

An Overview of Steganography for the Computer Forensics Examiner

[Gary C. Kessler](#)

Associate Professor
Computer & Digital Forensics Program
Champlain College
Burlington, Vermont
February 2004

[An edited version of this paper appears in the July 2004 issue of [Forensic Science Communications](#).]

Contents

[Abstract](#) | [Introduction](#) | [Null Ciphers](#) | [Digital Image and Audio](#) | [Digital Carrier Methods](#)
[Steganography Examples](#) | [Detecting Steganography](#) | [Steganography Detection Tools](#) | [Summary and Conclusions](#)
[References](#) | [Additional Websites](#) | [Companion Downloads](#) | [Commercial Vendors](#) | [Author's Bio](#)

Abstract

Steganography is the art of covered or hidden writing. The purpose of steganography is covert communication-to hide the existence of a message from a third party. This paper is intended as a high-level technical introduction to steganography for those unfamiliar with the field. It is directed at forensic computer examiners who need a practical understanding of steganography without delving into the mathematics, although references are provided to some of the ongoing research for the person who needs or wants additional detail. Although this paper provides a historical context for steganography, the emphasis is on digital applications, focusing on hiding information in online image or audio files. Examples of software tools that employ steganography to hide data inside of other files as well as software to detect such hidden files will also be presented.

Introduction

Steganography is the art of covered or hidden writing. The purpose of steganography is covert communication to hide a message from a third party. This differs from cryptography, the art of secret writing, which is intended to make a message unreadable by a third party but does not hide the existence of the secret communication. Although steganography is separate and distinct from cryptography, there are many analogies between the two, and some authors categorize steganography as a form of cryptography since hidden communication is a form of secret writing (Bauer 2002). Nevertheless, this paper will treat steganography as a separate field.

Although the term steganography was only coined at the end of the 15th century, the use of steganography dates back several millennia. In ancient times, messages were hidden on the back of wax writing tables, written on the stomachs of rabbits, or tattooed on the scalp of slaves. Invisible ink has been in use for centuries-for fun by children and students and for serious espionage by spies and terrorists. Microdots and microfilm, a staple of war and spy movies, came about after the invention of photography (Arnold et al. 2003; Johnson et al. 2001; Kahn 1996; Wayner 2002).

Steganography hides the covert message but not the fact that two parties are communicating with each other. The steganography process generally involves placing a hidden message in some transport medium, called the carrier. The secret message is embedded in the carrier to form the steganography medium. The use of a steganography key may be employed for encryption of the hidden message and/or for randomization in the steganography scheme. In summary:

$$\text{steganography_medium} = \text{hidden_message} + \text{carrier} + \text{steganography_key}$$

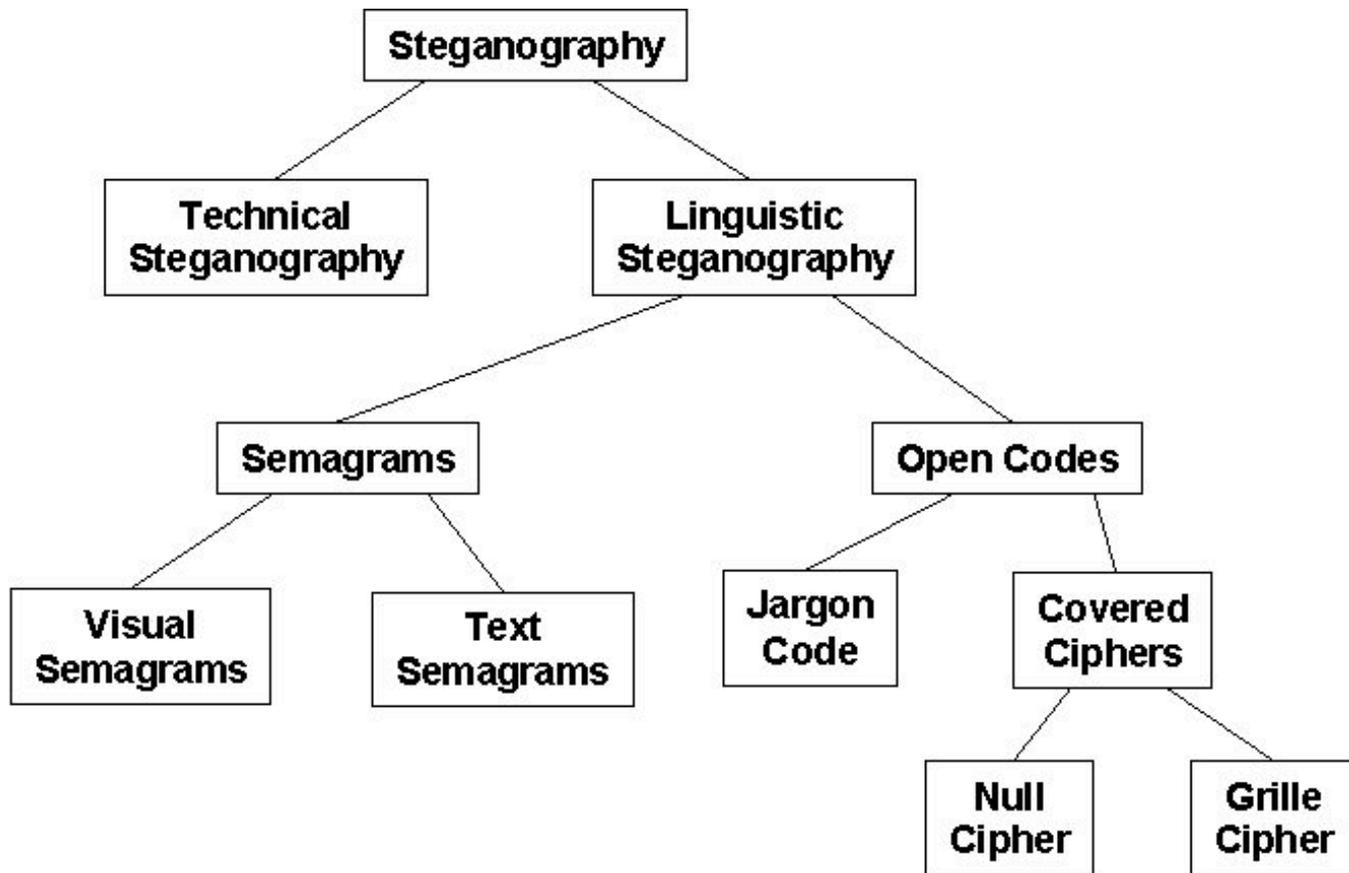


Figure 1. Classification of Steganography Techniques (Adapted from Bauer 2002).

Figure 1 shows a common taxonomy of steganographic techniques (Arnold et al. 2003; Bauer 2002).

- Technical steganography uses scientific methods to hide a message, such as the use of invisible ink or microdots and other size-reduction methods.
- Linguistic steganography hides the message in the carrier in some nonobvious ways and is further categorized as semagrams or open codes.
- Semagrams hide information by the use of symbols or signs. A visual semagram uses innocent-looking or everyday physical objects to convey a message, such as doodles or the positioning of items on a desk or Website. A text semagram hides a message by modifying the appearance of the carrier text, such as subtle changes in font size or type, adding extra spaces, or different flourishes in letters or handwritten text.
- Open codes hide a message in a legitimate carrier message in ways that are not obvious to an unsuspecting observer. The carrier message is sometimes called the overt communication whereas the hidden message is the covert communication. This category is subdivided into jargon codes and covered ciphers.
- Jargon code, as the name suggests, uses language that is understood by a group of people but is meaningless to others. Jargon codes include warchalking (symbols used to indicate the presence and type of wireless network signal [Warchalking 2003]), underground terminology, or an innocent conversation that conveys special meaning because of facts known only to the speakers. A subset of jargon codes is cue codes, where certain prearranged phrases convey meaning.
- Covered or concealment ciphers hide a message openly in the carrier medium so that it can be recovered by anyone who knows the secret for how it was concealed. A grille cipher employs a template that is used to cover the carrier message. The words that appear in the openings of the template are the hidden message. A null cipher hides the message according to some prearranged set of rules, such as "read every fifth word" or "look at the third character in every word."

As an increasing amount of data is stored on computers and transmitted over networks, it is not surprising that steganography has entered the digital age. On computers and networks, steganography applications allow for someone to hide any type of binary file in any other binary file, although image and audio files are today's most common carriers.

Steganography provides some very useful and commercially important functions in the digital world, most notably digital watermarking. In this application, an author can embed a hidden message in a file so that ownership of intellectual property can later be asserted and/or to ensure the integrity of the content. An artist, for example, could post original artwork on a Website. If someone else steals the file and claims the work as his or her own, the artist can later prove ownership because only he/she can recover the watermark (Arnold et al. 2003; Barni et al. 2001; Kwok 2003). Although conceptually similar to steganography, digital watermarking usually has different technical goals. Generally only a small amount of repetitive information is inserted into the carrier, it is not necessary to hide the watermarking information, and it is useful for the watermark to be able to be removed while maintaining the integrity of the carrier.

Steganography has a number of nefarious applications; most notably hiding records of illegal activity, financial fraud, industrial espionage, and communication among members of criminal or terrorist organizations (Hosmer and Hyde 2003).

Null Ciphers

Historically, null ciphers are a way to hide a message in another without the use of a complicated algorithm. One of the simplest null ciphers is shown in the classic examples below:

PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT IMMENSELY.

APPARENTLY NEUTRAL'S PROTEST IS THOROUGHLY DISCOUNTED AND IGNORED. ISMAN HARD HIT. BLOCKADE ISSUE AFFECTS PRETEXT FOR EMBARGO ON BYPRODUCTS, EJECTING SUETS AND VEGETABLE OILS.

The German Embassy in Washington, DC, sent these messages in telegrams to their headquarters in Berlin during World War I (Kahn 1996). Reading the first character of every word in the first message or the second character of every word in the second message will yield the following hidden text:

PERSHING SAILS FROM N.Y. JUNE 1

On the Internet, spam is a potential carrier medium for hidden messages. Consider the following:

Dear Friend , This letter was specially selected to be sent to you ! We will comply with all removal requests ! This mail is being sent in compliance with Senate bill 1621 ; Title 5 ; Section 303 ! Do NOT confuse us with Internet scam artists . Why work for somebody else when you can become rich within 38 days ! Have you ever noticed the baby boomers are more demanding than their parents & more people than ever are surfing the web ! Well, now is your chance to capitalize on this ! WE will help YOU sell more & SELL MORE . You can begin at absolutely no cost to you ! But don't believe us ! Ms Anderson who resides in Missouri tried us and says "My only problem now is where to park all my cars" . This offer is 100% legal . You will blame yourself forever if you don't order now ! Sign up a friend and your friend will be rich too . Cheers ! Dear Salaryman , Especially for you - this amazing news . If you are not interested in our publications and wish to be removed from our lists, simply do NOT respond and ignore this mail ! This mail is being sent in compliance with Senate bill 2116 , Title 3 ; Section 306 ! This is a legitimate business proposal ! Why work for somebody else when you can become rich within 68 months ! Have you ever noticed more people than ever are surfing the web and nobody is getting any younger ! Well, now is your chance to capitalize on this . We will help you decrease perceived waiting time by 180% and SELL MORE . The best thing about our system is that it is absolutely risk free for you ! But don't believe us ! Mrs Ames of Alabama tried us and says "My only problem now is where to park all my cars" . We are licensed to operate in all states ! You will blame yourself forever if you don't order now ! Sign up a friend and you'll get a discount of 20% ! Thanks ! Dear Salaryman , Your email address has been submitted to us indicating your interest in our briefing ! If you no longer wish to receive our publications simply reply with a Subject: of "REMOVE" and you will immediately be removed from our mailing list . This mail is being sent in compliance with Senate bill 1618 , Title 6 , Section 307 . THIS IS NOT A GET RICH SCHEME . Why work for somebody else when you can become rich within 17 DAYS ! Have you ever noticed more people than ever are surfing the web and more people than ever are surfing the web ! Well, now is your chance to capitalize on this ! WE will help YOU turn your business into an E-BUSINESS and deliver goods right to the customer's doorstep ! You are guaranteed to succeed because we take all the risk ! But don't believe us . Ms Simpson of Wyoming tried us and says "Now I'm rich, Rich, RICH" ! We assure you that we operate within all applicable laws . We implore you - act now ! Sign up a friend and you'll get a discount of 50% . Thank-you for your serious consideration of our offer .

This message looks like typical spam, which is generally ignored and discarded. This message was created at spam mimic, a Website that converts a short text message into a text block that looks like spam using a grammar-based mimicry idea first proposed by Peter Wayner (spam mimic 2003; Wayner 2002). The reader will learn nothing by looking at the word spacing or misspellings in the message; the zeros and ones are encoded by the choice of the words. The hidden message in the spam carrier above is:

Meet at Main and Willard at 8:30

Special tools or skills to hide messages in digital files using variances of a null cipher are not necessary. An image or text block can be hidden under another image in a PowerPoint file, for example. Messages can be hidden in the properties of a Word file. Messages can be hidden in comments in Web pages or in other formatting vagaries that are ignored by browsers (Artz 2001). Text can be hidden as line art in a document by putting the text in the same color as the background and placing another drawing in the foreground. The recipient could retrieve the hidden text by changing its color (Seward 2004). These are all decidedly low-tech mechanisms, but they can be very effective.

Digital Image and Audio

Many common digital steganography techniques employ graphical images or audio files as the carrier medium. It is instructive, then, to review image and audio encoding before discussing how steganography and steganalysis works with these carriers.

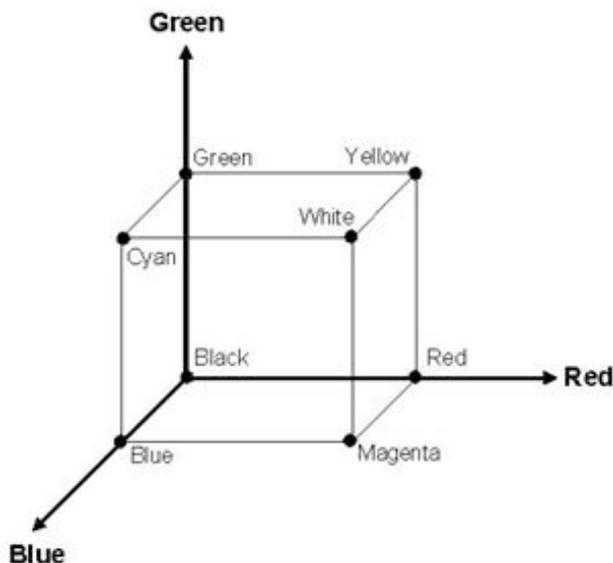


Figure 2. The RGB Color Cube.

Figure 2 shows the RGB color cube, a common means with which to represent a given color by the relative intensity of its three component colors—red, green, and blue—each with their own axis (moreCrayons 2003). The absence of all colors yields black, shown as the intersection of the zero point of the

three-color axes. The mixture of 100 percent red, 100 percent blue, and the absence of green form magenta; cyan is 100 percent green and 100 percent blue without any red; and 100 percent green and 100 percent red with no blue combine to form yellow. White is the presence of all three colors.

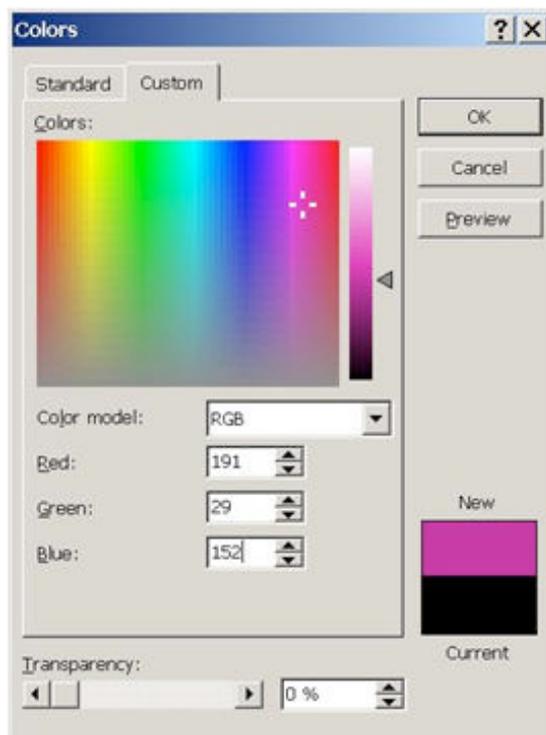


Figure 3. This color selection dialogue box shows the red, green, and blue (RGB) levels of this selected color.

Figure 3 shows the RGB intensity levels of some random color. Each RGB component is specified by a single byte, so that the values for each color intensity can vary from 0-255. This particular shade is denoted by a red level of 191 (hex BF), a green level of 29 (hex 1D), and a blue level of 152 (hex 98). One pix of magenta, then, would be encoded using 24 bits, as 0xBF1D98. This 24-bit encoding scheme supports 16,777,216 (2^{24}) unique colors (Curran and Bailey 2003; Johnson and Jajodia 1998A).

Most digital image applications today support 24-bit true color, where each picture element (pixel) is encoded in 24 bits, comprising the three RGB bytes as described above. Other applications encode color using eight bits/pix. These schemes also use 24-bit true color but employ a palette that specifies which colors are used in the image. Each pix is encoded in eight bits, where the value points to a 24-bit color entry in the palette. This method limits the unique number of colors in a given image to 256 (2^8).

The choice color encoding obviously affects image size. A 640 X 480 pixel image using eight-bit color would occupy approximately 307 KB (640 x 480 = 307,200 bytes), whereas a 1400 X 1050 pix image using 24-bit true color would require 4.4 MB (1400 x 1050 x 3 = 4,410,000 bytes).

Color palettes and eight-bit color are commonly used with Graphics Interchange Format (GIF) and Bitmap (BMP) image formats. GIF and BMP are generally considered to offer lossless compression because the image recovered after encoding and compression is bit-for-bit identical to the original image (Johnson and Jajodia 1998A).

The Joint Photographic Experts Group (JPEG) image format uses discrete cosine transforms rather than a pix-by-pix encoding. In JPEG, the image is divided into 8 X 8 blocks for each separate color component. The goal is to find blocks where the amount of change in the pixel values (the energy) is low. If the energy level is too high, the block is subdivided into 8 X 8 subblocks until the energy level is low enough. Each 8 X 8 block (or subblock) is transformed into 64 discrete cosine transforms coefficients that approximate the luminance (brightness, darkness, and contrast) and chrominance (color) of that portion of the image. JPEG is generally considered to be lossy compression because the image recovered from the compressed JPEG file is a close approximation of, but not identical to, the original (Johnson and Jajodia 1998A; Monash University 2004; Provos and Honeyman 2003).

Audio encoding involves converting an analog signal to a bit stream. Analog sound-voice and music-is represented by sine waves of different frequencies. The human ear can hear frequencies nominally in the range of 20-20,000 cycles/second (Hertz or Hz). Sound is analog, meaning that it is a continuous signal. Storing the sound digitally requires that the continuous sound wave be converted to a set of samples that can be represented by a sequence of zeros and ones.

Analog-to-digital conversion is accomplished by sampling the analog signal (with a microphone or other audio detector) and converting those samples to voltage levels. The voltage or signal level is then converted to a numeric value using a scheme called pulse code modulation. The device that performs this conversion is called a coder-decoder or codec.

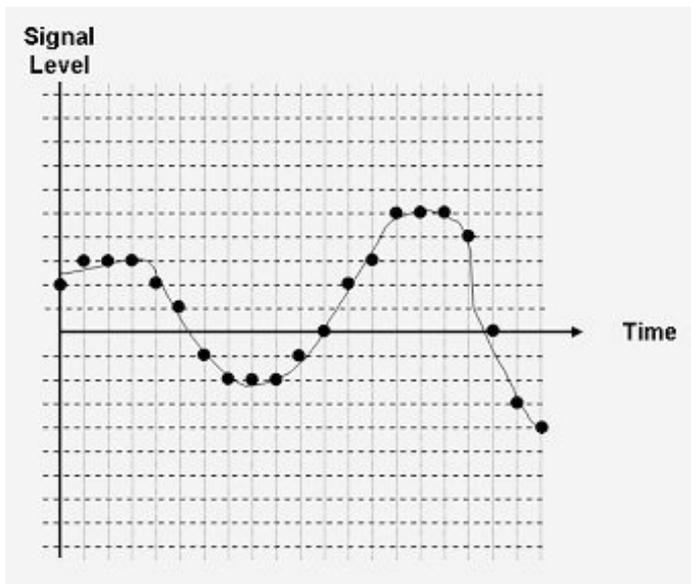


Figure 4. Simple Pulse Code Modulation.

Pulse code modulation provides only an approximation of the original analog signal, as shown in Figure 4. If the analog sound level is measured at a 4.86 level, for example, it would be converted to a five in pulse code modulation. This is called quantization error. Different audio applications define a different number of pulse code modulation levels so that this "error" is nearly undetectable by the human ear. The telephone network converts each voice sample to an eight-bit value (0-255) whereas music applications generally use 16-bit values (0-65,535) (Fries and Fries 2000; Rey 1983).

Analog signals need to be sampled at a rate of twice the highest frequency component of the signal so that the original can be correctly reproduced from the samples alone. In the telephone network, the human voice is carried in a frequency band 0-4000 Hz (although only about 400-3400 Hz is actually used to carry voice); therefore, voice is sampled 8,000 times per second (an 8 kHz sampling rate). Music audio applications assume the full spectrum of the human ear and generally use a 44.1 kHz sampling rate (Fries and Fries 2000; Rey 1983).

The bit rate of uncompressed music can be easily calculated from the sampling rate (44.1 kHz), pulse code modulation resolution (16 bits), and number of sound channels (two) to be 1,411,200 bits per second. This would suggest that a one-minute audio file (uncompressed) would occupy 10.6 MB (1,411,200*60/8 = 10,584,000). Audio files are, in fact, made smaller by using a variety of compression techniques. One obvious method is to reduce the number of channels to one or to reduce the sampling rate, in some cases as low as 11 kHz. Other codecs use proprietary compression schemes. All of these solutions reduce the quality of the sound.

Table 1: Some Common Digital Audio Formats (Fries and Fries 2000)

Audio Type	File Extension	Codec
AIFF (Mac)	.aif, .aiff	Pulse code modulation (or other)
AU (Sun/Next)	.au	μ-law (or other)
CD audio (CDDA)	n/a	Pulse code modulation
MP3	.mp3	MPEG Audio Layer III
Windows Media Audio	.wma	Microsoft proprietary
QuickTime	.qt	Apple Computer proprietary
RealAudio	.ra, .ram	Real Networks proprietary
WAV	.wav	Pulse code modulation (or other)

Digital Carrier Methods

There are many ways in which messages can be hidden in digital media. Digital forensics examiners are familiar with data that remains in file slack or unallocated space as the remnants of previous files, and programs can be written to access slack and unallocated space directly. Small amounts of data can also be hidden in the unused portion of file headers (Curran and Bailey 2003).

Information can also be hidden on a hard drive in a secret partition. A hidden partition will not be seen under normal circumstances, although disk configuration and other tools might allow complete access to the hidden partition (Johnson et al. 2001). This theory has been implemented in a steganographic ext2fs file system for Linux. A hidden file system is particularly interesting because it protects the user from being inextricably tied to certain information on their hard drive. This form of plausible deniability allows a user to claim to not be in possession of certain information or to claim that certain events never occurred. Under this system users can hide the number of files on the drive, guarantee the secrecy of the files' contents, and not disrupt nonhidden files by the removal of the steganography file driver (Anderson et al. 1998; Artz 2001; McDonald and Kuhn 2000).

Another digital carrier can be the network protocols. Covert Transmission Control Protocol by Craig Rowland, for example, forms covert communications channels using the Identification field in Internet Protocol packets or the sequence number field in Transmission Control Protocol segments (Johnson et al. 2001; Rowland 1996).

There are several characteristics of sound that can be altered in ways that are indiscernible to human senses, and these slight alterations, such as tiny shifts in phase angle, speech cadence, and frequency, can transport hidden information (Curran and Bailey 2003).

Nevertheless, image and audio files remain the easiest and most common carrier media on the Internet because of the plethora of potential carrier files already in existence, the ability to create an infinite number of new carrier files, and the easy access to steganography software that will operate on these carriers. For that reason, the manuscript focus will return to image and audio files.

The most common steganography method in audio and image files employs some type of least significant bit substitution or overwriting. The least significant bit term comes from the numeric significance of the bits in a byte. The high-order or most significant bit is the one with the highest arithmetic value (i.e., $2^7=128$), whereas the low-order or least significant bit is the one with the lowest arithmetic value (i.e., $2^0=1$).

As a simple example of least significant bit substitution, imagine "hiding" the character 'G' across the following eight bytes of a carrier file (the least significant bits are underlined):

```
10010101 00001101 11001001 10010110
00001111 11001011 10011111 00010000
```

A 'G' is represented in the American Standard Code for Information Interchange (ASCII) as the binary string 01000111. These eight bits can be "written" to the least significant bit of each of the eight carrier bytes as follows:

```
10010100 00001101 11001000 10010110
00001110 11001011 10011111 00010001
```

In the sample above, only half of the least significant bits were actually changed (shown above in italics). This makes some sense when one set of zeros and ones are being substituted with another set of zeros and ones.

Least significant bit substitution can be used to overwrite legitimate RGB color encodings or palette pointers in GIF and BMP files, coefficients in JPEG files, and pulse code modulation levels in audio files. By overwriting the least significant bit, the numeric value of the byte changes very little and is least likely to be detected by the human eye or ear.

Least significant bit substitution is a simple, albeit common, technique for steganography. Its use, however, is not necessarily as simplistic as the method sounds. Only the most naive steganography software would merely overwrite every least significant bit with hidden data. Almost all use some sort of means to randomize the actual bits in the carrier file that are modified. This is one of the factors that makes steganography detection so difficult.

One other way to hide information in a paletted image is to alter the order of the colors in the palette or use least significant bit encoding on the palette colors rather than on the image data. These methods are potentially weak, however. Many graphics software tools order the palette colors by frequency, luminance, or other parameter, and a randomly ordered palette stands out under statistical analysis (Fridrich and Du 2000).

Newer, more complex steganography methods continue to emerge. Spread-spectrum steganography methods are analogous to spread-spectrum radio transmissions (developed in World War II and commonly used in data communications systems today) where the "energy" of the signal is spread across a wide-frequency spectrum rather than focused on a single frequency, in an effort to make detection and jamming of the signal harder. Spread-spectrum steganography has the same function-avoid detection. These methods take advantage of the fact that little distortions to image and sound files are least detectable in the high-energy portions of the carrier (i.e., high intensity in sound files or bright colors in image files). Even when viewed side by side, it is easier to fool human senses when small changes are made to loud sounds and/or bright colors (Wayner 2002).

Steganography Examples

There are more than 100 steganography programs currently available, ranging from free downloads to commercial products. This section will show some simple steganography examples by hiding an 11,067-byte GIF map of the Burlington, Vermont, airport (Figure 5) in GIF, JPEG, and WAV files.



Figure 5. This map is hidden in the various carriers in this article.



Figure 6. A GIF carrier file containing the airport map.

The first example employs Gif-It-Up, a Nelsonsoft program that hides information in GIF files using least significant bit substitution (and includes an encryption option). Figure 6 shows a GIF image of the Washington, DC, mall at night where Gif-It-Up has been used to insert the airport map shown in Figure 5. The original carrier is 632,778 bytes in length and uses 249 unique colors, whereas the steganography file is 677,733 bytes in length and uses 256 unique colors. The file size is larger in the steganography file because of a color extension option used to minimize distortion in the steganography image. If color extension is not employed, the file size differences are slightly less noticeable.





Figure 7. The palette from the Washington mall carrier file before (left) and after (right) the map file was hidden.

Figure 7 shows the carrier file's palettes before and after message insertion. Like all least significant bit insertion programs that act on eight-bit color images, Gif-It-Up modifies the color palette and generally ends up with many duplicate color pairs.

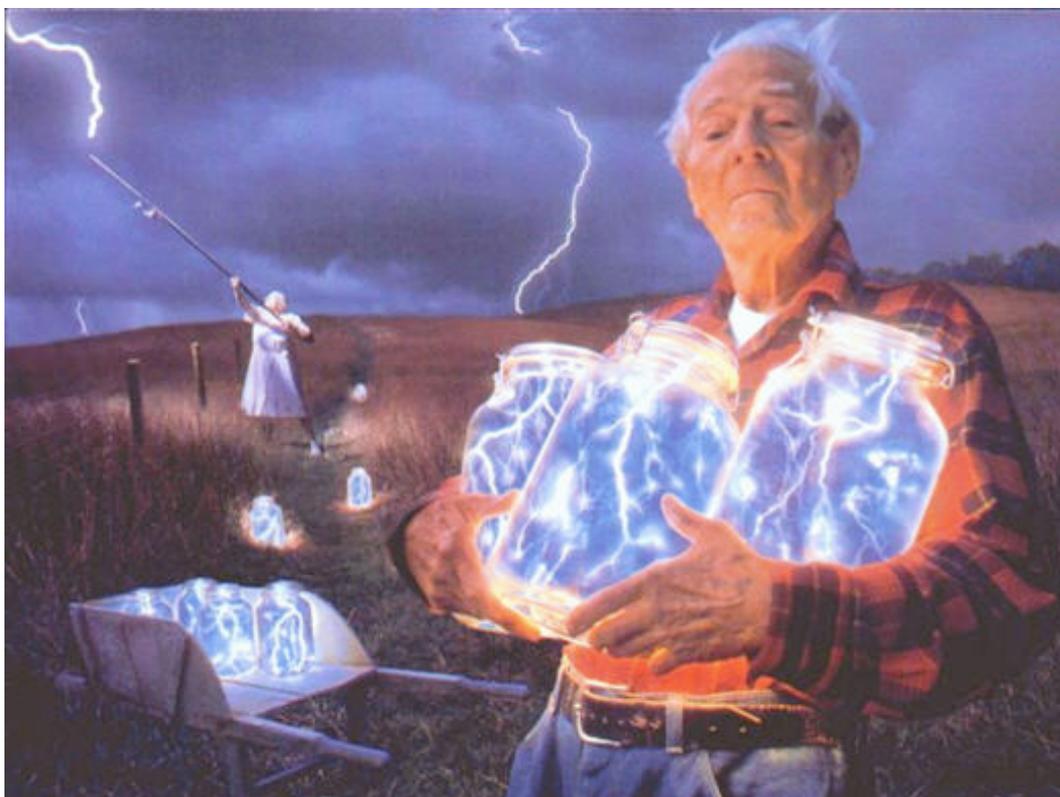


Figure 8. A JPEG carrier file containing the airport map.

JP Hide-&-Seek (JPHS) by Allan Latham is designed to be used with JPEG files and lossy compression. JPHS uses least significant bit overwriting of the discrete cosine transform coefficients used by the JPEG algorithm. The Blowfish crypto algorithm is used for least significant bit randomization and encryption (Johnson and Jajodia 1998B). Figure 8 shows an example JPEG file with the airport map embedded in it. The original carrier file is 207,244 bytes in size and contains 224,274 unique colors. The steganography file is 207,275 bytes in size and contains 227,870 unique colors. There is no color palette to look at because JPEG uses 24-bit color coding and discrete cosine transforms.

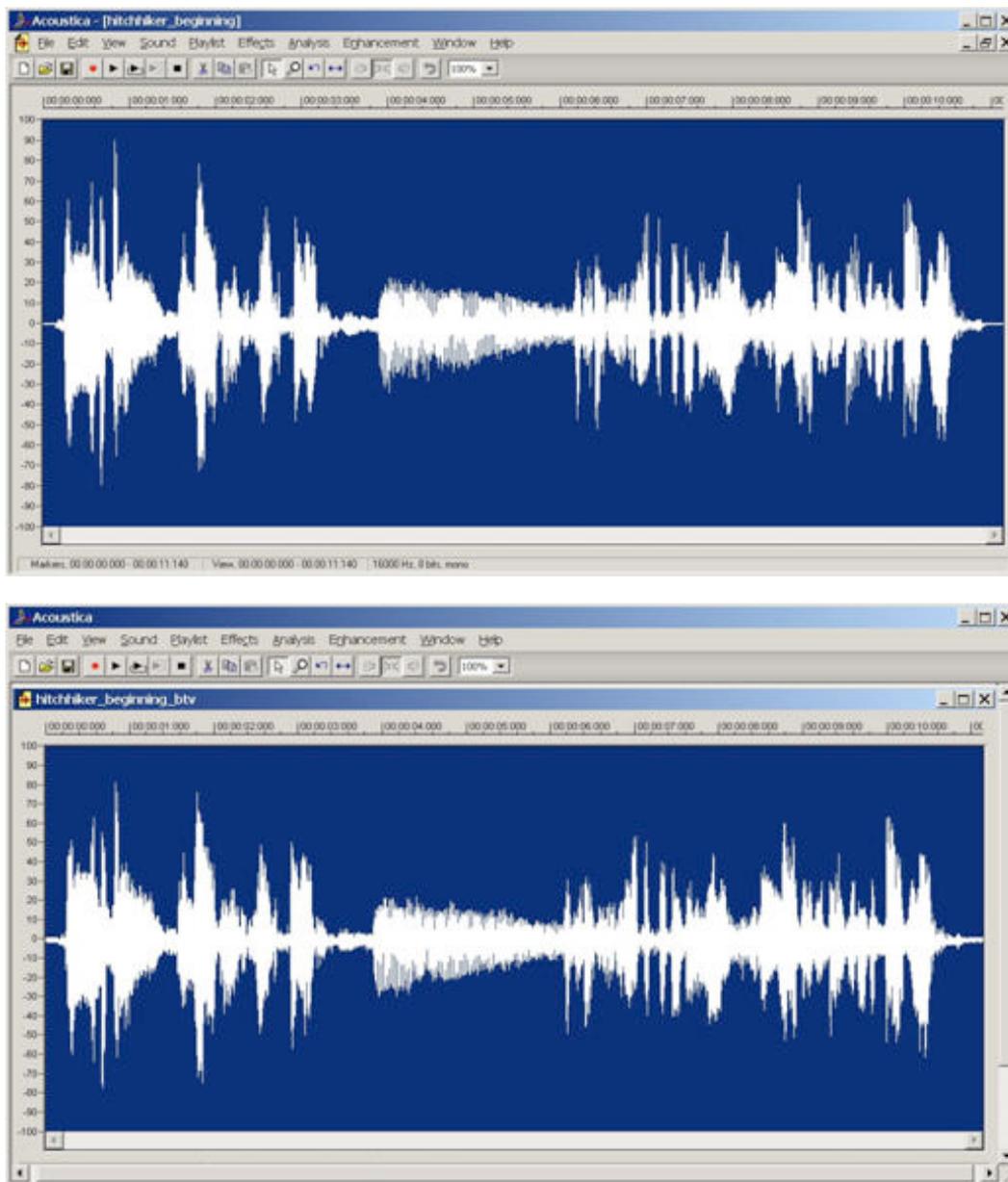


Figure 9. The signal level comparisons between a WAV carrier file before (above) and after (below) the airport map is inserted.

The final example employs S-Tools, a program by Andy Brown that can hide information inside GIF, BMP, and WAV files. S-Tools uses least significant bit substitution in files that employ lossless compression, such as eight- or 24-bit color and pulse code modulation. S-Tools employs a password for least significant bit randomization and can encrypt data using the Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), Message Digest Cipher (MDC), or Triple-DES (Johnson and Jajodia 1998A; Johnson and Jajodia 1998B; Wayner 2002). Figure 9 shows a signal level comparison between a WAV carrier file before and after the airport map was hidden. The original WAV file is 178,544 bytes in length, whereas the steganography WAV file is 178,298 bytes in length. Although the relatively small size of the figure makes it hard to see details, some differences are noticeable at the beginning and end of the audio sample (i.e., during periods of silence). (Some steganography tools have built-in intelligence to avoid the low-intensity portions of the signal.) Audio files are well suited to information hiding because they are usually relatively large, making it difficult to find small hidden items.

Gif-It-Up, JPHS, and S-Tools are used above for example purposes only. They are free, easy to use, and perform their tasks well. There are many other programs that can be used to hide information in BMP, GIF, JPEG, MP3, Paintbrush (PCX), Portable Network Graphics (PNG), Tag Image File Format (TIFF), WAV, and other carrier file types. The StegoArchive.Com Website has a very good list of freeware, shareware, and commercial steganography software for DOS, Linux/Unix, MacOS, Windows, and other operating systems (StegoArchive.com 2003).

Although the discussion above has focused only on image and audio files, steganography media are not limited to these types of files. Other file types also have characteristics that can be exploited for information hiding. Hydan, for example, can conceal text messages in OpenBSD, FreeBSD, NetBSD, Red Hat Linux, and Windows XP executable files. Developed by Rakan El-Khalil, Hydan takes advantage of redundancy in the i386 instruction set and inserts hidden information by defining sets of functionally equivalent instructions, conceptually like a grammar-based mimicry (e.g., where ADD instructions are a zero bit and SUB instructions are a one bit). The program can hide approximately one message byte in every 110-instruction bytes and maintains the original size of the application file. Blowfish encryption can also be employed (El-Khalil 2003).

Detecting Steganography

The Prisoner's Problem (Simmons 1983) is often used to describe steganography, although it was originally introduced to describe a cryptography scenario. The problem involves two prisoners, Alice and Bob, who are locked in separate prison cells and wish to communicate some secret plan to each other. Alice and Bob are allowed to exchange messages with each other, but William, the warden, can read all of the messages. Alice and Bob know that William will terminate the communications if he discovers the secret channel (Chandramouli 2002; Fridrich et al. 2003B).

William can act in either a passive or active mode. In the passive warden model, William examines each message and determines whether to forward the message or not based on his ability to detect a hidden message. In the active warden model, William can modify messages if he wishes. A conservative or malicious warden might actually modify all messages in an attempt to disrupt any covert channel so that Alice and Bob would need to use a very robust steganography method (Chandramouli 2002; Fridrich et al. 2003B).

The difficulty of the warden's task will depend largely on the complexity of the steganography algorithm and the amount of William's prior knowledge (Chandramouli 2002; Fridrich et al. 2003B; Provos and Honeyman 2003).

In a pure steganography model, William knows nothing about the steganography method employed by Alice and Bob. This is a poor assumption on Alice and Bob's part since security through obscurity rarely works and is particularly disastrous when applied to cryptography. This is, however, often the model of the digital forensics analyst searching a Website or hard drive for the possible use of steganography.

Secret key steganography assumes that William knows the steganography algorithm but does not know the secret stego/crypto key employed by Alice and Bob. This is consistent with the assumption that a user of cryptography should make, per Kerckhoff's Principle (i.e., "the security of the crypto scheme is in key management, not secrecy of the algorithm.") (Kahn 1996). This may also be too strong of an assumption for practice, however, because complete information would include access to the carrier file source.

Steganalysis, the detection of steganography by a third party, is a relatively young research discipline with few articles appearing before the late-1990s. The art and science of steganalysis is intended to detect or estimate hidden information based on observing some data transfer and making no assumptions about the steganography algorithm (Chandramouli 2002). Detection of hidden data may not be sufficient. The steganalyst may also want to extract the hidden message, disable the hidden message so that the recipient cannot extract it, and/or alter the hidden message to send misinformation to the recipient (Jackson et al. 2003). Steganography detection and extraction is generally sufficient if the purpose is evidence gathering related to a past crime, although destruction and/or alteration of the hidden information might also be legitimate law enforcement goals during an on-going investigation of criminal or terrorist groups.

Steganalysis techniques can be classified in a similar way as cryptanalysis methods, largely based on how much prior information is known (Curran and Bailey 2003; Johnson and Jajodia 1998B).

- Steganography-only attack: The steganography medium is the only item available for analysis.
- Known-carrier attack: The carrier and steganography media are both available for analysis.
- Known-message attack: The hidden message is known.
- Chosen-steganography attack: The steganography medium and algorithm are both known.
- Chosen-message attack: A known message and steganography algorithm are used to create steganography media for future analysis and comparison.
- Known-steganography attack: The carrier and steganography medium, as well as the steganography algorithm, are known.

Steganography methods for digital media can be broadly classified as operating in the image domain or transform domain. Image domain tools hide the message in the carrier by some sort of bit-by-bit manipulation, such as least significant bit insertion. Transform domain tools manipulate the steganography algorithm and the actual transformations employed in hiding the information, such as the discrete cosine transforms coefficients in JPEG images (Johnson and Jajodia 1998B).

It follows, then, that steganalysis broadly follows the way in which the steganography algorithm works. One simple approach is to visually inspect the carrier and steganography media. Many simple steganography tools work in the image domain and choose message bits in the carrier independently of the content of the carrier. Although it is easier to hide the message in the area of brighter color or louder sound, the program may not seek those areas out. Thus, visual inspection may be sufficient to cast suspicion on a steganography medium (Wayner 2002).

A second approach is to look for structural oddities that suggest manipulation. Least significant bit insertion in a palette-based image often causes a large number of duplicate colors, where identical (or nearly identical) colors appear twice in the palette and differ only in the least significant bit. Steganography programs that hide information merely by manipulating the order of colors in the palette cause structural changes, as well. The structural changes often create a signature of the steganography algorithm that was employed (Jackson et al. 2003; Wayner 2002).

Steganographic techniques generally alter the statistics of the carrier and, obviously, longer hidden messages will alter the carrier more than shorter ones (Farid 2001; Fridrich and Du 2000; Fridrich and Goljan 2002; Ozer et al. 2003). Statistical analysis is commonly employed to detect hidden messages, particularly when the analyst is working in the blind (Jackson et al. 2003). There is a large body of work in the area of statistical steganalysis.

Statistical analysis of image and audio files can show whether the statistical properties of the files deviate from the expected norm (Farid 2001; Ozer et al. 2003; Provos and Honeyman 2001). These so-called first-order statistics—means, variances, chi-square (χ^2) tests—can measure the amount of redundant information and/or distortion in the medium. Although these measures can yield a prediction as to whether the contents have been modified or seem suspicious, they are not definitive (Wayner 2002).

Statistical steganalysis is made harder because some steganography algorithms take pains to preserve the carrier file's first-order statistics to avoid just this type of detection. Encrypting the hidden message also makes detection harder because encrypted data generally has a high degree of randomness, and ones and zeros appear with equal likelihood (Farid 2001; Provos and Honeyman 2001).

Recovery of the hidden message adds another layer of complexity compared to merely detecting the presence of a hidden message. Recovering the message requires knowledge or an estimate of the message length and, possibly, an encryption key and knowledge of the crypto algorithm (Fridrich et al. 2003B).

Carrier file type-specific algorithms can make the analysis more straightforward. JPEG, in particular, has received a lot of research attention because of the way in which different algorithms operate on this type of file. JPEG is a poor carrier medium when using simple least significant bit insertion because the modification to the file caused by JPEG compression eases the task of detecting the hidden information (Fridrich and Du 2000). There are several algorithms that hide information in JPEG files, and all work differently. JSteg sequentially embeds the hidden data in least significant bits, JP Hide&Seek uses a random process to select least significant bits, F5 uses a matrix encoding based on a Hamming code, and OutGuess preserves first-order statistics (Fridrich et al. 2001; Fridrich et al. 2002A; Fridrich et al. 2002B; Fridrich et al. 2003A; Provos and Honeyman 2001; Provos and Honeyman 2003).

More advanced statistical tests using higher-order statistics, linear analysis, Markov random fields, wavelet statistics, and more on image and audio files have been described (Farid 2001; Farid and Lyu 2003; Fridrich and Goljan 2002; Ozer et al. 2003). Detailed discussion is beyond the scope of this paper, but the results of this research can be seen in some steganography detection tools.

Most steganalysis today is signature-based, similar to antivirus and intrusion detection systems. Anomaly-based steganalysis systems are just beginning to emerge. Although the former systems are accurate and robust, the latter will be more flexible and better able to quickly respond to new steganography techniques. One form of so-called "blind steganography detection" distinguishes between clean and steganography images using statistics based on wavelet decomposition, or the examination of space, orientation, and scale across subsets of the larger image (Farid 2001; Jackson et al. 2003).

This type of statistical steganalysis is not limited to image and audio files. The Hydan program retains the size of the original carrier but, by using sets of

"functionally equivalent" instructions, employs some instructions that are not commonly used. This opens Hydan to detection when examining the statistical distribution of a program's instructions. Future versions of Hydan will maintain the integrity of the statistical profile of the original application to defend against this analysis (El-Khalil 2003).

The law enforcement community does not always have the luxury of knowing when and where steganography has been used or the algorithm that has been employed. Generic tools that can detect and classify steganography are where research is still in its infancy but are already becoming available in software tools, some of which are described in the next section (McCullagh 2001).

And the same cycle is recurring as seen in the crypto world—steganalysis helps find embedded steganography but also shows writers of new steganography algorithms how to avoid detection.

Tools for Steganography Detection

This article has a stated focus on the practicing computer forensics examiner rather than the researcher. This section, then, will show some examples of currently available software that can detect the presence of steganography programs, detect suspect carrier files, and disrupt steganographically hidden messages. This is by no means a survey of all available tools, but an example of available capabilities. StegoArchive.com lists many steganalysis programs (StegoArchive.com 2003).

The detection of steganography software on a suspect computer is important to the subsequent forensic analysis. As the research shows, many steganography detection programs work best when there are clues as to the type of steganography that was employed in the first place. Finding steganography software on a computer would give rise to the suspicion that there are actually steganography files with hidden messages on the suspect computer. Furthermore, the type of steganography software found will directly impact any subsequent steganalysis (e.g., S-Tools might direct attention to GIF, BMP, and WAV files, whereas JP Hide-&-Seek might direct the analyst to look more closely at JPEG files).

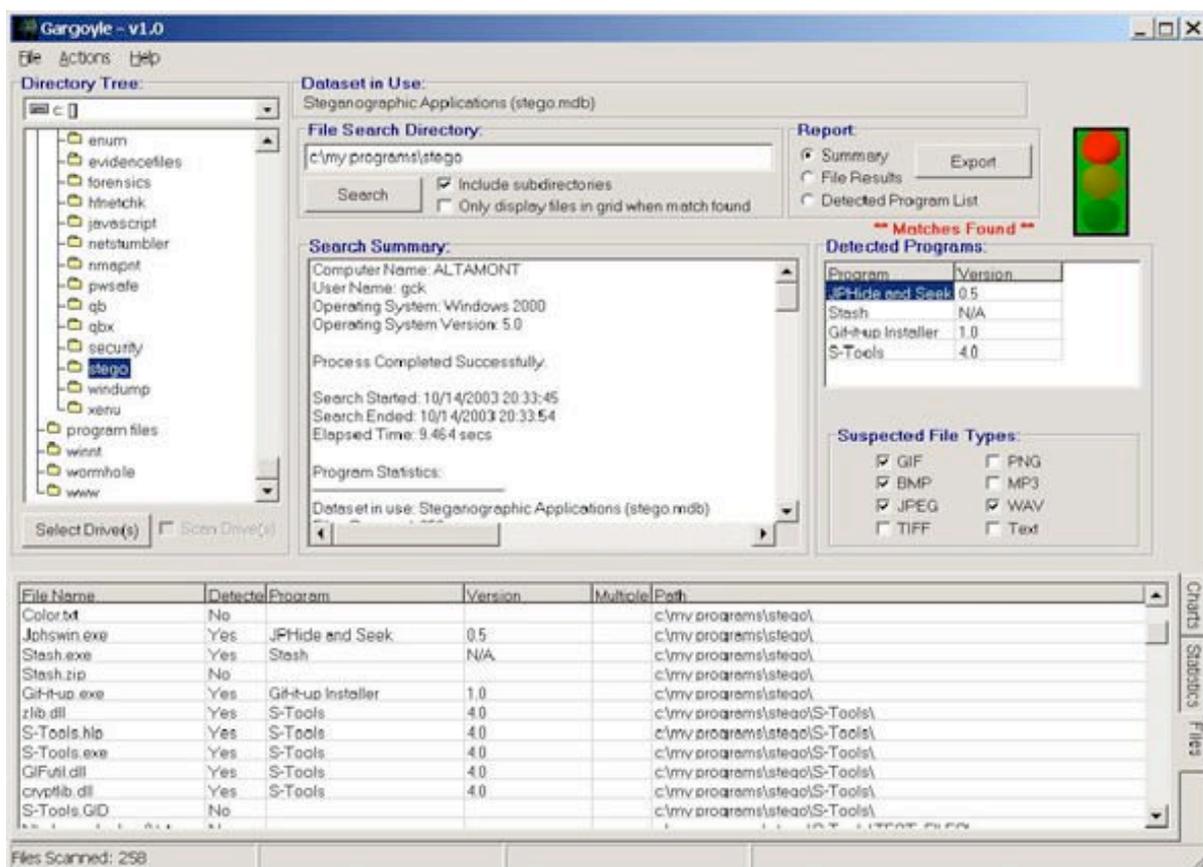


Figure 10. The output from Gargoyles when aimed at one of the directories on a hard drive.

WetStone Technologies' Gargoyles (formerly StegoDetect) software (WetStone Technologies 2004A) can be used to detect the presence of steganography software. Gargoyles employs a proprietary data set (or hash set) of all of the files in the known steganography software distributions, comparing them to the hashes of the files subject to search. Figure 10 shows the output when Gargoyles was aimed at a directory where steganography programs are stored. Gargoyles data sets can also be used to detect the presence of cryptography, instant messaging, key logging, Trojan horse, password cracking, and other nefarious software.

AccessData's Forensic Toolkit (AccessData 2003) and Guidance Software's EnCase (Guidance Software 2003) can use the HashKeeper (Hashkeeper 2003), Maresware (Maresware 2003), and National Software Reference Library (National Software Reference Library 2003) hash sets to look for a large variety of software. In general, these data sets are designed to exclude hashes of known "good" files from search indexes during the computer forensic analysis. Gargoyles can also import these hash sets.

The detection of steganography software continues to become harder for another reason—the small size of the software coupled with the increasing storage capacity of removable media. S-Tools, for example, requires less than 600 KB of disk space and can be executed directly, without additional installation, from a floppy or USB memory key. Under those circumstances, no remnants of the program would be found on the hard drive.

The second important function of steganography detection software is to find possible carrier files. Ideally, the detection software would also provide some clues as to the steganography algorithm used to hide information in the suspect file so that the analyst might be able to attempt recovery of the hidden information.

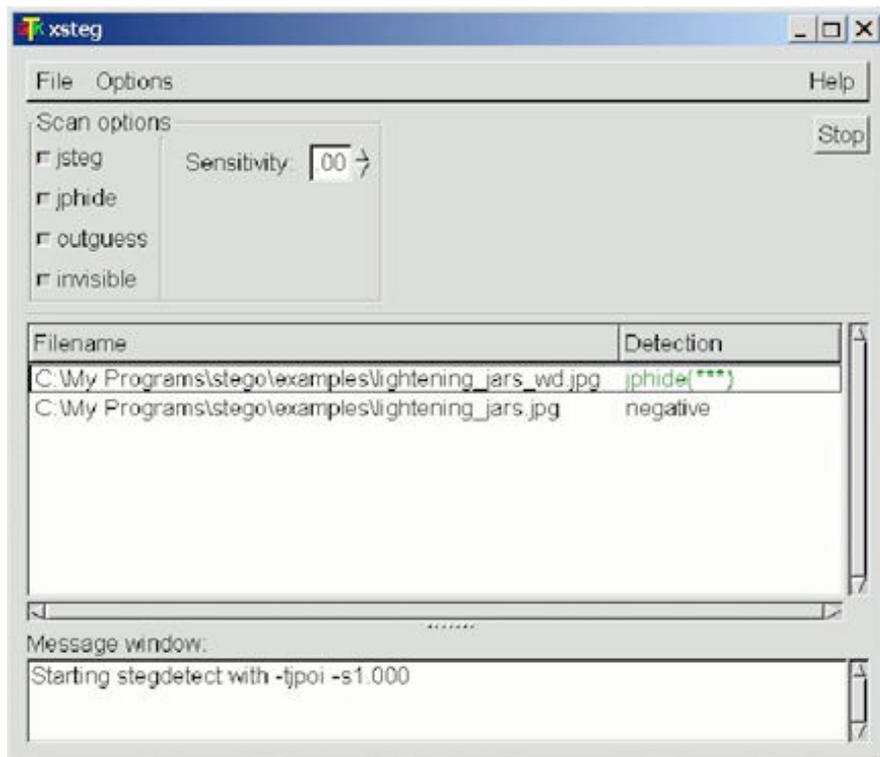


Figure 11. The output from Xsteg when examining two suspect JPEG files.

One commonly used detection program is Niels Provos' stegdetect. Stegdetect can find hidden information in JPEG images using such steganography schemes as F5, Invisible Secrets, JPHide, and JSteg (OutGuess 2003). Figure 11 shows the output from xsteg, a graphical interface for stegdetect, when used to examine two files on a hard drive—the original carrier and steganography image for the JPEG image shown in Figure 8. Note that the steganography file is not only flagged as containing hidden information, but the program also suggests (correctly) the used of the JPHide steganography scheme.

WetStone Technologies' Stego Watch (WetStone Technologies 2004B) analyzes a set of files and provides a probability about which are steganography media and the likely algorithm used for the hiding (which, in turn, provides clues as to the most likely software employed). The analysis uses a variety of user-selectable statistical tests based on the carrier file characteristics that might be altered by the different steganography methods. Knowing the steganography software that is available on the suspect computer will help the analyst select the most likely statistical tests.



Figure 12. Information from Stego Watch about a JPEG file suspected to be a steganography carrier.

Figure 12 shows the output from Stego Watch when aimed at the JPEG carrier file shown in Figure 8. The Steganography Detection Algorithms section in the display show the statistical algorithms employed for analysis and the ones that bore fruit for this image. As above, Stego Watch correctly identifies the JPEG steganography software that was employed.

Although not yet available, the Institute for Security Technology Studies at Dartmouth College has developed software capable of detecting hidden data in image files using statistical models that are independent of the image format or steganography technique. This program has been tested on 1,800 images and four different steganography algorithms and was able to detect the presence of hidden messages with 65 percent accuracy with a false-positive rate less than 0.001 percent (Dartmouth College 2003).

Finding steganography in a file suspected to contain it is relatively easy compared to extracting hidden data. Most steganography software uses passwords for secrecy, randomization, and/or encryption. Stegbreak, a companion program to stegdetect, uses a dictionary attack against JSteg-Shell, JPHide, and OutGuess to find the password of the hidden data but, again, this is only applicable to JPEG files (OutGuess 2003). Similarly, Stego Break is a companion program to WetStone's Stego Watch that uses a dictionary attack on suspect files (WetStone Technologies 2004B). Steganography detection schemes do not directly help in the recovery of the password. Finding appropriate clues is where the rest of the investigation and computer forensics comes into play.

A computer forensics examiner looking at evidence in a criminal case probably has no reason to alter any evidence files. However, an examination that is part of an ongoing terrorist surveillance might well want to disrupt the hidden information even if it cannot be recovered. Hidden content, such as steganography and digital watermarks, can be attacked in several ways so that it can be removed or altered (Hernandez Martin and Kutter 2001; Voloshynovskiy et al. 2001), and there is software specifically designed to attack digital watermarks. Such attacks have one of two possible effects—they either reduce the steganography carrying capacity of the carrier (necessary to avoid the attack) or fully disable the capability of the carrier as a steganography medium.

Although this subject is also beyond the scope of this paper, one interesting example of steganography disruption software can be used to close this discussion. 2Mosaic by Fabien Petitcolas employs a so-called "presentation attack" primarily against images on a Website. 2Mosaic attacks a digital watermarking system by chopping an image into smaller subimages. On the Website, the series of small images are positioned next to each other and appear the same as the original large image (Petitcolas 2003).



Figure 13. A portion of the JPEG image with the hidden airport map, created by 2Mosaic.

Figure 13 shows an example of 2Mosaic when used against the JPEG image from Figure 8. In this case, the carrier file is split into 165 subimages as above (11 rows of 15 subimages). The 2Mosaic approach is obvious when used. The viewer of the altered image knows immediately that something is amiss.

Summary and Conclusions

Consider the following hypothetical scenario. By prearrangement with members of a terrorist organization, the leader of the terrorist cell puts an item for sale on eBay every Monday and posts a photograph of the item. The item for sale is legitimate. Bids are accepted, money is collected, and items are dutifully delivered. But at some prearranged time during the week, a version of the photograph is posted that contains a hidden message. The cell members know when that time is and download the weekly message. Unless the people are under active investigation, it is unclear that anyone will notice this activity.

This scenario, or one like it, is a viable method for terrorists or criminals to communicate, but is it real? In the aftermath of September 11, 2001, a number of articles appeared suggesting that al Qaeda terrorists employ steganography (Kelly 2001; Kolata 2001; Manoo 2002; McCullagh 2001). In partial response to these reports, several attempts have been made to ascertain the presence of steganography images on the Internet. One well-known study searched more than three million JPEG images on eBay and USENET archives. Using stegdetect, one to two percent of the images were found to be suspicious, but no hidden messages were recovered using stegbreak (Provos and Honeyman 2001; Provos and Honeyman 2003). Another study examined several hundred thousand images from a random set of Websites and, also using stegdetect and stegbreak, obtained similar results (Callinan and Kemick 2003).

Although these projects provide a framework for searching a Website for steganography images, no conclusions can be drawn from them about steganography images on the Internet. First and foremost, stegdetect only looks at JPEG images. Other image types were never examined. Second, a limited number of Websites were examined, too few to make any definitive statements about the Internet as a whole. It is also interesting to note that several steganography researchers are purposely not publishing information about what Internet sites they are examining or what they are finding (Kolata 2001; McCullagh 2001).

There are few hard statistics about the frequency with which steganography software or media are discovered by law enforcement officials in the course of computer forensics analysis. Anecdotal evidence suggests, however, that many computer forensics examiners do not routinely search for steganography software, and many might not recognize such tools if they found them. In addition, the tools that are employed to detect steganography software are often inadequate, with the examiner frequently relying solely on hash sets or the steganography tools themselves (Kruse and Heiser 2001; Nelson et al. 2003; Security Focus 2003). A thorough search for evidence of steganography on a suspect hard drive that might contain thousands of images, audio files, and video clips could take days (Hosmer and Hyde 2003).

Indeed, many digital forensics examiners consider the search for steganography tools and/or steganography media to be a routine part of every examination (Security Focus 2003). But what appears to be lacking is a set of guidelines providing a systematic approach to steganography detection. Even the U.S. Department of Justice search and seizure guidelines for digital evidence barely mention steganography (U.S. Department of Justice 2001; U.S. Department of Justice 2002). Steganalysis will only be one part of an investigation; however, and an investigator might need clues from other aspects of the case to point them in the right direction. A computer forensics examiner might suspect the use of steganography because of the nature of the crime, books in the suspect's library, the type of hardware or software discovered, large sets of seemingly duplicate images, statements made by the suspect or witnesses, or other factors. A Website might be suspect by the nature of its content or the population that it serves. These same items might give the examiner clues to passwords, as well. And searching for steganography is not only necessary in criminal investigations and intelligence gathering operations. Forensic accounting investigators are realizing the need to search for steganography as this becomes a viable way to hide financial records (Hosmer and Hyde 2003; Seward 2003).

It is impossible to know how widespread the use of steganography is by criminals and terrorists (Hosmer and Hyde 2003). Today's truth, however, may not even matter. The use of steganography is certain to increase and will be a growing hurdle for law enforcement and counterterrorism activities. Ignoring the significance of steganography because of the lack of statistics is "security through denial" and not a good strategy.

Steganography will not be found if it is not being looked for. There are some reports that al Qaeda terrorists used pornography as their steganography media (Kelly 2001; Manoo 2002). Steganography and pornography may be technologically and culturally unexpected from that particular adversary, but it demonstrates an ability to work "out of the box." In computer investigations, we too must think and investigate creatively.

References

- AccessData. Forensic Toolkit product page [Online]. (December 29, 2003). Available: http://www.accessdata.com/Product04_Overview.htm.
- Anderson, R., Needham, R., and Shamir, A. Steganographic file system. In: *Proceedings of the Second International Workshop on Information Hiding (IH '98)*, Lecture Notes in Computer Science, vol. 1525. D. Aucsmith, ed., Portland, Oregon, April 14-17, 1998. Springer-Verlag, Berlin, Germany, 1998, pp. 73-82. Also available: <http://www.cl.cam.ac.uk/ftp/users/rja14/sfs3.pdf>.
- Arnold, M., Schmucker, M., and Wolthusen, S. D. *Techniques and Applications of Digital Watermarking and Content Protection*. Artech House, Norwood, Massachusetts, 2003.
- Artz, D. Digital Steganography: Hiding data within data. *IEEE Internet Computing* (2001) 5(3):75-80. Also available: http://www.cc.gatech.edu/classes/AY2003/cs6262_fall/digital_steganography.pdf.
- Barni, M., Podilchuk, C. I., Bartolini, F., and Delp, E. J. Watermark embedding: Hiding a signal within a cover image, *IEEE Communications* (2001) 39(8):102-108.
- Bauer, F. L. *Decrypted Secrets: Methods and Maxims of Cryptology*, 3rd ed. Springer-Verlag, New York, 2002.
- Callinan, J. and Kemick, D. Detecting steganographic content in images found on the Internet. Department of Business Management, University of Pittsburgh at Bradford [Online]. (December 11, 2003). Available: <http://www.chromesplash.com/jcallinan.com/publications/steg.pdf>.
- Chandramouli, R. Mathematical approach to steganalysis. In: *Proceedings of the SPIE Security and Watermarking of Multimedia Contents IV*, vol. 4675. International Society for Optical Engineering, San Jose, California, January 21-24, 2002, pp. 14-25. Also available: <http://www.ece.stevens-tech.edu/~mouli/spiesteg02.pdf>.
- Curran, K. and Bailey, K. An evaluation of image-based steganography methods. *International Journal of Digital Evidence* [Online]. (Fall 2003). Available: http://www.iide.org/docs/03_fall_steganography.pdf.
- Dartmouth College, Institute for Security Technology Studies. A Novel Software for Detection of Hidden Messages within Digital Images [Online]. (December 29, 2003). Available: <http://www.ists.dartmouth.edu/text/steganography.php>.
- El-Khail, R. Hydan [Online]. (December 30, 2003). Available: <http://www.crazyboy.com/hydan/>.
- Farid, H. Detecting Steganographic Messages in Digital Images. Technical Report TR2001-412, Dartmouth College, Computer Science Department, 2001. Also available: <http://www.cs.dartmouth.edu/~farid/publications/tr01.pdf>.
- Farid, H. and Lyu, S. Higher-order wavelet statistics and their application to digital forensics. *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, Wisconsin, June 2003. Also available: <http://www.cs.dartmouth.edu/~farid/publications/sacv03.pdf>.
- Fridrich, J. and Du, R. Secure steganographic methods for palette images. In: *Proceedings of the 3rd Information Hiding Workshop*, Lecture Notes in Computer Science, vol. 1768. Dresden, Germany, September 1999. Springer-Verlag, Berlin, Germany, 2000, pp. 47-60. Also available: http://www.ws.binghamton.edu/fridrich/Research/ihw99_paper1.dot.

- Fridrich, J. and Goljan, M. Practical steganalysis of digital images: State of the art. In: *Proceedings of the SPIE Security and Watermarking of Multimedia Contents IV*, vol. 4675. International Society for Optical Engineering, San Jose, California, January 21-24, 2002, pp. 1-13. Also available: <http://www.ws.binghamton.edu/fridrich/Research/steganalysis01.pdf>.
- Fridrich, J., Goljan, M., and Du, R. Steganalysis based on JPEG compatibility. In: *Proceedings of the SPIE Multimedia Systems and Applications IV, Special Session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding*, vol. 4518. International Society for Optical Engineering, Denver, Colorado, August 21-22, 2001, pp. 275-280. Also available: <http://www.ws.binghamton.edu/fridrich/Research/jpgstego01.pdf>.
- Fridrich, J., Goljan, M., and Hogeia, D. Attacking the OutGuess. In: *Proceedings of the ACM Workshop on Multimedia and Security 2002*, Juan-les-Pins, France, December 2002A. Also available: http://www.ws.binghamton.edu/fridrich/Research/acm_outguess.pdf.
- Fridrich, J., Goljan, M., and Hogeia, D. New methodology for breaking steganographic techniques for JPEGs. In: *Proceedings of the SPIE Security and Watermarking of Multimedia Contents V*, vol. 5020. International Society for Optical Engineering, Santa Clara, California, January 21-24, 2003A, pp. 143-155. Also available: <http://www.ws.binghamton.edu/fridrich/Research/ipeq01.pdf>.
- Fridrich, J., Goljan, M., and Hogeia, D. Steganalysis of JPEG images: Breaking the F5 algorithm. *Proceedings of the 5th International Workshop on Information Hiding (IH 2002)*. F. A. P. Petitcolas, ed., Noordwijkerhout, The Netherlands, October 7-9, 2002B. Springer-Verlag, Berlin, Germany, pp. 310-323. Also available: <http://www.ws.binghamton.edu/fridrich/Research/f5.pdf>.
- Fridrich, J., Goljan, M., Hogeia, D., and Soukal, D. Quantitative steganalysis of digital images: Estimating the secret message length, *Multimedia Systems* (2003B) 9(3):288-302. Also available: <http://www.ws.binghamton.edu/fridrich/Research/mms100.pdf>.
- Fries, B. and Fries, M. *MP3 and Internet Audio Handbook*. TeamCom Books, Burtonsville, Maryland, 2000.
- Guidance Software. EnCase [Online]. (December 29, 2003). Available: <http://www.guidancesoftware.com/>.
- Hashkeeper. Hashkeeper Files [Online]. (December 29, 2003) Available: <http://www.hashkeeper.org/files/>.
- Hernandez Martin, J. R. and Kutter, M. Information retrieval in digital watermarking, *IEEE Communications* (2001) 39(8):110-116.
- Hosmer, C. and Hyde, C. Discovering covert digital evidence. *Digital Forensic Research Workshop (DFRWS) 2003*, August 2003 [Online]. (January 4, 2004). Available: <http://www.dfrws.org/dfrws2003/presentations/Paper-Hosmer-digital-evidence.pdf>.
- Jackson, J. T., Gregg, H., Gunsch, G. H., Claypoole, R. L., and Lamont, G. B. Blind Steganography detection using a computational immune system: A work in progress. *International Journal of Digital Evidence* [Online]. (Winter 2003) (December 21, 2003). Available: http://www.iide.org/docs/02_winter_art4.pdf.
- Johnson, N. F., Duric, Z. and Jajodia, S. *Information Hiding: Steganography and Watermarking: Attacks and Countermeasures*. Kluwer Academic, Norwell, Massachusetts, 2001.
- Johnson, N. F. and Jajodia, S. Exploring steganography: Seeing the unseen, *Computer* (1998A) 31(2):26-34. Also available: <http://www.iitc.com/pub/r2026.pdf>.
- Johnson, N. F. and Jajodia, S. Steganalysis of images created using current steganography software. In: *Proceedings of the Second International Workshop on Information Hiding (IH '98)*, Lecture Notes in Computer Science, vol. 1525. D. Aucsmith, ed. Portland, Oregon, April 14-17, 1998. Springer-Verlag, Berlin, Germany, 1998B, pp.273-289. Also available: <http://www.iitc.com/ihtws98/iigmu.html>.
- Kahn, D. *Codebreakers: The Story of Secret Writing*. Revised ed., Scribner, New York, 1996.
- Kelly, J. Terror groups hide behind Web encryption. *USA Today*, February 5, 2001. Also available: <http://www.usatoday.com/tech/news/2001-02-05-binladen.htm>.
- Kolata, G. Veiled messages of terror may lurk in cyberspace, *New York Times*, October 30, 2001, p. 1.
- Kruse, W. G. and Heiser, J. G. *Computer Forensics: Incident Response Essentials*. Addison-Wesley, Boston, Massachusetts, 2001.
- Kwok, S. H. Watermark-based copyright protection system security, *Communications of the ACM* (2003) 46(10):98-101.
- Manoo, F. Case of the missing code, *Salon.com*, July 17, 2002 [Online]. (December 29, 2003). Available: <http://www.salon.com/tech/feature/2002/07/17/steganography/>.
- Maresware. Hash Set CD [Online]. (December 29, 2003). Available: http://www.dmares.com/maresware/hash_cd.htm.
- McCullagh, D. Secret messages come in .Wavs. *WIRED News*, February 20, 2001 [Online]. (December 11, 2003). Available: <http://www.wired.com/news/politics/0,1283,41861,00.html>.
- McDonald, A. D. and Kuhn, M. G. StegFS: A steganographic file system for Linux. In: *Proceedings of the Third International Workshop on Information Hiding (IH '99)*, Lecture Notes in Computer Science, vol. 1768. A. Pfitzmann, ed., Dresden, Germany, September 29-October 1, 1999. Springer-Verlag, Berlin, Germany, 2000, pp. 462-477. Also available: <http://www.cl.cam.ac.uk/~mqk25/ih99-stegfs.pdf>.
- Monash University. JPEG Image Coding Standard [Online]. (January 10, 2004). Available: <http://www.ctie.monash.edu.au/emerge/multimedia/ipeq/>.
- moreCrayons. color cube [Online]. (December 12, 2003). Available: <http://www.morecrayons.com/palettes/webSmart/colorcube.php>.
- National Software Reference Library. NSRL Project Web Site [Online]. (December 29, 2003). Available: <http://www.nsrll.nist.gov/>.
- Nelson, B., Phillips, A., Enfinger, F., and Steuart, C. *Guide to Computer Forensics and Investigations*. Course Technology, Boston, Massachusetts, 2003.
- OutGuess. Steganography Detection with Stegdetect [Online]. (December 29, 2003). Available: <http://www.outguess.org/detection.php>.
- Ozer, H., Avciabas, I., Sankur, B., and Memon N. Steganalysis of audio based on audio quality metrics. In: *Proceedings of the SPIE, Security and Watermarking of Multimedia Contents V*, vol. 5020, SPIE, Santa Clara, California, 2003, pp. 55-66. Also available: www.busim.ee.boun.edu.tr/~sankur/SankurFolder/Audio_Steganalysis_16.doc.
- Petitcolas, F. A. P. 'mosaic' attack [Online]. (December 29, 2003). Available: <http://www.petitcolas.net/fabien/watermarking/2mosaic/index.html>.
- Provos, N. and Honeyman, P. Detecting Steganographic Content on the Internet. Center for Information Technology Integration, University of Michigan, CITI Technical Report 01-11 [Online]. (August 2001). Available: <http://www.citi.umich.edu/techreports/reports/citi-tr-01-11.pdf>.
- Provos, N. and Honeyman, P. Hide and seek: An introduction to steganography. *IEEE Security & Privacy* (2003) 1(3):32-44. Also available: <http://niels.xtdnet.nl/papers/practical.pdf>.
- Rey, R. F. (ed.). *Engineering and Operations in the Bell System*, 2nd. ed., AT&T Bell Laboratories, Murray Hill, New Jersey, 1983.
- Rowland, C. H. Covert Channels in the TCP/IP Protocol Suite. *First Monday*, 1996 [Online]. (January 10, 2004). Available: http://www.firstmonday.dk/issues/issue2_5/rowland/ or <http://www.guides.sk/psionic/covert/covert.tcp.txt>.
- Security Focus. Forensics mailing list, personal communication, December 1-26, 2003.

Seward, J. Debtor's digital reckonings. *International Journal of Digital Evidence*, Fall 2003 [Online]. (January 3, 2004). Available: http://www.iide.org/docs/03_fall_seward.pdf.

Seward, J. Personal communication, January 2004.

Simmons, G. J. Prisoners' problem and the subliminal channel. In: *Advances in Cryptology: Proceedings of CRYPTO 83*. D. Chaum, ed. Plenum, New York, 1983, pp. 51-67.

spam mimic [Online]. (December 29, 2003). Available: <http://www.spammimic.com/>.

StegoArchive.com [Online]. (December 30, 2003). Available: <http://www.stegoarchive.com/>.

U.S. Department of Justice. *Electronic Crime Scene Investigation: A Guide for First Responders*. Office of Justice Programs, National Institute of Justice, Technical Working Group for Electronic Crime Scene Investigation, NCJ 187736, July 2001. Also available: <http://www.ncjrs.org/pdffiles1/nij/187736.pdf>.

U.S. Department of Justice. *Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations*. Criminal Division, Computer Crime and Intellectual Property Section, July 2002. Also available: <http://www.cybercrime.gov/s&smanual2002.pdf>.

Voloshynovskiy, S., Pereira, S., Pun, T., Eggers, J. J., and Su, J. K. Attacks on digital watermarks: Classification, estimation-based attacks, and benchmarks, *IEEE Communications* (2001) 39(8):118-126.

Warchalking. Warchalking: Collaboratively creating a hobo-language for free wireless networking [Online]. (December 21, 2003). Available: <http://www.warchalking.org/>.

Wayner, P. *Disappearing Cryptography: Information Hiding: Steganography & Watermarking*. 2nd. ed., Morgan Kaufmann, San Francisco, California, 2002.

WetStone Technologies. Gargoyle [Online]. (May 24, 2004A). Available: http://www.wetstonetech.com/f/Gargoyle_2.1_Datasheet.pdf.

WetStone Technologies. Stego Suite [Online]. (May 24, 2004B). Available: http://www.wetstonetech.com/f/Stego_Suite_Datasheet_for_web.pdf.

Appendix A: Additional Websites

- Computer Forensics, Cybercrime and Steganography Resources Website, Steganography & Data Hiding - Articles, Links, and Whitepapers page (<http://www.forensics.nl/steganography>)
- GCK's steganography links (www.garykessler.net/library/securityurl.html#crypto)
- Neil Johnson's Steganography and Digital Watermarking page (<http://www.iitc.com/Steganography/>)

Appendix B: Companion Downloads to this Article

The hidden, carrier, and steganography files mentioned in this article can be downloaded from the <http://digitalforensics.champlain.edu/fsc/> directory. Use the password "tyui" to recover the hidden file from the steganography files.

- Figure 5 airport image: [btv_map.gif](#)
- Figure 6 original carrier: [mall_at_night.gif](#)
- Figure 6 stego file: [mall_at_night_btv2.gif](#)
- Figure 8 original carrier: [lightening_jars.jpg](#)
- Figure 8 stego file: [lightening_jars_btv.jpg](#)
- Figure 9 original carrier: [hitchhiker_beginning.wav](#)
- Figure 9 stego file: [hitchhiker_beginning_btv.wav](#)
- Figure 13 disrupted stego file: [disrupt/lighte~1.html](#)

The noncommercial software employed in the examples in this article can be downloaded from the following mirror site:

- 2Mosaic (http://digitalforensics.champlain.edu/download/2Mosaic_0_2_2.zip)
- Gif-It-Up (<http://digitalforensics.champlain.edu/download/Gif-it-up.exe>)
- JPHS for Windows (http://digitalforensics.champlain.edu/download/jphs_05.zip)
- Stegdetect (<http://digitalforensics.champlain.edu/download/stegdetect-0.4.zip>)
- S-Tools (<http://digitalforensics.champlain.edu/download/s-tools4.zip>)

Appendix C: Commercial Vendors Mentioned in this Article

AccessData Corp.
Orem, Utah
www.accessdata.com

Guidance Software
Pasadena, California
www.guidancesoftware.com

WetStone Technologies
Cortland, New York
www.wetstonetech.com

Author Bio and Acknowledgements

Gary C. Kessler is an Associate Professor at [Champlain College](#), Burlington, Vermont, and the director of the [Computer & Digital Forensics](#) program. He holds a B.A. in Mathematics and M.S. in Computer Science, and works closely with the [Vermont Internet Crimes Against Children Task Force](#). He can be reached via e-mail at gary.kessler@champlain.edu.

The author gratefully acknowledges comments and suggestions put forth by Eric Cole, Jack Seward, and FSC's anonymous reviewers.