International Journal of Computer Applications Technology and Research
Volume 2– Issue 5, 587 - 590, 2013, ISSN: 2319–8656

RELIABLE DATA TRANSMISSION OVER WIRELESS NETWORK USING IMAGE STEGANOGRAPHY

Roja Ramani.A
Department of Computer Science and Engineering,
TKR College of Engineering and Technology
Hyderabad, A.P-500 097, India

P.V.S. Srinivas
Department of Computer Science and Engineering,
TKR College of Engineering and Technology
Hyderabad, A.P-500 097, India

ABSTRACT Image steganography is the science of hiding data inside cover images for security. Images have a lot of visual redundancy in the sense that our eyes do not usually care about subtle changes in color in an image region. One can use this redundancy to hide text, audio or even image data inside cover images without making significant changes to the visual perception. Image steganography is becoming popular on the internet these days since a steganography image, which just looks like any other image, attracts a lot less attention than an encrypted text and a secure channel. Steganography is the science of hiding messages in such a way that no one apart from the sender and the intended recipient, suspects the existence of the message. There are multiple techniques in order to embed data in an image, however some techniques are better at being undetected then others. These techniques depend on three different aspects: capacity, security, and robustness. Capacity refers to the amount of information that can be hidden in the cover medium. Security to an eavesdropper's inability to detect hidden information. Robustness to the amount of modification the stego medium can withstand before an adversary can destroy hidden information.

KEY WORDS: Introduction, Methods to Embed files and text, LSB Algorithm LSB Embedding, Visual Attack.

---

1. INTRODUCTION

Steganography proves to be an incredibly effective way of hiding the act of communication. The ease and effectiveness of LSB embedding make it an attractive method to transmit messages without detection. With the rise in popularity of image sharing services on the Internet, it is increasingly likely that an image shared online for a short period of time would not be analyzed. It is important to note that while steganography does not guarantee that a message cannot be decoded, paring steganography with encryption provides a means of communication that is difficult to detect and can be nearly impossible for a third party to decode. While steganography can be detected by statistical attacks, relying on safety in numbers and obscure embedding patterns can limit the decoding of any particular hidden message. Steganography effectiveness, ease of implementation, and extensibility all suggest that it will be a considerable security concern for the foreseeable future. Steganography is a technique for transmitting information without detection. Steganography relies on the fact that it is difficult to detect in order to remain secure. It uses parts of an image that do not strongly influence the colours shown to embed data [3]. Where embedding is most practical varies with different image formats, but one technique that works well across formats is least significant bit embedding. Other algorithms, such as Jsteg, exploit the design of a specific image file format to embed without detection. The general principle of Steganography is that perturbing a particular value in an image using a value from the data will create a small difference in the original image. The image created by this process is a stego object. The stego object contains data from the cover and information about the data that was used to perturb the cover image. The stego object can then be decoded by the intended recipient(s) and the hidden message retrieved. Because the values in the original image are only changed slightly, an observer will struggle to visually detect that an embedding has taken place. Through this series of minor perturbations based on the message's contents the data is hidden in the cover image. A third party will have to analyze the image in order to determine if an embedding has taken place. The development of different analyses has led to an arms race between those developing steganographic algorithms and those trying to detect embeddings. From this point forward, it is assumed that the cover and data to hide are both images.

2. METHODS TO EMBED FILES AND TEXT

Image Steganography uses two methods to embed files and text into images:

1) Difference - the 'Difference' mode will output a seemingly identical image to the original input image, but this is the most noticeable mode (and thus I might remove it at a later date). By using the two images (the processed image and the original image), the 'Difference' mode compares each pixel, computes the difference, and turns it back into a byte.

2) Enlarge - the 'Enlarge' mode outputs an image 4 times bigger than the input image (2 x Width, 2 x Height). By doing this, it can has 3 times the data capacity of the 'Difference' mode, and the original image isn't required (and so this mode is the default).

3) Embed - the 'Embed' mode will output an almost identical image to the input image. It encodes the data in the last two bits of the red and Blue colour channels; but by doing this, only has half a byte per pixel.
For encryption and decryption of text messages using the secret keys steganographic system uses algorithms known as steganographic algorithms [8]. The mostly used algorithms for embedding data into images are

A. JSteg Algorithm
B. F5 Algorithm
C. LSB (Least Significant Bit) Algorithm

A. JSTEG algorithm

JSteg algorithm is one of the steganographic techniques for embedding .The hiding process will be done by replacing Least Significant Bits(LSB). JSteg algorithm replaces LSBs of quantized Discrete Courier Transform(DCT) coefficients. In this process the hiding mechanism skips all coefficients with the values of 0 or 1. This algorithm is resistant to visual attacks and offers an admirable capacity for steganographic messages[6]. Generally, JSteg steganographic algorithm embedded the messages in lossy compressed JPEG images. It has high capacity and had a compression ratio of 12%. JSteg algorithm is restricted for visual attacks and it is less immune for statistical attacks. Normally, JSteg embeds only in JPEG images. In these JPEG images, the content of the image is transformed into “frequency coefficients so as to achieve storage in
a very compressed format. There is no visual attack in the sense presented here, due to the influence of one steganographic bit up to 256 pixels[11].

B. F5 algorithm

F5 algorithm was introduced by German researchers Pfitzmann and Westfeld in order to avoid the security problem when embedding the data into the JPEG images. The F5 algorithm embeds the message into randomly chosen Discrete Courier Transform (DCT) coefficients. It utilizes matrix embedding which minimises the changes to be made to the length of certain message [5]. The F5 Algorithm provides high steganographic capacity, and can prevent visual attacks. F5 algorithm is also resistant to statistical attacks. This algorithm uses matrix encoding such that it reduces the number of changes needed to embed a message of certain length. This algorithm avoids the chi-square attack since it doesn’t replace or exchange the bits. The resistance is high for both visual and statistical attacks. It has high embedding capacity that is greater than 13%.This algorithm supports TIFF, BMP, JPEG and GIFformats.The performance of the algorithms differs with the type of cover image or source on which the data is embedded[9].

C.LSB algorithm

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 in to 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[10]. For 24 bit image, the colours 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 [10]. 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. For 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 algorithm for OPA explains the procedure of hiding the sample text in an image [2].

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.
ElseStep6: 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 procedure for data hiding using steganographic application in this project is as follows

The sender first uses the steganographic application for encrypting the secret message.

3. SECRET FILE ENCRYPTION DECRYPTION TRANSMISSION

The sender first uses the steganographic application for encrypting the Secret message.
For this encryption, the sender uses text document in which the data is written and the image as a carrier file in which the secret message or text document to be hidden.
The sender sends the carrier file and text document to the encryption phase for data embedding, in which the text document is embedded into the image file. The procedure of encryption is discussed in the next phase.
In encryption phase, the data is embedded into carrier file which was protected with the password. Now the carrier file acts as an input for the decryption phase[3].
The image in which data is hidden i.e. the carrier file is sent to the receiver using a transmission medium. E.g. Web or e-mail.
The receiver receives the carrier file and places the image in the decryption phase.
In the decryption phase, the original text document can be revealed using the appropriate password.
The decryption phase decrypts the original text document using the least significant bit decoding and decrypts the

original message.

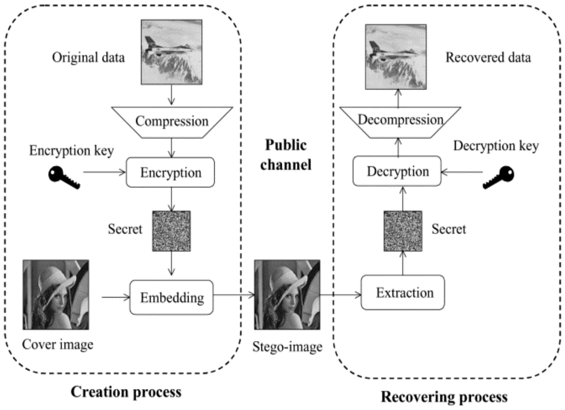

Fig-1

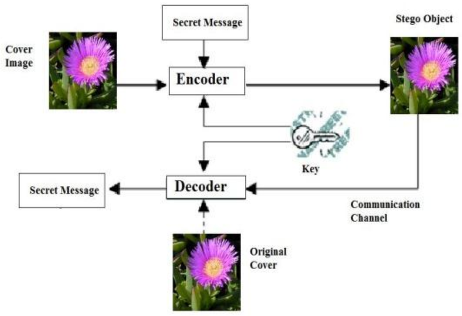

Fig-2

3.LSB Embedding

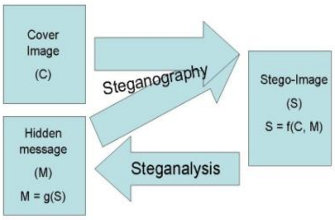

FIG -3

Least Significant Bit (LSB) embedding is a simple strategy to implement Steganography. Like all steganographic methods, it embeds the data into the cover so that it cannot be detected by a casual observer[11]. The technique works by replacing some of the information in a given pixel with information from the data in the image. While it is possible to embed data into an image on any bit-plane, LSB embedding is performed on the least significant bit(s). This minimizes the variation in colours that the embedding creates. For example, embedding into the least significant bit changes the colour value by one. Embedding into the second bit-plane can change the colour value by 2. If embedding is performed on the least significant two pixels, the result is that a colour in the cover can be any of four colours after embedding. Steganography avoids introducing as much variation as possible, to minimize the likelihood of detection. In a LSB embedding, we always lose some information from the cover image. This is an effect of embedding directly into a pixel. To do this we must discard some of the cover's information and replace it with information from the data to hide. LSB algorithms have a choice about how they embed that data to hide. They can embed losslessly, preserving all information about the data, or the data may be generalized so that it takes up less space.

Hidden files or pictures can be hidden in picture files because pictures files are so complex. Pictures on a computer are represented by tons and tons of pixels. Each pixel consists of a variation of all three primary colours, red, green and blue [1]. In a standard 24-bit bitmap, 8 bits will represent each of the three colours. 8 times 3 is 24. That means there are 256 different variations of each colour in every pixel that makes up a picture. So, to represent the colour white, the code would look like 11111111 11111111 11111111. Now, the human eye cannot distinguish the difference between too many colours and so the colour 11111110 11111110 11111110 would look exactly the same as white. Because of this, the last digit in every bit in every pixel could be changed. This is the basis of the Least Significant Bit Insertion technique. Now to show how this becomes useful. You only need 8 bits to represent Ascii text and there are three extra in every pixel of a picture. Therefore, with every three pixels, you could form one letter of Ascii text[4]. This may not seem like a lot, but when the standard image size is 640 x 480 pixels, that adds up to a lot in a hurry. In order to make this practical to the user, a computer program would be needed. After you type in your secret message and determine a cover message (the picture you want to hide you message in) the program would go through every pixel and change the last digit to represent each letter of the message you wrote. You would then send the picture to the correct recipient who would then use his program to go through every pixel and take off the last digit and use that to form the message.

The problem of using Steganography over digital communications has been solved[7]. Also, the great thing about LSB (Least Significant Bit Insertion) is that the message is not lost if the file is compressed. Anyone who uses online pictures knows that bitmap files hold a lot of information and so are generally large in size. But because the secret message is encoded into the color bits, the message is never lost when compressed. The one problem with this approach is that it does not work for every picture type. LSB works mainly with Bitmaps because of the way bitmaps are compressed[9]. JPEG's, on the other hand, are compressed using sophisticated algorithms and so a lot of the original information is lost.

Because information could so easily be lost with certain compression programs, other techniques were developed. One technique is called the Masking and Filtering technique. This technique is very similar to watermarking. The image is marked with the secret message or image and then cannot be seen unless the luminosity level is changed to an exact amount. This worked better because the text/image was now actually part of the picture and no longer in the coding part. Another technique developed used the way certain pictures are compressed to its advantage[4]. As stated earlier, JPEG's are compressed using sophisticated algorithms and because of this, a lot of the original information of the picture is lost. So, basically, what this last technique does is, it determines how the picture is going to be compressed with all the algorithms. It then changes the information of the picture accordingly to the secret message. It changes the information in a way that when decompressed, it will look similar to the LSB approach[5]. This way, when the picture is viewed, it still

looks the same but the secret message could be determined by taking the last bit of each pixel just like the LSB approach.
Today, the Internet is filled with tons of programs that uses Steganography to hide secret messages. A majority of the programs use a variation of the algorithm approach[6].

4.VISUAL ATTACK

A visual attack is the simplest way of trying to detect an embedding. It is particularly effective against LSB embeddings, but it is useless against more advanced algorithms that do not embed into the pixels of the image directly like Jsteg. A visual attack begins by looking at the image as a whole. If an embedding is detected through color abnormalities the steganographic algorithm has been successfully attacked[8]. If an embedding is not detected by the observer, the bit planes of the image are then examined, beginning with the least significant plane.

5. REFERENCES

[1]. Alfred J, M et al., 1996.Hand book of applied Cryptography.First edn.
[2]. Ali-al, H. Mohammad, A. 2010.Digital Audio Watermarking Based on the Discrete Wavelets Transform and Singular Value Decomposition,European Journal of Scientific Research. vol 39(1), pp 231-239.
[3].Amirthanjan,R. Akila,R & Deepikachowdavarapu, P., 2010. A Comparative Analysis of Image Steganography,International Journal of Computer Application , 2(3), pp.2-10.
[4]. Arnold, M. 2000. Audio watermarking: Features, applications and algorithms, Proceeding of the IEEE International Conference on Multimedia and Expo, pp 1013-1016.
[5].Bandyopadhyay, S.K., 2010.An Alternative Approach of Steganography UsingReference Image.International Journal of Advancements in Technology , 1(1), pp.05-11.
[6].Bloom,J. A. et al.,2008.Digital watermarking and Steganography . 2nd ed.
[7].MorganKaufmann.Bishop, M., 2005. Introduction to computer security. 1st ed. Pearson publications.
[8]. Cachin, C., 2004. Information: Theoretic model for steganography. Work shop on information hiding, USA.
[9]. Chan, C.K. Cheng, L.M., 2004. Hiding data in images by simple lsb substitution: pattern recognition. vol 37.
[10]. Pergamon.Cox, I. Miller, M. Bloom, J. Fridrich, J & Kalker, T. 2008. Digital watermarking and Steganography. 2ndEd. Elsevier.
[11]. Cummins, J. Diskin, P. Lau, S. & Parett, R., 2004. Steganography and digitalwatermarking. School of computer science.Vol 1.