniques make use of mathematical formulas to analyses and condense image data, resulting in smaller file sizes. This process is called compression [15]. In images there are two types of compression: lossy and lossless [1]. Both methods save storage space, but the procedures that they implement differ. Lossy compression creates smaller files by discarding excess image data from the original image. It removes details that are too small for the human eye to differentiate [15], resulting in close approximations of the original image, although not an exact duplicate. An example of an image format that uses this compression technique is JPEG (Joint Photographic Experts Group) [14]. Lossless compression, on the other hand, never removes any information from the original image, but instead represents data in mathematical formulas [15]. The original image's integrity is maintained and the decompressed image output is bit-by-bit identical to the original image input [1]. The most popular image formats that use lossless compression is GIF (Graphical Interchange Format) and 8-bit BMP (a Microsoft Windows bitmap file) [14].

Compression plays a very important role in choosing which steganographic algorithm to use. Lossy compression techniques result in smaller image file sizes, but it increases the possibility that the embedded message may be partly lost due to the fact that excess image data will be removed [7]. Lossless compression though, keeps the original digital image intact without the chance of lost, although is does not compress the image to such a small file size [14].

Image and Transform Domain

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain [2]. Image – also known as spatial – domain techniques embed messages in the intensity of the pixels directly, while for transform – also known as frequency – domain, images are first transformed and then the message is embedded in the image [20].Image domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterised as "simple systems" [17]. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format [18]. Steganography in the transform domain involves the manipulation of algorithms and image transforms [17]. These methods hide messages in more significant areas of the cover image, making it more robust [4]. Many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression [18]. In the next sections steganographic algorithms will be explained in categories according to image file formats and the domain in which they are performed.

Image Processing: JPEG Compression

JPEG compression is a commonly used method for reducing the file size of an image, without reducing the aesthetic qualities enough to become noticeable by the naked eye.

Image Processing: JPEG Compression

Broadly speaking, it extracts all the information from an image that the human eye is not perceptible to - and would therefore not miss - should it not be there.

The compression of JPEG images contains several processes:

1. Converting pixel values to YCbCr
2. Down sampling the chrominance values
3. Transforming values to frequencies
4. Quantisation
5. Zig-Zag ordering
6. Lossless Compression

Transforming values to frequencies

The Discrete Cosine Transform (DCT) is used for JPEG images to transform them into frequencies. DCT is a mathematical transform (typically a cosine function) that converts the pixels by seemingly 'spreading' the location of the pixel values over part of the image. It does this by grouping the pixels into 8 x 8 blocks and transforming them from 64 values into 64 frequencies (DCT coefficients. By modifying just a single DCT coefficient, the entire 64 pixels in that block will be affected.

REFERENCES

[1] T. Morkel, J.H.P. Eloff, M.S. Olivier AN OVERVIEW OF IMAGE STEGANOGRAPHY, ,Information and Computer Security Architecture (ICSA) Research Group,Department of Computer Science, University of Pretoria, 0002, Pretoria, South Africa

[2] Moerland, T., "Steganography and Steganalysis", *Leiden Institute of Advanced Computing Science*,www.liacs.nl/home/ tmoerl/privtech.pdf

[3] Silman, J., "Steganography and Steganalysis: An Overview", *SANS Institute*, 2001

[4] Jamil, T., "Steganography: The art of hiding information is plain sight", *IEEE Potentials*, 18:01, 1999

[5] Wang, H & Wang, S, "Cyber warfare: Steganography vs. Steganalysis", *Communications of the ACM*,47:10, October 2004

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

[7] Marvel, L.M., Boncelet Jr., C.G. & Retter, C., "Spread Spectrum Steganography", *IEEE Transactions on image processing*, *8:08*, 1999

[8] Dunbar, B., "Steganographic techniques and their use in an Open-Systems environment", *SANS Institute*, January 2002

[9] Artz, D., "Digital Steganography: Hiding Data within Data", *IEEE Internet Computing Journal*, June 2001

[10] Simmons, G., "The prisoners problem and the subliminal channel", *CRYPTO*, 1983

[11] Chandramouli, R., Kharrazi, M. & Memon, N., "Image steganography and steganalysis: Concepts and Practice", *Proceedings of the 2nd International Workshop on Digital Watermarking*, October 2003

[12] Currie, D.L. & Irvine, C.E., "Surmounting the effects of lossy compression on Steganography", *19th National Information Systems Security Conference*, 1996

[13] Handel, T. & Sandford, M., "Hiding data in the OSI network model", *Proceedings of the 1st International Workshop on Information Hiding*, June 1996

[14] Ahsan, K. & Kundur, D., "Practical Data hiding in TCP/IP", *Proceedings of the Workshop on Multimedia Security at ACM Multimedia*, 2002

[15] Johnson, N.F. & Jajodia, S., "Exploring Steganography: Seeing the Unseen", *Computer Journal*,February 1998

[16] "Reference guide: Graphics Technical Options and Decisions",http://www.devx.com/projectcool/Article/19997

[17] Owens, M., "A discussion of covert channels and steganography", *SANS Institute*, 2002

[18] Johnson, N.F. & Jajodia, S., "Steganalysis of Images Created Using Current Steganography Software",*Proceedings of the 2nd Information Hiding Workshop*, April 1998
[19] Venkatraman, S., Abraham, A. & Paprzycki, M., "Significance of Steganography on Data Security",*Proceedings of the International Conference on Information Technology: Coding and Computing*,2004
[20] Krenn, R., "Steganography and Steganalysis", http://www.krenn.nl/univ/cry/steg/article.pdf
[21] Lee, Y.K. & Chen, L.H., "High capacity image steganographic model", *Visual Image Signal Processing*, 147:03, June 2000
[22] Provos, N. & Honeyman, P., "Hide and Seek: An introduction to steganography", *IEEE Security and Privacy Journal*, 2003
[23] Bender, W., Gruhl, D., Morimoto, N. & Lu, A., "Techniques for data hiding", *IBM Systems Journal*, Vol 35, 1996
[24] Petitcolas, F.A.P., Anderson, R.J. & Kuhn, M.G., "Information Hiding – A survey", *Proceedings of the IEEE*, 87:07, July 1999

AUTHORS:

ADAPA.ROJA RAMANI ;Asst.Prof;Dept of M.C.A, with -13Years Teaching Experience.

CHURUKANTI.RAVINDER REDDY;H.O.D; Asst.Prof;Dept of M.C.A, with -13Years OF Teaching&INDUSTRIAL Experience