AN OVERVIEW OF IMAGE STEGANOGRAPHY

T. Morkel [1], J.H.P. Eloff [2], M.S. Olivier [3]

Information and Computer Security Architecture (ICSA) Research Group

Department of Computer Science

University of Pretoria, 0002, Pretoria, South Africa

Tel: +27 12 420-2361

Fax: +27 12 362-5188

[1] E-mail: tmorkel@cs.up.ac.za

[2] E-mail: eloff@cs.up.ac.za

[3] E-mail: molivier@cs.up.ac.za

Abstract:

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 steganographic techniques some are more complex than others and all of them have respective strong and weak points. Different applications have different requirements of the steganography technique used. For example, some applications may require absolute invisibility of the secret information, while others require a larger secret message to be hidden. This paper intends to give an overview of image steganography, its uses and techniques. It also attempts to identify the requirements of a good steganographic algorithm and briefly reflects on which steganographic techniques are more suitable for which applications.

1. 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. Unfortunately it is sometimes 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 the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words "*stegos*" meaning "cover" and "*grafia*" meaning "writing" [1] defining it as "covered writing". In image steganography the information is hidden exclusively in images.

The idea and practice of hiding information has a long history. In *Histories* the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son-in-law in Greece. He shaved the head of one of his most trusted slaves and tattooed the message onto the slave's scalp. When the slave's hair grew back the slave was dispatched with the hidden message [2]. In the Second World War the Microdot technique was developed by the Germans. Information, especially photographs, was reduced in size until it was the size of a typed period. Extremely difficult to detect, a normal cover message was sent over an insecure channel with one of the periods on the paper containing hidden information [3]. Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels.

Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret [4]. Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. Once the presence of hidden information is revealed or even suspected, the purpose of steganography is partly defeated [4]. The strength of steganography can thus be amplified by combining it with cryptography.

Two other technologies that are closely related to steganography are watermarking and fingerprinting [5]. These technologies are mainly concerned with the protection of intellectual property, thus the algorithms have different requirements than steganography. These requirements of a good steganographic algorithm will be discussed below. In watermarking all of the instances of an object are "marked" in the same way. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection [6]. With fingerprinting on the other hand, different, unique marks are embedded in distinct copies of the carrier object that are supplied to different customers. This enables the intellectual property owner to identify customers who break their licensing agreement by supplying the property to third parties [5]. In watermarking and fingerprinting the fact that information is hidden inside the files may be public knowledge – sometimes it may even be visible – while in steganography the imperceptibility of the information is crucial [4]. A successful attack on a steganographic system consists of an adversary observing that there is information hidden inside a file, while a successful attack on a watermarking or fingerprinting system would not be to detect the mark, but to remove it [5].

Research in steganography has mainly been driven by a lack of strength in cryptographic systems. Many governments have created laws to either limit the strength of a cryptographic system or to prohibit it altogether [7], forcing people to study other methods of secure information transfer. Businesses have also started to realise the potential of steganography in communicating trade secrets or new product information. Avoiding communication through well-known channels greatly reduces the risk of information being leaked in transit [8]. Hiding information in a photograph of the company picnic is less suspicious than communicating an encrypted file.

This paper intends to offer a state of the art overview of the different algorithms used for image steganography to illustrate the security potential of steganography for business and personal use. After the overview it briefly reflects on the suitability of various image steganography techniques for various applications. This reflection is based on a set of criteria that we have identified for image steganography. The remainder of the paper is structured as follows: Section 2 gives the reader an overview of steganography in general and differentiates between different kinds of steganography. In section 3 the most popular algorithms for image steganography are discussed and compared in section 4. In Section 5 a conclusion is reached.

2. Overview of Steganography

To provide an overview of steganography, terms and concepts should first be explained. An overview of the different kinds of steganography is given at a later stage.

2.1 Steganography concepts

Although steganography is an ancient subject, the modern formulation of it is often given in terms of the *prisoner's problem* proposed by Simmons [9], where two inmates wish to communicate in secret to hatch an escape plan. All of their communication passes through a warden who will throw them in solitary confinement should she suspect any covert communication [10].

The warden, who is free to examine all communication exchanged between the inmates, can either be passive or active. A *passive* warden simply examines the communication to try and determine if it potentially contains secret information. If she suspects a communication to contain hidden information, a passive warden takes note of the detected covert communication, reports this to some outside party and lets the message through without blocking it. An *active* warden, on the other hand, will try to alter the communication with the suspected hidden information deliberately, in order to remove the information [5].

2.2 Different kinds of steganography

Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display [11]. The redundant bits of an object are those bits that can be altered without the alteration being detected easily [5]. Image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding. Figure 1 shows the four main categories of file formats that can be used for steganography.

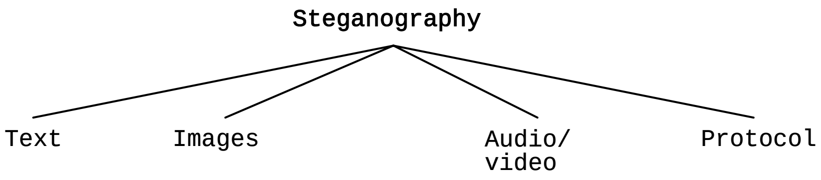

Figure 1: Categories of steganography

Hiding information in text is historically the most important method of steganography. An obvious method was to hide a secret message in every n^{th} letter of every word of a text message. It is only since the beginning of the

Internet and all the different digital file formats that is has decreased in importance [1]. Text steganography using digital files is not used very often since text files have a very small amount of redundant data.

Given the proliferation of digital images, especially on the Internet, and given the large amount of redundant bits present in the digital representation of an image, images are the most popular cover objects for steganography. This paper will focus on hiding information in images in the next sections.

To hide information in audio files similar techniques are used as for image files. One different technique unique to audio steganography is masking, which exploits the properties of the human ear to hide information unnoticeably. A faint, but audible, sound becomes inaudible in the presence of another louder audible sound [1]. This property creates a channel in which to hide information. Although nearly equal to images in steganographic potential, the larger size of meaningful audio files makes them less popular to use than images [8].

The term protocol steganography refers to the technique of embedding information within messages and network control protocols used in network transmission [13]. In the layers of the OSI network model there exist covert channels where steganography can be used [12]. An example of where information can be hidden is in the header of a TCP/IP packet in some fields that are either optional or are never used. A paper by Ahsan and Kundur provides more information on this [13].

3. Image steganography

As stated earlier, images are the most popular cover objects used for steganography. In the domain of digital images many different image file formats exist, most of them for specific applications. For these different image file formats, different steganographic algorithms exist.

3.1 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 colours 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 analyse 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]. Different steganographic algorithms have been developed for both of these compression types and will be explained in the following sections.

3.3 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.

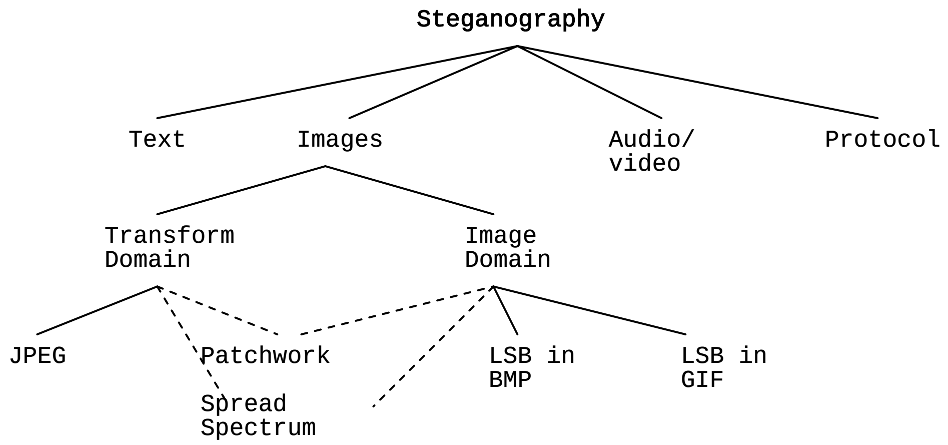

Figure 2: Categories of image steganography

3.3.1 Image Domain

- **Least Significant Bit**

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [14]. 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 colour 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 colour, changing the LSB of a pixel results in small changes in the intensity of the colours. 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 [14].

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 [4]. 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 [5].

In its simplest form, LSB makes use of BMP images, since they use lossless compression. Unfortunately to be able to hide a secret message inside a BMP file, one would require a very large cover image. Nowadays, BMP images of 800 × 600 pixels are not often used on the Internet and might arouse suspicion [19]. For this reason, LSB steganography has also been developed for use with other image file formats.

- **LSB and Palette Based Images**

Palette based images, for example GIF images, are another 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 [15]. 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 [15]. Each pixel is represented as a single byte and the pixel data is an index to the colour palette [14]. The colours of the palette are typically ordered from the most used colour to the least used colours to reduce lookup time [17].

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 [17]. 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 [17]. 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) [1]. Using this approach, one should thus carefully choose the right cover image. Unfortunately any tampering with the palette of an indexed image leaves a very clear signature, making it easier to detect.

A final solution to the problem is to use greyscale images. In an 8-bit greyscale GIF image, there are 256 different shades of grey [14]. The changes between the colours are very gradual, making it harder to detect.

3.2 Transform Domain

To understand the steganography algorithms that can be used when embedding data in the transform domain, one must first explain the type of file format connected with this domain. The JPEG file format is the most popular image file format on the Internet, because of the small size of the images.

- **JPEG compression**

To compress an image into JPEG format, the RGB colour representation is first converted to a YUV representation. In this representation the Y component corresponds to the luminance (or brightness) and the U and V components stand for chrominance (or colour) [1]. According to research the human eye is more sensitive to changes in the brightness (luminance) of a pixel than to changes in its colour [11]. This fact is exploited by the JPEG compression by downsampling the colour data to reduce the size of the file. The colour components (U and V) are halved in horizontal and vertical directions, thus decreasing the file size by a factor of 2 [1].

The next step is the actual transformation of the image. For JPEG, the Discrete Cosine Transform (DCT) is used, but similar transforms are for example the Discrete Fourier Transform (DFT). These mathematical transforms convert the pixels in such a way as to give the effect of "spreading" the location of the pixel values over part of the image [11]. The DCT transforms a signal from an image representation into a frequency representation, by grouping the pixels into 8 × 8 pixel blocks and transforming the pixel blocks into 64 DCT coefficients each [19]. A modification of a single DCT coefficient will affect all 64 image pixels in that block.

The next step is the quantization phase of the compression. Here another biological property of the human eye is exploited: The human eye is fairly good at spotting small differences in brightness over a relatively large area, but not so good as to distinguish between different strengths in high frequency brightness [1]. This means that the strength of higher frequencies can be diminished, without changing the appearance of the image. JPEG does this by dividing all the values in a block by a quantization coefficient. The results are rounded to integer values and the coefficients are encoded using Huffman coding to further reduce the size [11].

- **JPEG steganography**

Originally it was thought that steganography would not be possible to use with JPEG images, since they use lossy compression which results in parts of the image data being altered. One of the major characteristics of steganography is the fact that information is hidden in the redundant bits of an object and since redundant bits are left out when using JPEG it was feared that the hidden message would be destroyed. Even if one could somehow keep the message intact it would be difficult to embed the message without the changes being noticeable because of the harsh compression applied. However, properties of the compression algorithm have been exploited in order to develop a steganographic algorithm for JPEGs.

One of these properties of JPEG is exploited to make the changes to the image invisible to the human eye. During the DCT transformation phase of the compression algorithm, rounding errors occur in the coefficient data that are not noticeable [14]. Although this property is what classifies the algorithm as being lossy, this property can also be used to hide messages.

It is neither feasible nor possible to embed information in an image that uses lossy compression, since the compression would destroy all information in the process. Thus it is important to recognize that the JPEG compression algorithm is actually divided into lossy and lossless stages. The DCT and the quantization phase form part of the lossy stage, while the Huffman encoding used to further compress the data is lossless. Steganography can take place between these two stages. Using the same principles of LSB insertion the message can be embedded into the least significant bits of the coefficients before applying the Huffman encoding. By embedding the information at this stage, in the transform domain, it is extremely difficult to detect, since it is not in the visual domain.

3.3 Image or Transform domain

As seen in Figure 2, some steganographic algorithms can either be categorised as being in the image domain or in the transform domain depending on the implementation.

- **Patchwork**

Patchwork is a statistical technique that uses redundant pattern encoding to embed a message in an image [14]. The algorithm adds redundancy to the hidden information and then scatters it throughout the image [17]. A pseudorandom generator is used to select two areas of the image (or patches), patch A and patch B [22]. All the pixels in patch A is lightened while the pixels in patch B is darkened [22]. 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 [6]. The contrast changes in this patch subset encodes one bit and the changes are typically small and imperceptible, while not changing the average luminosity [17].

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 [23]. 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 [17]. 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 [14].

The patchwork approach is used independent of the host image and proves to be quite robust as the hidden message can survive conversion between lossy and lossless compression [23].

- **Spread Spectrum**

In spread spectrum techniques, hidden data is spread throughout the cover-image making it harder to detect [4]. A system proposed by Marvel et al. combines spread spectrum communication, error control coding and image processing to hide information in images [6].

Spread spectrum communication can be defined as the process of spreading the bandwidth of a narrowband signal across a wide band of frequencies [6]. 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 [6]. 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 [6].

4. Evaluation of different techniques

All the above mentioned algorithms for image steganography have different strong and weak points and it is important to ensure that one uses the most suitable algorithm for an application. All steganographic algorithms have to comply with a few basic requirements. The most important requirement is that a steganographic algorithm has to be imperceptible. The authors propose a set of criteria to further define the imperceptibility of an algorithm. These requirements are as follows:

- **Invisibility** – The invisibility of a steganographic algorithm is the first and foremost requirement, since the strength of steganography lies in its ability to be unnoticed by the human eye. The moment that one can see that an image has been tampered with, the algorithm is compromised

- **Payload capacity** – Unlike watermarking, which needs to embed only a small amount of copyright information, steganography aims at hidden communication and therefore requires sufficient embedding capacity.

- **Robustness against statistical attacks** – Statistical steganalysis is the practice of detecting hidden information through applying statistical tests on image data. Many steganographic algorithms leave a 'signature' when embedding information that can be easily detected through statistical analysis. To be able to pass by a warden without being detected, a steganographic algorithm must not leave such a mark in the image as be statistically significant.

- **Robustness against image manipulation** – In the communication of a stego image by trusted systems, the image may undergo changes by an active warden in an attempt to remove hidden information. Image manipulation, such as cropping or rotating, can be performed on the image before it reaches its destination. Depending on the manner in which the message is embedded, these manipulations may destroy the hidden message. It is preferable for steganographic algorithms to be robust against either malicious or unintentional changes to the image.

- **Independent of file format** – With many different image file formats used on the Internet, it might seem suspicious that only one type of file format is continuously communicated between two parties. The most powerful steganographic algorithms thus possess the ability to embed information in any type of file. This also solves the problem of not always being able to find a suitable image at the right moment, in the right format to use as a cover image.

- **Unsuspicious files** – This requirement includes all characteristics of a steganographic algorithm that may result in images that are not used normally and may cause suspicion. Abnormal file size, for example, is one property of an image that can result in further investigation of the image by a warden.

The following table compares least significant bit (LSB) insertion in BMP and in GIF files, JPEG compression steganography, the patchwork approach and spread spectrum techniques as discussed in section 3, according to the above requirements:

	LSB in BMP	**LSB in GIF**	**JPEG compression**	**Patchwork**	**Spread spectrum**
Invisibility	High*	Medium*	High	High	High
Payload capacity	High	Medium	Medium	Low	Medium
Robustness against statistical attacks	Low	Low	Medium	High	High
Robustness against image manipulation	Low	Low	Medium	High	Medium
Independent of file format	Low	Low	Low	High	High
Unsuspicious files	Low	Low	High	High	High

* - Depends on cover image used

Table 1: Comparison of image steganography algorithms

The levels at which the algorithms satisfy the requirements are defined as high, medium and low. A high level means that the algorithm completely satisfies the requirement, while a low level indicates that the algorithm has a weakness in this requirement. A medium level indicates that the requirement depends on outside influences, for example the cover image used. LSB in GIF images has the potential of hiding a large message, but only when the most suitable cover image has been chosen.

The ideal, in other words a perfect, steganographic algorithm would have a high level in every requirement. Unfortunately in the algorithms that are evaluated here, there is not one algorithm that satisfies all of the requirements. Thus a trade-off will exist in most cases, depending on which requirements are more important for the specific application.

- **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. The main disadvantage regarding LSB in BMP images is surely the suspicion that might arise from a very large BMP image being transmitted between parties, since BMP is not widely used anymore.
  Suggested applications: LSB in BMP is most suitable for applications where the focus is on the amount of information to be transmitted and not on the secrecy of that information.

- **LSB in GIF** – The strong and weak points regarding embedding information in GIF images using LSB are more or less the same as those of using LSB with BMP. The main difference is that since GIF images only have a bit depth of 8, the amount of information that can be hidden is less than with BMP. GIF images are especially vulnerable to statistical – or visual attacks – since the palette processing that has to be done leaves a very definite signature on the image. This approach is dependent on the file format as well as the image itself, since a wrong choice of image can result in the message being visible.
  Suggested applications: LSB in GIF is a very efficient algorithm to use when embedding a reasonable amount of data in a greyscale image.

- **JPEG compression –** The process of embedding information during JPEG compression results in a stego image with a high level of invisibility, since the embedding takes place in the transform domain. JPEG is the most popular image file format on the Internet and the image sizes are small because of the compression, thus making it the least suspicious algorithm to use. However, the process of the compression is a very mathematical process, making it more difficult to implement.
  Suggested applications: The JPEG file format can be used for most applications of steganography, but is especially suitable for images that have to be communicated over an open systems environment like the Internet.

- **Patchwork** – The biggest disadvantage of the patchwork approach is the small amount of information that can be hidden in one image. This property can be changed to accommodate more information but one may have to sacrifice the secrecy of the information. Patchwork’s main advantage, however, is its robustness against malicious or unintentional image manipulation. Should 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.
  Suggested applications: Patchwork is most suitable for transmitting a small amount of very sensitive information.

- **Spread spectrum –** Spread spectrum techniques satisfies most requirements and is especially robust against statistical attacks, since the hidden information is scattered throughout the image, while not changing the statistical properties.
  Suggested applications: Spread spectrum techniques can be used for most steganography applications, although its highly mathematical and intricate approach may prove too much for some.

5. Conclusion

Although only some of the main image steganographic techniques were discussed in this paper, one can see that there exists a large selection of approaches to hiding information in images. All the major image file formats have different methods of hiding messages, with different strong and weak points respectively. Where one technique lacks in payload capacity, the other lacks in robustness. For example, the patchwork approach has a very high level of robustness against most type of attacks, but can hide only a very small amount of information. Least significant bit (LSB) in both BMP and GIF makes up for this, but both approaches result in suspicious files that increase the probability of detection when in the presence of a warden.

Thus for an agent to decide on which steganographic algorithm to use, he would have to decide on the type of application he want to use the algorithm for and if he is willing to compromise on some features to ensure the security of others.

6. List of references

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[19] Krenn, R., "Steganography and Steganalysis", http://www.krenn.nl/univ/cry/steg/article.pdf

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

[21] Provos, N. & Honeyman, P., "Hide and Seek: An introduction to steganography", *IEEE Security and Privacy Journal*, 2003

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

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