Deepa et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(1), January - 2013, pp.54-57

Volume 3, Issue 1, January 2013 **ISSN: 2277 128X**

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Study on Digital Image Steganography

Deepa S
Department of Computer Science
Periyar University College of Arts & Science, TN, India

Umarani R
Department of Computer Science
Sri Sarada College for Women, TN, India

Abstract-- Steganography is the art and science of invisible communication. The word steganography is derived from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" defining it as "covered writing". The existence of a message is secret. Steganography is usually implemented computationally, where cover Works such as text files, images, audio files, and video files are tweaked in such a way that a secret message can be embedded within them. In order to embed secret data into a cover message, the cover must contain a sufficient amount of redundant data or noise. This is because in the embedding process Steganography actually replaces this redundant data with the secret message. In image steganography the information is hidden exclusively in images. This paper intends to give an overview of image steganography.

Keywords-- steganography, image, color table, image domain, transform domain

I. INTRODUCTION

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. But it is not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography.

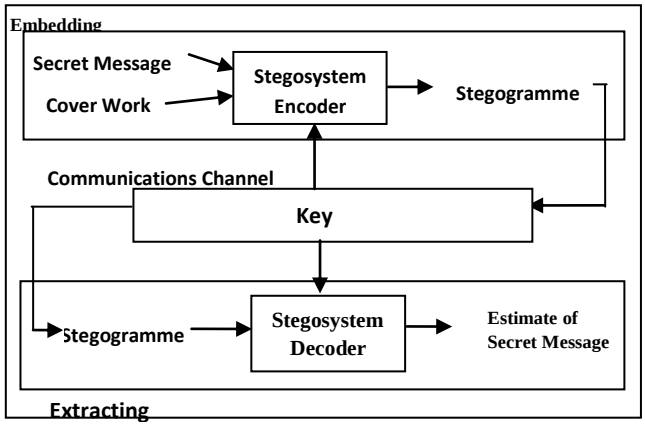
Steganography is defined by Markus Kahn as follows, "Steganography is the art and science of communicating in a way which hides the existence of the communication. The goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second message present".

The objectives of Steganography are undetectability, robustness (resistance to various image processing methods and compression) and capacity of the hidden data.

II. FRAMEWORK OF STEGANOGRAPHY

Steganography is comprised of two algorithms, one for embedding and one for extracting. The embedding process is concerned with hiding a secret message within a cover Work, and is the most carefully constructed process of the two. A great deal of attention is paid to ensuring that the secret message goes unnoticed if a third party were to intercept the cover Work. The extracting process is traditionally a much simpler process as it is simply an inverse of the embedding process, where the secret message is revealed at the end.

The entire process of steganography can be presented graphically as shown in Fig 1.



Two inputs required for the embedding process are secret message and the cover work that is used to construct a stegogramme that contains a secret message.

The inputs are passed through the stego-system encoder to embed the message within an exact copy of the cover work. The stego-system requires a key which is also used at the extraction phase. The resulting output from the stego-system encoder is the stegogramme that contains the secret message. This stegogramme is then sent over some communications

channel along with the key that was used to embed the message. Both the stegogramme and the key are then fed into the stego-system decoder where an estimate of the secret message is extracted [9].

III. IMAGE STEGANOGRAPHY

The most cover media used for steganography is image. The reason is that the large amount of redundant data present in the images can be easily altered to hide secret messages inside them without attracting attention to human visual system (HVS).

A. Image Definition

A computer image is an array of points called pixels which are represented as light intensity. The 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. The smallest bit depth in current colour schemes is 8, meaning that there are 8 bits used to describe the colour of each pixel [1]. 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 [1].

1) Color Table: In a 24-bit color scheme each pixel is represented by 3 bytes, each byte representing the intensity of the 3 primary colors Red Green and Blue respectively. Each of these 3 colors has a value that can range from 0 to 255. 0 means the color is not active and 255 means a full amount of color. Pixels with the following values make specific colors:

255 0 0 is red
0 255 0 is green
0 0 255 is blue
0 0 0 is black
255 255 255 is white

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 [1]. The RGB values for each pixel are stored in a color table. Each entry has a value for the row and a value for red, green and blue. Each pixel has a color associated with it stored in the color table. The pixel contains a value that corresponds to the row in the color table that contains the REB value for that pixel. The first number is the row number that the pixel references to get its corresponding color. The second number is the value for red, the third number is the value for green and the fourth number is the value for blue.

B. Categories of Image Steganography

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain.

Image – also known as spatial domain techniques embed messages in the intensity of the pixels directly, Image domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterised as "simple systems" [2]. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format [3].

Transform – also known as frequency domain, images are first transformed and then the message is embedded in the image [4]. Steganography in the transform domain involves the manipulation of algorithms and image transforms [2]. These methods hide messages in more significant areas of the cover image, making it more robust. Many transform domain methods are independent of the image format [3].

IV. SPATIAL DOMAIN

A. Least Significant Bit (LSB)

The simplest approach for hiding data within an image file is called Least Significant Bit insertion. The value of the LSB 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. Changing the LSB value from a 0 to a 1 will only change the color 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.

The LSBs of some or all of the bytes inside an image is changed to a bit of the secret message. If 24-bit color image is used, the amount of change will be minimal and indiscernible to the human eye. Consider the 3 adjacent pixels (9 bytes) with the following RGB encoding:

10010101 00001101 11001001
10010110 00001111 11001010
10011111 00010000 11001011

To hide the following 8 bits of data 10110110, overlaying these 8 bits over the LSB of the 9 bytes, we get the following (where bits in bold have been changed)

10010101 0000110**0** 11001001
10010111 0000111**0** 1100101**1**
10011111 00010000 11001011

The 8 bits have been successfully hidden at a cost of only changing 4 bits or roughly 50% of the LSBs. 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.

Least Significant Bit Substitution algorithms encompass two different embedding schemes: sequential and randomised [11]. In Sequential embedding the algorithm starts at the first pixel of the cover image Co,o and embeds the bits of the message data in order until there is nothing left to embed. Randomised embedding scatters the locations of the values that will be modified to contain the bits of the message data. The main reason for randomising 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.

1) LSB and Palette Based Images: Palette based images, for example GIF images are popular image file format commonly used on the Internet. By definition a GIF image cannot have a bit depth greater than 8, thus the maximum number of colours that a GIF can store is 256 [5]. GIF images are indexed images where the colours used in the image are stored in a palette, sometimes referred to as a colour lookup table [5]. Each pixel is represented as a single byte and the pixel data is an index to the colour palette [6]. The colours of the palette are typically ordered from the most used colour to the least used colours to reduce lookup time [2].

GIF images can also be used for LSB steganography, although extra care should be taken. The problem with the palette approach used with GIF images is that should one change the least significant bit of a pixel, it can result in a completely different colour since the index to the colour palette is changed [2]. If adjacent palette entries are similar, there might be little or no noticeable change, but should the adjacent palette entries be very dissimilar, the change would be evident [2]. One possible solution is to sort the palette so that the colour differences between consecutive colours are minimized [10]. Another solution is to add new colours which are visually similar to the existing colours in the palette. This requires the original image to have less unique colours than the maximum number of colours (this value depends on the bit depth used).

In an 8-bit greyscale GIF image, there are 256 different shades of grey [6]. The changes between the colours are very gradual, making it harder to detect. LSB in GIF is a very efficient algorithm to use when embedding a reasonable amount of data in a greyscale image.

2) LSB in BMP: When embedding a message in a "raw" image, that has not been changed with compression, such as a BMP, there exists a trade-off between the invisibility of the message and the amount of information that can be embedded. A BMP is capable of hiding quite a large message, but the fact that more bits are altered results in a larger possibility that the altered bits can be seen with the human eye.

3) Patchwork: Patchwork is a statistical technique that uses redundant pattern encoding to embed a message in an image [6]. The algorithm adds redundancy to the hidden information and then scatters it throughout the image [2]. A pseudorandom generator is used to select two areas of the image (or patches), patch A and patch B [7]. All the pixels in patch A is lightened while the pixels in patch B is darkened [7]. In other words the intensities of the pixels in the one patch are increased by a constant value, while the pixels of the other patch are decreased with the same constant value [10]. The contrast changes in this patch subset encodes one bit and the changes are typically small and imperceptible, while not changing the average luminosity [2].

A disadvantage of the patchwork approach is that only one bit is embedded. One can embed more bits by first dividing the image into sub-images and applying the embedding to each of them [8]. The advantage of using this technique is that the secret message is distributed over the entire image, so should one patch be destroyed, the others may still survive [2]. This however, depends on the message size, since the message can only be repeated throughout the image if it is small enough. If the message is too big, it can only be embedded once [6].

The patchwork approach is used independent of the host image and proves to be quite robust [8] against malicious or unintentional image manipulation. A stego image using patchwork be cropped or rotated, some of the message data may be lost but since the message is repeatedly embedded in the image, most of the information will survive. Patchwork is most suitable for transmitting a small amount of very sensitive information.

4) Spread Spectrum: In spread spectrum techniques, hidden data is spread throughout the cover-image making it harder to detect. A system proposed by Marvel et al. combines spread spectrum communication, error control coding and image processing to hide information in images [10].

Spread spectrum communication can be defined as the process of spreading the bandwidth of a narrowband signal across a wide band of frequencies [10]. This can be accomplished by adjusting the narrowband waveform with a wideband waveform, such as white noise. After spreading, the energy of the narrowband signal in any one frequency band is low and therefore difficult to detect [10]. In spread spectrum image steganography the message is embedded in noise and then combined with the cover image to produce the stego image. Since the power of the embedded signal is much lower than the power of the cover image, the embedded image is not perceptible to the human eye or by computer analysis without access to the original image [10].

A spread spectrum technique satisfies most requirements and is especially robust against statistical attacks, since the hidden information is scattered throughout the image. Spread spectrum techniques can be used for most steganography applications.

B. Hide & Seek: The Sequential Approach

The simplest form of image steganography is the method known as Hide & Seek [9] which replaces the LSBs of pixel values with the bits from the message bit stream.

The algorithm is so straightforward that it does not require a key to be implemented. This makes things a lot simpler to program and exchange the secret, it does mean that the security lies solely in the algorithm. If a key were used, then it

might still be impossible for the adversary to decode the hidden message, as the key would usually index the manipulated regions of the image.

Encoding

Algorithm 1 The encoding process of the Hide & Seek algorithm in sequential mode.

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

2: p = LSB(ci)

3: if p ≠ mi then

4: ci = mi

5: end if

6: end for

The algorithm works by taking the first pixel of the image ci and obtaining its LSB value. This is typically achieved by calculating the modulus 2 of the pixel value. This will return a 0 if the 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 mi that we are trying to embed. If they are already the same, then we do nothing, but if they are different then we replace ci with mi. This process continues till there are values in m that need to be encoded.

Decoding

As the encoder replaced the LSBs of the pixel values in c in sequence, the order is already known that should be used to retrieve the data. Calculate the modulus 2 of all the pixel values in the stegogramme s, and reconstruct m as mi'.

Algorithm 2 The decoding process of the Hide & Seek algorithm in sequential mode.

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

2: mi'=LSB(si)

3: end for

Run the loop for l(s) instead of l(m). If a key were used, it would probably reveal this information, but instead simply retrieve the LSB value of every pixel. When converted this to ASCII, the message will be readable up to the point that the message was encoded.

V. CONCLUSION

Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. The simpler systems can be used in such a way that they make life harder for the steganalyst, simply by embedding shorter messages. Short messages create a shorter bit-stream, which in turn requires less bit-flips to embed. With fewer modifications made to an image, it is much harder to spot a difference between the stegogramme and a clean version of the same image. It is still highly likely that a complete steganographic system might employ cryptographic measures as a safety-net to protect the content of the message in the event that the steganography is broken.

REFERENCES

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

[2] ohnson, N.F. & Jajodia, S., *Steganalysis of Images Created Using Current Steganography Software*, *Proceedings of the 2nd Information Hiding Workshop*, April 1998

[3] Venkatraman, S., Abraham, A. & Paprzycki, M., *Significance of Steganography on Data Security*, *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2004

[4] Lee, Y.K. & Chen, L.H., *High capacity image steganographic model*, *Visual Image SignalProcessing*, 147:03, June 2000

[5] *Reference Guide: Graphics Technical Options and Decisions*, http://www.devx.com/projectcool/Article/19997

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

[7] Bender, W., Gruhl, D., Morimoto, N. & Lu, A., *Techniques for data hiding*, *IBM Systems Journal*, Vol 35, 1996

[8] Petitcolas, F.A.P., Anderson, R.J. & Kuhn, M.G., *Information Hiding – A survey*, *Proceedings of the IEEE*, 87:07, July 1999

[9] Philip Bateman, *Image Steganography and Steganalysis*, August 2008

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

