International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991
Vol. 3, No. 4
November 2014

www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 3, No. 4, November 2014

Research Paper

A SECURE IMAGE STEGANOGRAPHY USING LEAST SIGNIFICANT BIT TECHNIQUE

Amit[1]* and Jyoti Pruthi[1]

Corresponding Author: **Amit ✉ amit.ahlawat89@gmail.com*

Steganography is a science of hiding messages into multimedia documents. A message can be hidden in a document only if the content of a document has high redundancy. Although the embedded message changes the characteristics and nature of the document, it is required that these changes are difficult to be identified by an unsuspecting user. On the other hand, steg analysis develops theories, methods and techniques that can be used to detect hidden messages in multimedia documents. 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.

Keywords: Component, Formatting, Style, Styling, Insert

INTRODUCTION

The word steganography comes from the Greek "Seganos [1]", which mean covered or secret and – "graphy" mean writing or drawing. Therefore, steganography mean, literally, covered writing. It is the art and science of hiding information such its presence cannot be detected and a communication is happening. A secrete information is encoding in a manner such that the very existence of the information is concealed.

In general, security denotes "the quality or state of being secure to be free from danger". Security is classified into different layers depending on the type of content intended to be secured:

Physical security: Defines the required issues that are needed to protect the physical data or objects from unauthorized intrusion.

Personal security: It is defined as the security of the individuals who are officially authorized to access information about the company and its operations Operational security: It mainly relies on the protection of the information of a particular operation of the chain of activities.

Network security: The network security is

[1] Department of computer Science & Engineering, CBS Group of Institutions Fatehpuri, Jhajjar, Mahrashi Dayanand Uiversity Rohtak, Haryana.

responsible for safeguarding the information regarding the networking components, connections and contents.

Information security: Information security is the protection of information and the systems and hardware that use, store, and transmit that information. It can be defined as measures adopted to prevent the unauthorized use or modification of use of data or capabilities. Now that we are aware of the various types of security vulnerabilities, the main task of our paper is to address these problems by some suitable method. A technique which enables to have a secret communication in modern technology using public channel is known as steganography. This paper deals with constructing and implementing new algorithm based on hiding a large amount of payloads (image, audio, text) file into color image. We have been used adaptive image filtering and adaptive image segmentation with bits replacement on the appropriate pixels. The information has to be hidden in the LSB of the cover image, so there is no much chance present in the embedded image. High security layers have been proposed through three layers to make it difficult to break through the encryption of the input data and confuse steganalysis too.

STEGANOGRAPHY IN IMAGES

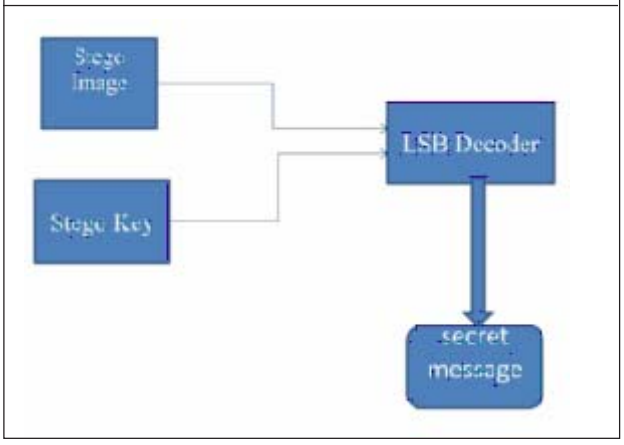
In this paper we deal with data encoding in still digital images. In essence, image steganography is about exploiting the limited powers of the Human Visual System (HVS). Within reason, any plain text, cipher text, other images, or anything that can be embedded in a bit stream can be hidden in an image. Image steganography has come quite far in recent years with the development of fast, powerful graphical computers, and steganographic software is now readily available over the Internet for everyday users. To a computer, an image is an array of numbers that represent light intensities at various points, or pixels. These pixels make up the image's raster data. An image size of 640 by 480 pixels, utilizing 256 colors (8 bits per pixel) is fairly common. Such an image would contain around 300 kilobits of data. Digital images are typically stored in either 24-bit or 8-bit per pixel files. 24-bit images are sometimes known as true color images. Obviously, a 24-bit image provides more space for hiding information; however, 24-bit images are generally large and not that common. A 24-bit image 1024 pixels wide by 768 pixels high would have a size in excess of 2 megabytes. As such large files would attract attention were they to be transmitted across a network or the Internet, image compression is desirable. However, compression brings with it other problems, as will explain shortly.

Alternatively, 8-bit color images can be used to hide information. In 8-bit color images (such as GIF files), each pixel is represented as a single byte. Each pixel merely points to a color index table, or palette, with 256 possible colors. If using an 8-bit image as the cover-image, many steganography experts recommend using images featuring 256 shades of grey as the palette, for reasons that will become apparent. Grey-scale images are preferred because the shades change very gradually between palette entries. This increases the image's ability to hide information. When dealing with 8-bit images, the steganographer will need to consider the image as well as the palette. Obviously, an image with large areas of solid color is a poor choice, as variances created by embedded data might be noticeable. Once a suitable cover image has been selected, an image encoding technique needs to be chosen.

LSB INSERTION

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 loss less. 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 color 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.

Figure 1: LSB Extraction Mechanism



Algorithm

Step 1: A few Least Significant Bits (LSB) are substituted with in data to be hidden.

Step 2: The pixels are arranged in a manner of placing the hidden bits before the pixel of each cover image to minimize the errors.

Step 3: Let n LSBs be substituted in each pixel.

Step 4: 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.

Step 5: If (d1~d2)<=(2^n)/2 then no adjustment is made in that pixel.

Else

Step 6: If (d1<d2)

d = d – 2^n.

If (d1>d2)

d = d + 2^n.

Jsteg algorithm is one of the stenographic techniques for embedding data into JPEG images. The hiding process will be done by replacing Least Significant Bits (LSB). Jsteg algorithm replaces LSBs of quantized Discrete Courier Transform (DCT) coefficients. In fact, the Jsteg algorithm only differs from the Hide & Seek algorithm because it embeds the message data within the LSBs of the DCT coefficients of c, rather than its pixel values. Before the embedding process begins, the image is converted to the DCT domain in 8 x 8 blocks such that the value of c is witch from pixel values to DCT coefficients.

PROPOSED WORK

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

(00101101 00011100 11011100)

(10100110 11000100 00001100)

(11010010 10101101 01100011)

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:

(00101101 00011101 11011100)

(10100110 11000101 00001100)

(11010010 10101100 01100011)

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined 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 [19]. 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. In the above example, consecutive bytes of the image data – from the first byte to the end of the message – are used to embed the information. This approach is very easy to detect. A slightly more secure system is for the sender and receiver to share a secret key that specifies only certain pixels to be changed. Should an Adversary suspect that LSB steganography has been used, he has no way of knowing which pixels to target without the secret key.

The encoding algorithm

1: convert image c to DCT domain d in 8 x 8 blocks

2: for i = 1, ..., l(m) do

3:p == DCT(di)

4:while p = DC or p = 0 or p = 1 do

5:p = next DCT coefficient from d

6:end while

7:pi == ci mod 2 + mi

8:ci == pi

9: end for

10: convert each 8 x 8 block back to spatial domain.

The decoder algorithm

1: convert image s to DCT domain d in 8 x 8 blocks

2: for i = 1, ..., l(s) do

3: p == DCT(di)

4: while p = DC or p = 0 or p = 1 do

5: p = next DCT coefficient from d

6: end while

7: mi == di mod 2

8: end for

This work provides better security when transmitting or transferring the data or messages from one end to another. The main objective of the paper is to hide the message or a secret data into an image which further act as a carrier of secret data and to transmit to the destination securely without any modification. If there are any perceivable changes when we are inserting or embedding the information into the image or if any distortions occur in the image or on its resolution there may be a chance for an unauthorized person to modify the data.

PERFORMANCE ANALYSIS

As a performance measure for image distortion due to hiding of message, the well known Peak-Signal to Noise Ratio (PSNR). which is categorized under difference distortion metrics, can be applied to stego images. It is defined as:

PSNR = 10log (Cmax)2/MSE.

MSE = mean"square"error, which is given as:

MSE = 1/MN ((S-C)2).

Cmax = 255.

where M and N are the dimensions of the image, S is the resultant stego-image, and C is the cover image.

PSNR values below 30 dB indicate low quality (i.e., distortion caused by embedding is high). A high-quality stego image should strive for a PSNR of 40 dB, or higher.

We consider gray scale/RGB image as cover image as shown in Figures 3 and 4 and text file/ image as secret message for both the Techniques and then produced stego image.

Figure 2: LSB Technique

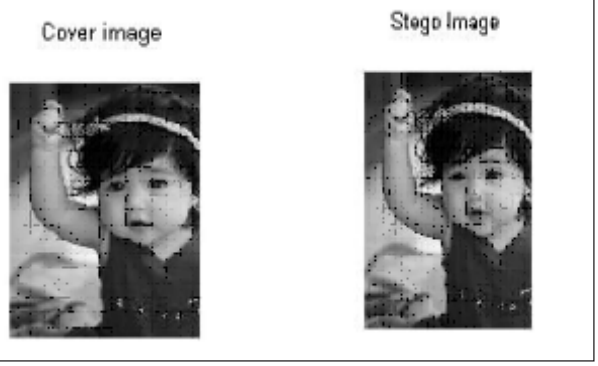
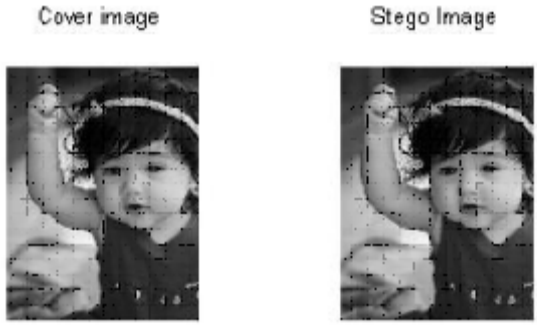


Figure 3: Pseudo-Random Encoding



Tables 1 and 2 are the result of RBG image with text file.

Table 1: PSNR of Pseudo-Random Encoding

SL.No	Cover Image	Secret Message	Stego-Image	SNR(dB)	MSE	PSNR(dB)
1	Gray image	Text message	Gray image	59.6374	0.0449	61.6065
2	RBG image	Text message	sisbr	61.3787	0.0111	67.6835
3	RBG image	Image	Images	53.9847	0.0911	58.5346

Table 2: PSNR of Least Significant Encoding

SL.No	Cover Image	Secret Message	Stego-Image	SNR(dB)	MSE	PSNR(dB)
1	Gray image	Text message	Gray image	59.5043	0.0463	61.4733
2	RBG image	Text message	sisbr	61.3649	0.021	67.6697
3	RBG image	Image	Hydrang	53.9812	0.0912	58.5311

CONCLUSION

Steganography is an effective way to hide sensitive information. In this paper we have used the LSB Technique and Pseudo-Random Encoding Technique on images to obtain secure stego-image. Our results indicate that the LSB insertion using random key is better than simple LSB insertion in case of loss less compression. 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 key. So, it is not possible to damage the data by unauthorized personnel. The algorithm is usage for both 8 bit and 24 bit image of the same size of cover and secret image, so it is easy to be implementing in both grayscale and color image. This paper focuses on the approach like increasing the security of the message and increasing PSNR and reducing the distortion rate.

REFERENCES

1. Anderson R and Petitcolas F (1998), "On the limits of steganography", *IEEE Journal of Selected Areas in Communications,* Vol. 16, No. 4.
2. Jessica Fridrich, Miroslav Goljan and Rui Du (2001), "Reliable Detection of LSB Steganography in Color and Grayscale Images", Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges, pp. 27-30.
3. Chi-Kwong Chan and L M Cheng (2004), "Hiding Data in Images by Simple LSB Substitution", Pattern Recognition, Vol. 37, No. 3, pp. 469–474.
4. Johnson N F and Jajodia S (1998), "Exploring Steganography: Seeing the Unseen", *Computer Journal*.
5. Morkel T, Eloff J H P and Olivier M S (2005), *An overview of image steganography*, In: Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), South Africa.
6. Niels Provos and Peter Honeyman (2003), "Hide and Seek: An Introduction to Steganography," *IEEE computer society.*
7. Raja K B, Venugopal K R and Patnaik L M (2004), "A Secure Stegonographic Algorithm using LSB, DCT and Image Compression on Raw Images" Technical Report, Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, December.
8. Ran-Zan Wang, Chi-Fang Lin and Ja-Chen Lin, "Hiding data in images by optimal moderately significant-bit replacement", *IEE Electron. Lett.*
9. Samir K Bandyopadhyay, Debnath Bhattacharyya, Debashis Ganguly, Swarnendu Mukherjee and Poulami Das, "A Tutorial Review on Steganography", by Heritage Institute of Technology.

International Journal of Engineering Research and Science & Technology
Hyderabad, INDIA. Ph: +91-09441351700, 09059645577
E-mail: editorijlerst@gmail.com or editor@ijerst.com
Website: www.ijerst.com