

## Introduction

Steganography, from the Greek, means covered or secret writing, and is a long-practiced form of hiding information. Although related to cryptography, they are not the same. Steganography's intent is to hide the existence of the message, while cryptography scrambles a message so that it cannot be understood.

More precisely,

*“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 secret message present.”*

Steganography includes a vast array of techniques for hiding messages in a variety of media. Among these methods are invisible inks, microdots, digital signatures, covert channels and spread-spectrum communications. Today, thanks to modern technology, steganography is used on text, images, sound, signals, and more.

The advantage of steganography is that it can be used to secretly transmit messages without the fact of the transmission being discovered. Often, using encryption might identify the sender or receiver as somebody with something to hide. For example, that picture of your cat could conceal the plans for your company's latest technical innovation.

However, steganography has a number of disadvantages as well. Unlike encryption, it generally requires a lot of overhead to hide a relatively few bits of information. However, there are ways around this. Also, once a steganographic system is discovered, it is rendered useless. This problem, too, can be overcome if the hidden data depends on some sort of key for its insertion and extraction.

In fact, it is common practice to encrypt the hidden message before placing it in the cover message. However, it should be noted that the hidden message does not need to be encrypted to qualify as steganography. The message itself can be in plain English and still be a hidden message. However, most steganographers like the extra layer of protection that encryption provides. If your hidden message is found, and then at least make it as protected as possible.

This seminar aims to outline a general introduction to steganography - what it is, and where it comes from. Methods for hiding data in three varied media (text, image, and audio) will

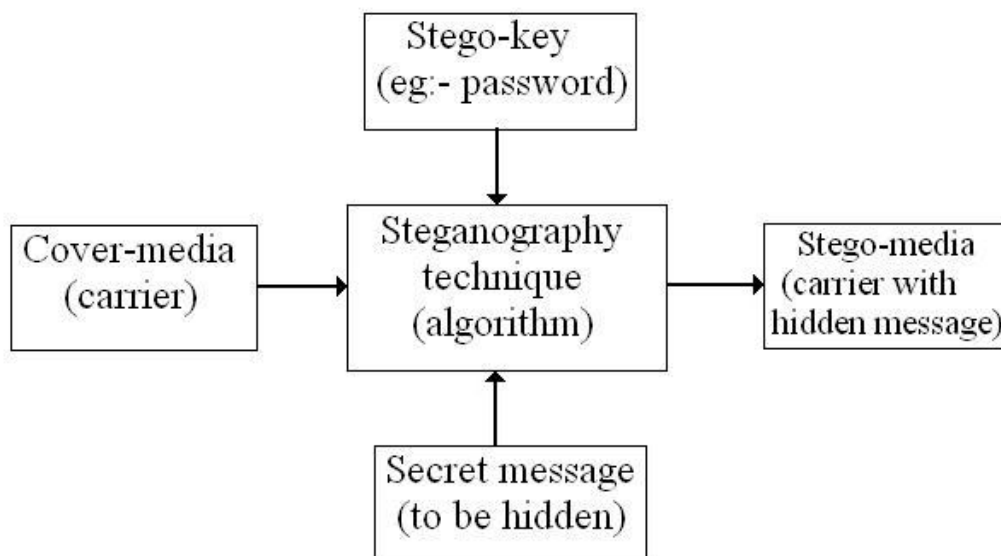
be described, and some guidelines for users of steganography will be provided where necessary. In addition, we will take a brief look at steganalysis, the science of detecting steganography, and destroying it.

## Introduction to Terms used

In the field of steganography, some terminology has developed.

The adjectives *cover*, *embedded* and *stego* were defined at the Information Hiding Workshop held in Cambridge, England. The term ``cover" is used to describe the original, innocent message, data, audio, still, video and so on. When referring to audio signal steganography, the cover signal is sometimes called the ``host" signal.

The information to be hidden in the cover data is known as the ``embedded" data. The ``stego" data is the data containing both the cover signal and the ``embedded" information. Logically, the processing of putting the hidden or embedded data, into the cover data, is sometimes known as embedding. Occasionally, especially when referring to image steganography, the cover image is known as the *container*.



## Steganography under Various Media

In the following three sections we will try to show how steganography can and is being used through the media of text, images, and audio.

Often, although it is not necessary, the hidden messages will be encrypted. This meets a requirement posed by the "Kerckhoff principle" in cryptography. This principle states that the security of the system has to be based on the assumption that the enemy has full knowledge of the design and implementation details of the steganographic system. The only missing information for the enemy is a short, easily exchangeable random number sequence, the secret key. Without this secret key, the enemy should not have the chance to even suspect that on an observed communication channel, hidden communication is taking place. Most of the software that we will discuss later meets this principle.

When embedding data, it is important to remember the following restrictions and features:

- The cover data should not be significantly degraded by the embedded data, and the embedded data should be as imperceptible as possible. (This does not mean the embedded data needs to be invisible; it is possible for the data to be hidden while it remains in plain sight.)
- The embedded data should be directly encoded into the media, rather than into a header or wrapper, to maintain data consistency across formats.
- The embedded data should be as immune as possible to modifications from intelligent attacks or anticipated manipulations such as filtering and resampling.
- Some distortion or degradation of the embedded data can be expected when the cover data is modified. To minimize this, error correcting codes should be used.
- The embedded data should be self-clocking or arbitrarily re-entrant. This ensures that the embedded data can still be extracted when only portions of the cover data are available. For example, if only a part of image is available, the embedded data should still be recoverable.

## Steganography in Text

The illegal distribution of documents through modern electronic means, such as electronic mail, means such as this allow infringers to make identical copies of documents without paying royalties or revenues to the original author. To counteract this possible wide-scale piracy, a method of marking printable documents with a unique codeword that is indiscernible to readers, but can be used to identify the intended recipient of a document just by examination of a recovered document.

The techniques they propose are intended to be used in conjunction with standard security measures. For example, documents should still be encrypted prior to transmission across a network. Primarily, their techniques are intended for use *after* a document has been decrypted, once it is readable to all.

An added advantage of their system is that it is not prone to distortion by methods such as photocopying, and can thus be used to trace paper copies back to their source.

An additional application of text steganography suggested by Bender, et al. is annotation, that is, checking that a document has not been tampered with. Hidden data in text could even be used by mail servers to check whether documents should be posted or not.

The marking techniques described are to be applied to either an image representation of a document or to a document format file, such as PostScript or Textiles. The idea is that a codeword (such as a binary number, for example) is embedded in the document by altering particular textual features. By applying each bit of the codeword to a particular document feature, we can encode the codeword. It is the type of feature that identifies a particular encoding method. Three features are described in the following subsections:

### Line-Shift Coding

In this method, text lines are vertically shifted to encode the document uniquely. Encoding and decoding can generally be applied either to the format file of a document, or the bitmap of a page image.

By moving every second line of document either 1/300 of an inch up or down, it was found that line-shift coding worked particularly well, and documents could still be completely decoded, even after the tenth photocopy.

However, this method is probably the most visible text coding technique to the reader. Also, line-shift encoding can be defeated by manual or automatic measurement of the number of pixels between text baselines. Random or uniform respacing of the lines can damage any attempts to decode the codeword.

However, if a document is marked with line-shift coding, it is particularly difficult to remove the encoding if the document is in paper format. Each page will need to be rescanned, altered, and reprinted. This is complicated even further if the printed document is a photocopy, as it will then suffer from effects such as blurring, and salt-and-pepper noise.

## **Word-Shift Coding**

In word-shift coding, codewords are coded into a document by shifting the horizontal locations of words within text lines, while maintaining a natural spacing appearance. This encoding can also be applied to either the format file or the page image bitmap. The method, of course, is only applicable to documents with variable spacing between adjacent words, such as in documents that have been text-justified. As a result of this variable spacing, it is necessary to have the original image, or to at least know the spacing between words in the unencoded document.

The following is a simple example of how word-shifting might work. For each text-line, the largest and smallest spaces between words are found. To code a line, the largest spacing is reduced by a certain amount, and the smallest is extended by the same amount. This maintains the line length, and produces little visible change to the text.

Word-shift coding should be less visible to the reader than line-shift coding, since the spacing between adjacent words on a line is often shifted to support text justification.

However, word-shifting can also be detected and defeated, in either of two ways.

If one knows the algorithm used by the formatter for text justification, actual spaces between words could then be measured and compared to the formatter's expected spacing. The differences in spacing would reveal encoded data.

A second method is to take two or more distinctly encoded, uncorrupted documents and perform page by page pixel-wise difference operations on the page images. One could then quickly pick up word shifts and the size of the word displacement.

By respacing the shifted words back to the original spacing produced under the formatter, or merely applying random horizontal shifts to all words in the document not found at column edges, an attacker could eliminate the encoding. However, it is felt that these methods would be time-consuming and painstaking.

## Feature Coding

A third method of coding data into text is known as feature coding. This is applied either to the bitmap image of a document, or to a format file. In feature coding, certain text features are altered, or not altered, depending on the codeword. For example, one could encode bits into text by extending or shortening the upward, vertical endlines of letters such as b, d, h, etc. Generally, before encoding, feature randomization takes place. That is, character endline lengths would be randomly lengthened or shortened, then altered again to encode the specific data. This removes the possibility of visual decoding, as the original endline lengths would not be known. Of course, to decode, one requires the original image, or at least a specification of the change in pixels at a feature.

Due to the frequently high number of features in documents that can be altered, feature coding supports a high amount of data encoding. Also, feature encoding is largely indiscernible to the reader. Finally, feature encoding can be applied directly to image files, which leaves out the need for a format file.

When trying to attack a feature-coded document, it is interesting that a purely random adjustment of endline lengths is not a particularly strong attack on this coding method. Feature coding can be defeated by adjusting each endline length to a fixed value. This can be done manually, but would be painstaking. Although this process can be automated, it can be made more challenging by varying the particular feature to be encoded. To even further complicate the issue, word shifting might be used in conjunction with feature coding, for example. Efforts such as this can place enough impediments in the attacker's way to make his job difficult and time-consuming.

## Alternative Methods

Alternative, interesting, major three text-coding methods of encoding data are:

- Open space methods, similar to the ones given

- Syntactic methods, that utilize punctuation and contractions
- Semantic methods, that encode using manipulation of the words themselves

The syntactic and semantic methods are particularly interesting. In syntactic methods, multiple methods of punctuation are harnessed to encode data. For example, the two phrases below are both considered correct, although the first line has an extra comma:

bread, butter, and milk

bread, butter and milk

Alternation between these two forms of listing can be used to represent binary data. Other methods of syntactic encoding include the controlled use of contractions and abbreviations. Although such syntactic encoding is very possible in the English language, the amount of data that could be encoded would be very low, somewhere in the order of a several bits per kilobyte of text.

The final category of data hiding suggested by Bender, et al. is semantic methods. By assigning values to synonyms, data could be encoded into the actual words of the text. For example, the word *big* might be given a value of one, the word *large* a value of zero. Then, when the word *big* is encountered in the coded text, a value of one can be decoded. Further synonyms can mean greater bit encoding. However, these methods can sometimes interfere with the nuances of meaning.

## Steganography in Images

In this section 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, ciphertext, 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.

### Some Guidelines to Image Steganography

Before proceeding further, some explanation of image files is necessary. 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. The pixel's value, then, is between 0 and 255. The image software merely needs to paint the indicated color on the screen at the selected pixel position.

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.

## Image Compression

Image compression offers a solution to large image files. Two kinds of image compression are *lossless* and *lossy* compression. Both methods save storage space but have differing effects on any uncompressed hidden data in the image.

Lossy compression, as typified by JPEG (Joint Photographic Experts Group) format files, offers high compression, but may not maintain the original image's integrity. This can impact negatively on any hidden data in the image. This is due to the lossy compression algorithm, which may "lose" unnecessary image data, providing a close approximation to high-quality digital images, but not an exact duplicate. Hence, the term "lossy" compression. Lossy compression is frequently used on true-color images, as it offers high compression rates.

Lossless compression maintains the original image data exactly; hence it is preferred when the original information must remain intact. It is thus more favored by steganographic techniques. Unfortunately, lossless compression does not offer such high compression rates as lossy compression. Typical examples of lossless compression formats are CompuServe's GIF (Graphics Interchange Format) and Microsoft's BMP (Bitmap) format.

## Image Encoding Techniques

Information can be hidden many different ways in images. Straight message insertion can be done, which will simply encode every bit of information in the image. More complex encoding can be done to embed the message only in "noisy" areas of the image, that will attract less attention. The message may also be scattered randomly throughout the cover image.

The most common approaches to information hiding in images are:

- Least significant bit (LSB) insertion
- Masking and filtering techniques
- Algorithms and transformations

Each of these can be applied to various images, with varying degrees of success. Each of them suffers to varying degrees from operations performed on images, such as cropping, or resolution decrementing, or decreases in the color depth.

## Least Significant bit insertion

The least significant bit insertion method is probably the most well known image steganography technique. It is a common, simple approach to embedding information in a graphical image file. Unfortunately, it is extremely vulnerable to attacks, such as image manipulation. A simple conversion from a GIF or BMP format to a lossy compression format such as JPEG can destroy the hidden information in the image.

When applying LSB techniques to each byte of a 24-bit image, three bits can be encoded into each pixel. (As each pixel is represented by three bytes) Any changes in the pixel bits will be indiscernible to the human eye. For example, the letter A can be hidden in three pixels. Assume the original three pixels are represented by the three 24-bit words below:

```
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
```

The binary value for the letter A is (10000011). Inserting the binary value of A into the three pixels, starting from the top left byte, would result in:

```
(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)
```

The emphasized bits are the only bits that actually changed. The main advantage of LSB insertion is that data can be hidden in the least and second to least bits and still the human eye would be unable to notice it.

When using LSB techniques on 8-bit images, more care needs to be taken, as 8-bit formats are not as forgiving to data changes as 24-bit formats are. Care needs to be taken in the selection of the cover image, so that changes to the data will not be visible in the stego-image. Commonly known images, (such as famous paintings, like the *Mona Lisa*) should be avoided. In fact, a simple picture of your dog would be quite sufficient.

When modifying the LSB bits in 8-bit images, the pointers to entries in the palette are changed. It is important to remember that a change of even one bit could mean the difference between a shade of red and a shade of blue. Such a change would be immediately noticeable on the displayed image, and is thus unacceptable. For this reason, data-hiding experts recommend using grey-scale palettes, where the differences between shades are not as pronounced. Alternatively, images consisting mostly of one color, such as the so-called Renoir palette, named because it comes from a 256 color version of Renoir's "Le Moulin de la Galette".

## Masking and Filtering

Masking and filtering techniques hide information by marking an image in a manner similar to paper watermarks. Because watermarking techniques are more integrated into the image, they may be applied without fear of image destruction from lossy compression. By covering, or masking a faint but perceptible signal with another to make the first non-perceptible, we exploit the fact that the human visual system cannot detect slight changes in certain temporal domains of the image.

Technically, watermarking is not a steganographic form. Strictly, steganography conceals data in the image; watermarking extends the image information and becomes an attribute of the cover image, providing license, ownership or copyright details.

Masking techniques are more suitable for use in lossy JPEG images than LSB insertion because of their relative immunity to image operations such as compression and cropping.

## Algorithms and transformations

Because they are high quality color images with good compression, it is desirable to use JPEG images across networks such as the Internet. Indeed, JPEG images are becoming abundant on the Internet.

JPEG images use the discrete cosine transform (DCT) to achieve compression. DCT is a lossy compression transform, because the cosine values cannot be calculated precisely, and rounding errors may be introduced. Variances between the original data and the recovered data depend on the values and methods used to calculate the DCT.

Images can also be processed using fast Fourier transformation and wavelet transformation. Other properties such as luminance can also be utilized. The HVS has a very low

sensitivity to small changes in luminance, being able to discern changes of no less than one part in thirty for random patterns. This figure goes up to one part in 240 for uniform regions of an image.

Modern steganographic systems use spread-spectrum communications to transmit a narrowband signal over a much larger bandwidth so that the spectral density of the signal in the channel looks like noise.

The two different spread-spectrum techniques these tools employ are called direct-sequence and frequency hopping. The former hides information by phase-modulating the data signal (carrier) with a pseudorandom number sequence that both the sender and the receiver know. The latter divides the available bandwidth into multiple channels and hops between these channels (also triggered by a pseudorandom number sequence).

The Patchwork method is based on a pseudorandom, statistical process that takes advantage of the human weaknesses to luminance variation. Using *redundant pattern encoding* to repeatedly scatter hidden information throughout the cover image, like a patchwork, Patchwork can hide a reasonably small message many times in a image. In the Patchwork method,  $n$  pairs of image points  $(a, b)$  are randomly chosen. The brightness of  $a$  is decreased by one and the brightness of  $b$  is increased by one. For a labeled image, the expected value of the sum of the differences of the  $n$  pairs of points is then  $2n$ . Bender shows that after JPEG compression, with the quality factor set to 75, the message can still be decoded with an 85

This algorithm is more robust to image processing such as cropping and rotating, but at the cost of message size. Techniques such as Patchwork are ideal for watermarking of images. Even if the image is cropped, there is a good probability that the watermark will still be readable.

Other methods also attempt to mark labels into the images by altering the brightness of pixel blocks of the image by a selected value  $k$ . This value  $k$  is dependent on a lower quality JPEG compressed version of the labeled block. This method is fairly resistant to JPEG compression, depending on the size of the pixel blocks used, and offers low visibility of the label. Unfortunately, it is not very suitable to real-time applications.

Other techniques encrypt and scatter the hidden throughout the image in some pre-determined manner. It is assumed that even if the message bits are extracted, they will be useless without the algorithm and stego-key to decode them. Although such techniques do help protect

against hidden message extraction, they are not immune to destruction of the hidden message through image manipulation.

## Image Downgrading Problem

In multilevel security systems, such as the ones used by the army, it sometimes becomes necessary to declassify some information from a high level of access to a lower level. Unfortunately, downgrading of images can present a problem. Information could be covertly hidden in a "top secret" image for later retrieval when the image is declassified.

## Steganography in Audio

Because of the range of the human auditory system (HAS), data hiding in audio signals is especially challenging. The HAS perceives over a range of power greater than one billion to one and range of frequencies greater than one thousand to one. Also, the auditory system is very sensitive to additive random noise. Any disturbances in a sound file can be detected as low as one part in ten million (80dB below ambient level). However, while the HAS has a large dynamic range, it has a fairly small differential range - large sounds tend to drown quiet sounds.

When performing data hiding on audio, one must exploit the weaknesses of the HAS, while at the same time being aware of the extreme sensitivity of the human auditory system.

## Audio Environments

When working with transmitted audio signals, one should bear in mind two main considerations. First, the means of audio storage, or digital representation of the audio, and second, the transmission medium the signal might take.

### Digital representation

Digital audio files generally have two primary characteristics:

- *Sample quantization method:* The most popular format for representing samples of high-quality digital audio is a 16-bit linear quantization, such as that used by WAV (Windows Audio-Visual) and AIFF (Audio Interchange File Format). Some signal distortion is introduced by this format.

- *Temporal sampling rate*: The most popular temporal sampling rates for audio include 8 kHz (kilohertz, 9.6 kHz, 10 kHz, 12 kHz, 16 kHz, 22.05 kHz and 44.1 kHz). Sampling rate puts an upper bound on the usable portion of the frequency range. Generally, usable data space increases at least linearly with increased sampling rate.

Another digital representation that should be considered is the ISO MPEG-Audio format, a *perceptual* encoding standard. This format drastically changes the statistics of the signal by encoding only the parts the listener perceives, thus maintaining the sound, but changing the signal.

## Transmission medium

The transmission medium, or transmission environment, of an audio signal refers to the environments the signal might go through on its way from encoder to decoder.

Bender identifies four possible transmission environments:

- *Digital end-to-end environment*: If a sound file is copied directly from machine to machine, but never modified, then it will go through this environment. As a result, the sampling will be exactly the same between the encoder and decoder. Very little constraints are put on data-hiding in this environment.
- *Increased/decreased resampling environment*: In this environment, signals is resampled to a higher or lower sampling rate, but remains digital throughout. Although the absolute magnitude and phase of most of the signal are preserved, the temporal characteristics of the signal are changed.
- *Analog transmission and resampling*: This occurs when a signal is converted to an analog state, played on a relatively clean analog line, and resampled. Absolute signal magnitude, sample quantization and temporal sampling rate are not preserved. In general, phase will be preserved.
- *"Over the air" environment*: This occurs when the signal is "played into the air" and "resampled with a microphone". The signal will be subjected to possible unknown nonlinear modifications causing phase changes, amplitude changes, drifting of different frequency components, echoes, etc.

The signal representation and transmission environment both need to be considered when choosing a data-hiding method.

# Methods of Audio Data Hiding

Some methods of audio data-hiding are,

## Low-bit encoding

Similarly to how data was stored in the least-significant bit of images, binary data can be stored in the least-significant bit of audio files. Ideally the channel capacity is 1kb per second per kilohertz; so for example, the channel capacity would be 44kbps in a 44kHz sampled sequence. Unfortunately, this introduces audible noise. Of course, the primary disadvantage of this method is its poor immunity to manipulation. Factors such as channel noise and resampling can easily destroy the hidden signal.

A particularly robust implementation of such a method results in a slight amplitude modification of each sample in a way that does not produce any perceptual difference. Their implementation offers high robustness to MPEG compression plus other forms of signal manipulation, such as filtering, resampling and requantization.

## Phase coding

The phase coding method works by substituting the phase of an initial audio segment with a reference phase that represents the data. The procedure for phase coding is as follows:

- The original sound sequence is broken into a series of N short segments.
- A discrete Fourier transform (DFT) is applied to each segment, to break create a matrix of the phase and magnitude.
- The phase difference between each adjacent segment is calculated.
- For segment S<sub>0</sub>, the first segment, an artificial absolute phase p<sub>0</sub> is created.
- For all other segments, new phase frames are created.
- The new phase and original magnitude are combined to get a new segment, S<sub>n</sub>.
- Finally, the new segments are concatenated to create the encoded output.

For the decoding process, the synchronization of the sequence is done before the decoding. The length of the segment, the DFT points, and the data interval must be known at the receiver. The value of the underlying phase of the first segment is detected as 0 or 1, which represents the coded binary string.

## Spread spectrum

Most communication channels try to concentrate audio data in as narrow a region of the frequency spectrum as possible in order to conserve bandwidth and power. When using a spread spectrum technique, however, the encoded data is spread across as much of the frequency spectrum as possible.

One particular method, Direct Sequence Spread Spectrum (DSSS) encoding, spreads the signal by multiplying it by a certain maximal length pseudorandom sequence, known as a *chip*. The sampling rate of the host signal is used as the *chip rate* for coding. The calculation of the start and end quanta for phase locking purposes is taken care of by the discrete, sampled nature of the host signal. As a result, a higher chip rate and therefore a higher associated data rate, is possible.

However, unlike phase coding, DSSS does introduce additive random noise to the sound.

## Echo data hiding

Echo data hiding embeds data into a host signal by introducing an echo. The data are hidden by varying three parameters of the echo: initial amplitude, decay rate, and offset, or delay. As the offset between the original and the echo decreases, the two signals blend. At a certain point, the human ear cannot distinguish between the two signals, and the echo is merely heard as added resonance. This point depends on factors such as the quality of the original recording, the type of sound, and the listener.

By using two different delay times, both below the human ear's perceptual level, we can encode a binary one or zero. The decay rate and initial amplitude can also be adjusted below the audible threshold of the ear, to ensure that the information is not perceivable. To encode more than one bit, the original signal is divided into smaller portions, each of which can be echoed to encode the desired bit. The final encoded signal is then just the recombination of all independently encoded signal portions.

As a binary one is represented by a certain delay  $y$ , and a binary zero is represented by a certain delay  $x$ , detection of the embedded signal then just involves the detection of spacing between the echoes.

Echo hiding was found to work exceptionally well on sound files where there is no additional degradation, such as from line noise or lossy encoding, and where there is no gaps of silence. Work to eliminate these drawbacks is being done.

## **Steganography in File Systems and TCP/IP Packets**

The way operating systems store files typically results in unused space that appears to be allocated to a file. For example, under Windows 95 operating system, drives formatted as FAT16 (MS-DOS compatible) without compression use cluster sizes of around 32 kilobytes (K). What this means is that the minimum space allocated to a file is 32K. If a file is 1K in size, then an additional 31K is "wasted" due to the way storage space is allocated. This "extra" space can be used to hide information without showing up in the directory.

Another method of hiding information in file systems is to create a hidden partition. These partitions are not seen if the system is started normally. However, in many cases, running a disk configuration utility (such as DOS's FDISK) exposes the hidden partition. These concepts have been expanded and a novel proposal of a steganographic file system. If the user knows the file name and password, then an access is granted to the file; otherwise, no evidence of the file exists in the system.

Protocols in the OSI network model have vulnerabilities that can be used to hide information. TCP/IP packets used to transport information across the Internet have unused space in the packet headers. The TCP packet header has six unused (reserved) bits and the IP packet header has two reserved bits. Thousands of packets are transmitted with each communication channel, which provides an excellent covert communication channel if unchecked.

The ease in use and abundant availability of steganography tools has law enforcement concerned in trafficking of illicit material via web page images, audio, and other files being transmitted through the Internet. Methods of message detection and understanding the thresholds of current technology are necessary to uncover such activities.

## Steganalysis

Whereas the goal of steganography is the avoidance of suspicion to hidden messages in other data, steganalysis aims to discover and render useless such covert messages.

Hiding information within electronic media requires alterations of the media properties that may introduce some form of degradation or unusual characteristics. These characteristics may act as signatures that broadcast the existence of the embedded message, thus defeating the purpose of steganography.

Attacks and analysis on hidden information may take several forms: detecting, extracting, and disabling or destroying hidden information. An attacker may also embed counter-information over the existing hidden information. Here two methods are looked into: detecting messages or their transmission and disabling embedded information. These approaches (attacks) vary depending upon the methods used to embed the information in to the cover media

Some amount of distortion and degradation may occur to carriers of hidden messages even though such distortions cannot be detected easily by the human perceptible system. This distortion may be anomalous to the "normal" carrier that when discovered may point to the existence of hidden information. Steganography tools vary in their approaches for hiding information. Without knowing which tool is used and which, if any, stegokey is used; detecting the hidden information may become quite complex. However, some of the steganographic approaches have characteristics that act as signatures for the method or tool used.

### Detecting Hidden Information

Unusual patterns stand out and expose the possibility of hidden information. In text, small shifts in word and line spacing may be somewhat difficult to detect by the casual observer. However, appended spaces and "invisible" characters can be easily revealed by opening the file with a common word processor. The text may look "normal" if typed out on the screen, but if the file is opened in a word processor, the spaces, tabs, and other characters distort the text's presentation.

Images too may display distortions from hidden information. Selecting the proper combination of steganography tools and carriers is the key to successful information hiding.

Some images may become grossly degraded with even small amounts of embedded information. This “visible noise” will give away the existence of hidden information.

The same is true with audio. Echoes and shadow signals reduce the chance of audible noise, but they can be detected with little processing. Only after evaluating many original images and stegoimages as to color composition, luminance, and pixel relationships do anomalies point to characteristics that are not "normal" in other images.

Patterns become visible when evaluating many images used for applying steganography. Such patterns are unusual sorting of color palettes, relationships between colors in color indexes, exaggerated "noise" An approach used to identify such patterns is to compare the original cover-images with the stego-images and note visible differences (known-cover attack). Minute changes are readily noticeable when comparing the cover and stego-images. In making these comparisons with numerous images, patterns begin to emerge as possible signatures of steganography software. Some of these signatures may be exploited automatically to identify the existence of hidden messages and even the tools used in embedding the messages. With this knowledge-base, if the cover images are not available for comparison, the derived known signatures are enough to imply the existence of a message and identify the tool used to embed the message. However, in some cases recurring, predictable patterns are not readily apparent even if distortion between the cover and stego-images is noticeable.

A number of disk analysis utilities are available that can report and filter on hidden information in unused clusters or partitions of storage devices. A steganographic file system may also be vulnerable to detection through analysis of the systems partition information.

Filters can also be applied to capture TCP/IP packets that contain hidden or invalid information in the packet headers. Internet firewalls are becoming more sophisticated and allow for much customization. Just as filters can be set to determine if packets originate from within the firewall's domain and the validity of the SYN and ACK bits, so to can the filters be configured to catch packets that have information in supposed unused or reserved space.

## **Disabling Steganography**

Detecting the existence of hidden information defeats the steganography's goal of imperceptibility. Methods exist, that produce results which are far more difficult to detect without the original image for comparison. At times the existence of hidden information may be known so detecting it is not always necessary. Disabling and rendering it useless seems to be the next best alternative. With each method of hiding information there is a trade off between the

sizes of the payload (amount of hidden information) that can be embedded and the survivability or robustness of that information to manipulation.

The distortions in text noted by appended spaces and "invisible" characters can be easily revealed by opening the file with a word processor. Extra spaces and characters can be quickly stripped from text documents. The disabling or removal of hidden information in images comes down to image processing techniques.

For LSB methods of inserting data, simply using a lossy compression technique, such as JPEG, is enough to render the embedded message useless. Images compressed with such a method are still pleasing to the human eye but no longer contain the hidden information.

Tools exist to test the robustness of information hiding techniques in images. These tools automate image processing techniques such as warping, cropping, rotating, and blurring. Such tools and techniques should be used by those considering making the investment of watermarking to provide a sense of security of copyright and licensing just as password cracking tools are used by system administrators to test the strength of user and system passwords. If the password fails, the administrator should notify the password owner that the password is not secure.

Hidden information may also be overwritten. If information is added to some media such that the added information cannot be detected, then there exists some amount of additional information that may be added or removed within the same threshold which will overwrite or remove the embedded covert information.

Audio and video are vulnerable to the same methods of disabling as with images. Manipulation of the signals will alter embedded signals in the noise level (LSB) which may be enough to overwrite or destroy the embedded message. Filters can be used in an attempt to cancel out echoes or subtle signals but becomes this may not be as successful as expected.

Caution must be used in hiding information in unused space in files or file systems. File headers and reserved spaces are common places to look for "out of place" information. In file systems, unless the steganographic areas are in some way protected (as in a partition), the operating system may freely overwrite the hidden data since the clusters are thought to be free. This is a particular annoyance of operating systems that do a lot of caching and creating of temporary files. Utilities are also available which "clean" or wipe unused storage areas. In wiping, clusters are overwritten several times to ensure any data has been removed. Even in this extreme case, utilities exist that may recover portions of the overwritten information.

As with unused or reserved space in file headers, TCP/IP packet headers can also be reviewed easily. Just as firewall filters are set to test the validity of the source and destination IP

addresses, the SYN and ACK bits, so to can the filters be configured to catch packets that have information in supposed unused or reserved space. If IP addresses are altered or spoofed to pass covert information, a reverse lookup in a domain name service (DNS) can verify the address. If the IP address is false, the packet can be terminated. Using this technique to hide information is risky as TCP/IP headers may get overwritten in the routing process. Reserved bits can be overwritten and passed along without impacting the routing of the packet.

## Comments and Conclusion

This seminar provides an overview of steganalysis and introduced some characteristics of steganographic software that point signs of information hiding. This work is but a fraction of the steganalysis approach. To date general detection techniques as applied to steganography have not been devised and methods beyond visual analysis are being explored. Too many images exist to be reviewed manually for hidden messages so development of a tool to automate the process will be beneficial to analysts. The ease in use and abundant availability of steganography tools has law enforcement concerned in trafficking of illicit material via web page images, audio, and other transmissions over the Internet.

Methods of message detection and understanding the thresholds of current technology are under investigation. Success in steganographic secrecy results from selecting the proper mechanisms. However, a stegomedium which seems innocent enough may, upon further investigation, actually broadcast the existence of embedded information.

Development in the area of covert communications and steganography will continue. Research in building more robust methods that can survive image manipulation and attacks continues to grow. The more information is placed in the public's reach on the Internet, the more owners of such information need to protect themselves from theft and false representation. Systems to recover seemingly destroyed information and steganalysis techniques will be useful to law enforcement authorities in computer forensics and digital traffic analysis.

Most data-hiding systems take advantage of human perceptual weaknesses, but have weaknesses of their own. For now, it seems that no system of data-hiding is totally immune to attack.

However, steganography has its place in security. It in no way can replace cryptography, but is intended to supplement it. Its application in watermarking and fingerprinting, for use in detection of unauthorized, illegally copied material, is continually being realized and developed.

Also, in places where standard cryptography and encryption is outlawed, steganography can be used for covert data transmission. Steganography, formerly just an interest of the military, is now gaining popularity among the masses. Soon, any computer user will be able to put his own watermark on his artistic creations.