

Introduction to Steganography

The word Steganography is derived from the Greek words Steganos meaning "covered" and graphy meaning "writing or drawing". It is also commonly known as 'Disappearing Cryptography'.

What is Steganography?

Steganography is the art and science of hiding information into covert channels so as to conceal the information and prevent the detection of the hidden message. Today, steganography refers to hiding information in digital picture files and audio files.

Steganography is defined as "hiding information within a noise; a way to supplement (not replace) encryption, to prevent the existence of encrypted data from being detected" [1]. Steganography and Cryptography are cousins in the data hiding techniques. Cryptography is the practice of scrambling a message to an obscured form to prevent others from understanding it. Steganography is the study of obscuring the message so that it cannot be seen.

Who is this tutorial for?

This tutorial is an elementary overview of some basic areas of steganography. It provides an introduction to steganography for those unfamiliar with the field. Specifically, the tutorial covers the history and basics of steganography, and looks at image files and how to hide information in them, and briefly makes mention of digital watermarking, and steganalysis - the process of detecting steganography. The tutorial walks through two example programs that hide text within an image.

History

Steganography dates back to ancient Greece when etching messages or images in wooden tablets and covering them with wax, and tattooing a shaved messenger's head, letting the hair grow back, and then shaving the head again to read the message, were common practices.

Early in WWII steganographic technology consisted almost exclusively of invisible inks. Sources for invisible inks include milk, vinegar, fruit juices and urine, that darken when heated. The following message was sent by a German spy during WWII [5]:

Apparently neutral's protest is thoroughly discounted
and ignored. Isman hard hit. Blockade issue affects
pretext for embargo on by products, ejecting suets and
vegetable oils.

Taking the second letter in each word the following message emerges:

Pershing sails from NY June 1.

When invisible inks became easy to decode through improved technology, null ciphers were used. Null ciphers are unencrypted messages that are indiscernible in innocent sounding messages. An example of such a message is [6]:

Fishing freshwater bends and saltwater
coasts rewards anyone feeling stressed.
Resourceful anglers usually find masterful
leapers fun and admit swordfish rank
overwhelming anyday.

Taking the third letter in each word the following message emerges:

Introduction to Steganography

Send Lawyers, Guns, and Money.

The Germans developed the microdot technology during WWII. Microdots are text or photographic images that are shrunk down to the size and shape of a period or the dot of an i or j. Microdots were usually sent by writing a letter containing periods, i's, or j's, and the intended recipient could read the messages using a microscope. Because of the extremely small size of the microdots the messages typically went unnoticed by inspectors.

A steganographic message generally appears to be something else, like an article or a picture, or some other "cover" message. Drawings have often been used to conceal information since it is easy to encode a message by varying lines, colors or other elements in pictures. This tutorial will focus on image files to hide text messages.

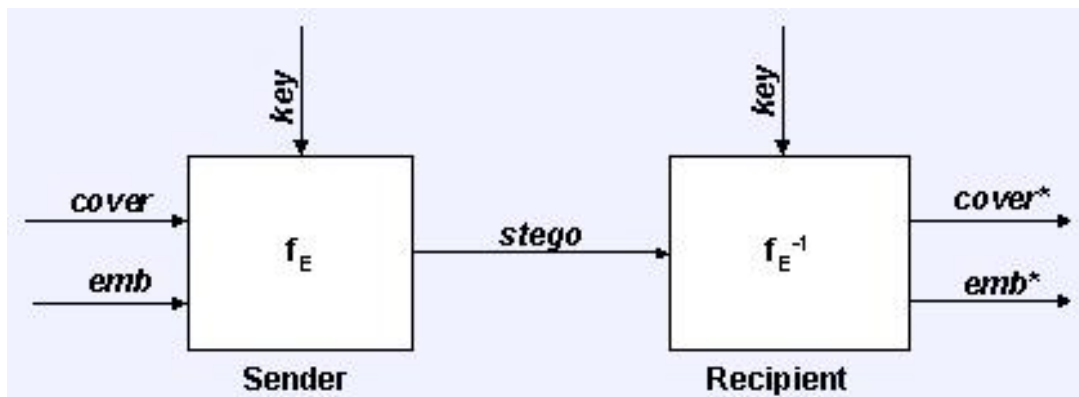
The Basics

Modern steganography refers to hiding information in digital picture files and audio files. It works by replacing bits of unused data in regular digital files with bits of invisible information. To embed hidden information into an image requires two files - the cover image file that will hold the hidden data and the secret message file. A message may be plain text, cypher text (or another image). When combined, the cover image and the hidden message makes a stego image. A stego-key or password may be used to hide and decode the message. Special software is needed for steganography. In this tutorial we will look at two programs that hide text within images.

Why use Steganography?

The goal of steganography is to avoid drawing attention to the transmission of a hidden message. If suspicion is raised, then this goal is defeated.

A Steganographic System [9]



f_E : steganographic function "embedding"

f_E^{-1} : steganographic function "extracting"

cover: cover data in which *emb* will be hidden

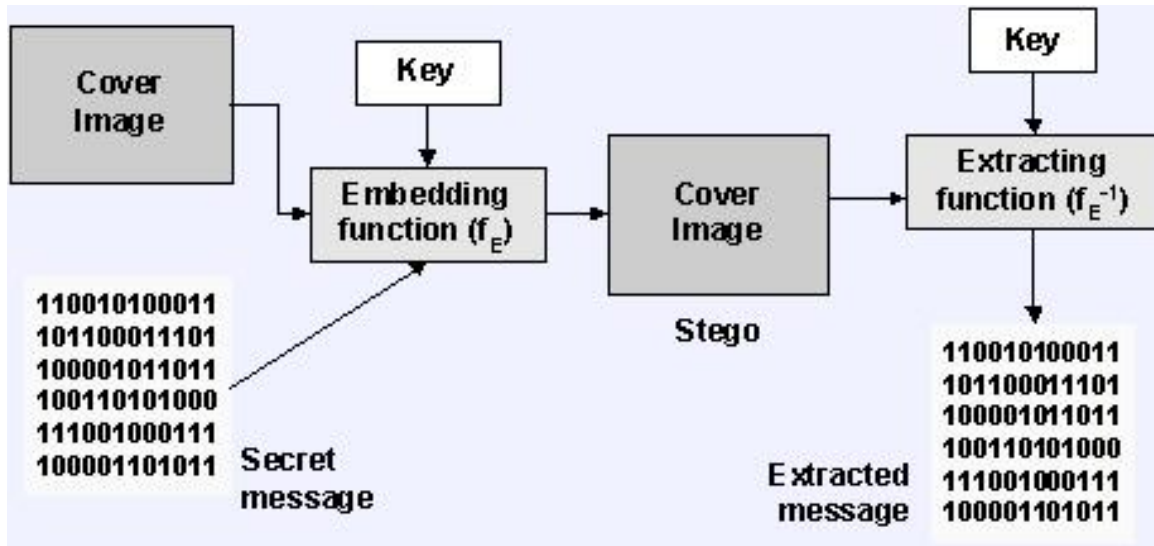
emb: message to be hidden

key: parameter of f_E

Introduction to Steganography

stego: cover data with the hidden message

A Graphical Version of the Steganographic System

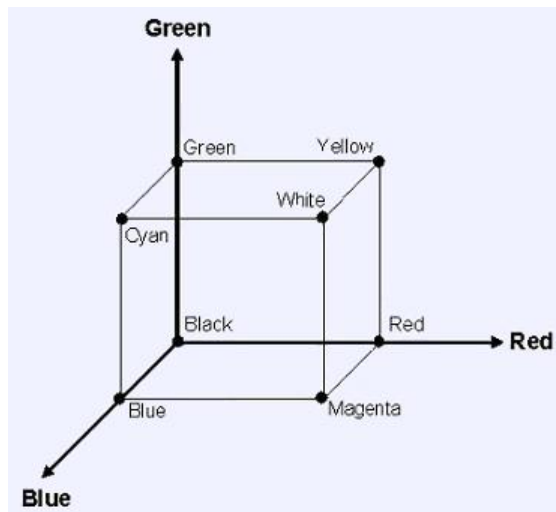


Steganographic messages may first be encrypted and then a cover message is modified to contain the encrypted message, resulting in stego text. Only those who know the technique used can recover the message and, if required, decrypt it.

The message may be a few thousand bits (often at 7 or 8 bits per text character) embedded in millions of other bits. Probably the most typical use is digital images. Digital images are commonly stored in either 24-bit or 8-bit files. If an 8-bit image is viewed as a grid and the grid is made up of cells, these cells are called pixels. Each pixel consists of an 8-bit binary number (or a single byte), and each 8-bit binary number refers to the color palette (a set of colors defined within the image). All color variations for the pixels are derived from three primary colors: red, green, and blue. Each primary color is represented by 1 byte (= 8 bits).

The RGB Color Cube [16]

Black is shown as the zero point intersection of the three-color axes. The mixture of 100 percent red, 100 percent blue, and no green combine to form magenta; cyan is 100 percent green and 100 percent blue with no red; and yellow is 100 percent green and 100 percent red with no blue. White is the presence of all three colors.



Introduction to Steganography

The hidden message must be smaller in size (in terms of number of bits) than the image file. Pixel representation contributes to file size. A common image size is 640 x 480 pixels and can contain about 300 kilobits (307,200) of data. A 24-bit image of 1024 x 768 pixels - common in high-resolution graphics - has the potential to produce a file over two megabytes (2,359,296) in size. 24-bit images can store 3 bits of data in each pixel since it uses 3 bytes per pixel to represent a color value.

GIF files usually use an 8-bit palette, thus allowing only 256 RGB colors in the image (0-255). JPEG files use a 24-bit palette, and BMP files use an 8-bit or 24-bit palette. 24-bit images provide the most space for concealing information.

The best pictures to use are those with many halftones (black and white) or loads of details and variations, such as images depicting nature, so that a lot of information can be hidden. When a message is hidden in the picture, the image changes although this is not visible to the human eye. In this case the stego text is the image that looks harmless to the casual eye.

It is possible to exploit lossy compression schemes (such as JPEG that saves space but may not maintain the original image's integrity) for steganographic use since the compression scheme always introduces some error into the decompressed data. However, if the image integrity is corrupted the hidden message may be corrupted as well.

Digital watermarking technology is viewed as "an enabling agent allowing more widespread sharing and use of that content while decreasing worry over piracy" [19]. Today steganography is often used for digital watermarking to hide copyright or ownership information in an image, movie, or audio file. A copyright holder can pull the hidden copyright or ownership information out of a suspect file to prove it is stolen. Digital watermarking is not used for authenticating documents. (Digital signatures perform this task.) A digital watermark refers to the ability to unobtrusively include information in a file, and is commonly executed through a variety of cryptographic techniques, collectively known as steganography.

In the next section we look at some of the modern techniques used to hide information in images.
Steganography Techniques

Information hiding techniques are receiving much attention today. The main motivation for this is largely due to fear of encryption services getting outlawed, and copyright owners who want to track confidential and intellectual property copyright against unauthorized access and use in digital materials such as music, film, book and software through the use of digital watermarks.

There are many ways to hide information in digital images. We look at the following approaches:

- least significant bit insertion
- masking and filtering
- algorithms and transformations

Each of these techniques have varying degrees of success.

Least significant bit insertion

Many stego tools make use of least significant bit (LSB). For example, 11111111 is an 8-bit binary number. The rightmost bit is called the LSB because changing it has the least effect on the value of the number. The idea is that the LSB of every byte can be replaced with little change to the overall file. The binary data of the secret message is broken up and then inserted into the LSB of each pixel in the image file.

Introduction to Steganography

Hiding the Data

Using the Red, Green, Blue (RGB) model a stego tool makes a copy of an image palette, say, an 8-bit image. The copy is rearranged so that colors near each other in the RGB model are near each other in the palette. The LSB of each pixels 8-bit binary number is replaced with one bit from the hidden message. A new RGB color in the copied palette is found. A new 8-bit binary number of the new RGB color in the original palette is found. The pixel is changed to the 8-bit binary number of the new RGB color.

Recovering the Data

The stego tool finds the 8-bit binary number of each pixels RGB color. The LSB of each pixel's 8-bit binary number is one bit of the hidden data file. Each LSB is then written to an output file.

A simplified example with an 8-bit image

```
1 pixel:
      (00  01  10  11)
      white red  green  blue
Insert 0011:
      (00  00  11  11)
      white white  blue  blue
```

As can be seen from the example, with an 8-bit image, the cover image must be carefully selected since LSB manipulation is not as forgiving because of the color limitations. To hide information in the LSBs of each byte of a 24-bit image, it is possible to store 3 bits in each pixel.

A simplified example with a 24-bit image

```
1 pixel:
      (00100111 11101001 11001000)
Insert 101:
      (00100111 11101000 11001001)
      red      green  blue
```

LSB insertion works well with gray-scale images as well. It is possible to hide data in the least and second least significant bits and the human eye would still not be able to discern it.

Unfortunately LSB insertion is vulnerable to slight image manipulation such as cropping and compression. For example, converting a GIF or a BMP image, which reconstructs the original message exactly (lossless compression), to a JPEG format, which does not (lossy compression), and then converting back, can destroy the data in the LSBs.

Masking and Filtering

Masking and filtering techniques hide information by marking an image and is usually restricted to 24-bit and gray-scale images. Digital watermarks include information such as copyright, ownership, or license. While traditional steganography conceals information, watermarks extend information since it becomes an attribute of the cover image.

Introduction to Steganography

Masking techniques hide information in such a way that the hidden message is more integral to the cover image than simply hiding data in the "noise" level. Masking adds redundancy to the hidden information. This makes the masking technique more suitable than LSB with lossy JPEG images. It may also help protect against some image processing such as cropping and rotating.

Algorithms and Transformations

Another steganography technique is to hide data in mathematical functions that are in compression algorithms. The idea is to hide the data bits in the least significant coefficients.

A key advantage of JPEG images over other formats is its lossy compression methods. It enables high quality images to be stored in relatively small files. The compressed data is stored as integers but the calculations for the quantization process require floating point calculations which are rounded. Errors introduced by rounding define the lossy characteristic of the JPEG compression method. JPEG images use the discrete cosine transform (DCT) technique to achieve image compression. The DCT is "a technique for expressing a waveform as a weighted sum of cosines" [10]. In a JPEG file, the image is made up of DCT coefficient. When a file is steganographically embedded into a JPEG image, the relation of these coefficients is altered. Instead of actual bits in the image being changed as in LSB steganography, it is the relation of the coefficients to one another that is altered.

In addition to DCT, images can be processed with fast Fourier transform (FFT). FFT is "an algorithm for computing the Fourier transform of a set of discrete data values" [11]. The FFT expresses a finite set of data points in terms of its component frequencies. It also solves the identical inverse problem of reconstructing a signal from the frequency data.

The wavelet transform is a transformation to basis functions that are localized in frequency. The wavelet compression methods are better at representing transients, such as an image of stars on a night sky. This means that "elements of some data signal that are transient can be represented by a smaller amount of information than would be the case if some other transform, such as the more widespread discrete cosine transform, had been used" [12]. Wavelet compressions are good for transient signal characteristics but not for smooth, periodic signals.

Many transform domain methods are not dependent on the image format so that the hidden message is retained after conversion between lossless and lossy formats.

Hiding the Data

The steps are to take the DCT or wavelet transform of the cover image and find the coefficients below a specific threshold. Replace these bits with bits to be hidden (for example, use LSB insertion) and then take the inverse transform and store it as a regular image.

Recovering the Data

To extract the hidden data take the transform of the modified image and find the coefficients below a specific threshold. Extract bits of data from these coefficients and combine the bits into an actual message.

Other techniques of steganography include spread spectrum steganography, statistical steganography, distortion, and cover generation steganography.

In the next section we look at two steganographic program examples.

Introduction to Steganography

Example Programs

There are many free tools available online that allow one to hide text inside images using a password. Some of these tools include S-Tools, Steganos, Windstorm, Hide4PGP, and JPHS. We consider two programs that are legal and free downloads on the Internet.

- JPHS - Hide and Seek
- 4t HIT Mail Privacy LITE 1.01

JPHS

JPHS stands for Jpeg hide and seek. The program by Allan Latham is designed to be used with JPEG files and lossy compression and is available in Windows and Linux versions. No installation is required. JPHS is made up of two programs - JPHIDE and JPSEEK - which allows you to hide a message file in a jpeg visual image. JPHIDE.EXE hides a data file in a jpeg file. JPSEEK.EXE recovers a file hidden with JPHIDE.EXE. For the Windows version, JPHSWIN.EXE performs the same functions as the two programs above.

JPHIDE and JPSEEK distributes the hidden file in the jpeg image so that both the visual and statistical effects are minimized. JPHS uses least significant bit overwriting of the discrete cosine transform coefficients used by the jpeg algorithm. Simple programs that store the hidden data in low order bits may result in the jpeg image being so statistically different from the normal jpeg file that the hidden file can be recovered easily.

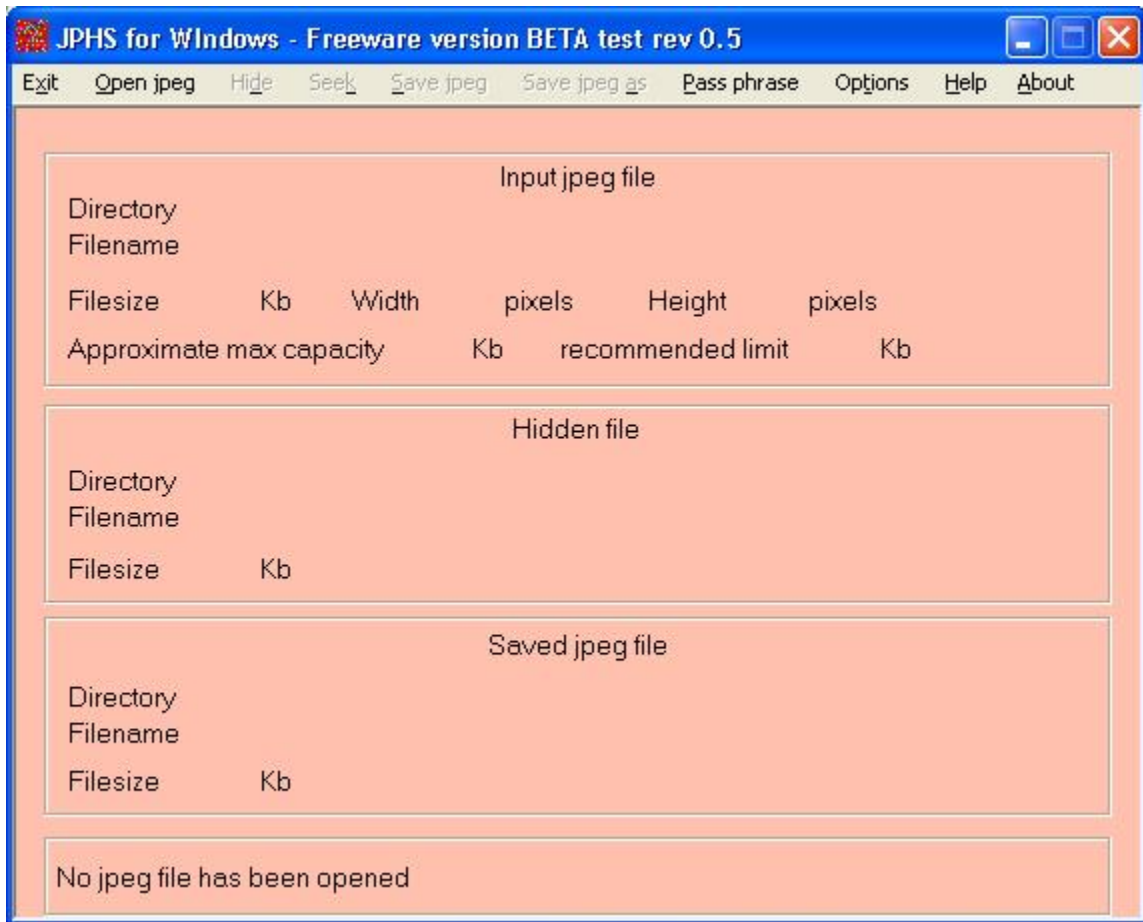
JPHIDE uses the Blowfish crypto algorithm for least significant bit randomization and encryption to determine where to store the bits of the hidden file. The program asks for a pass phrase to initialize this process. Although the hidden file is "encrypted" by the software it is recommended that the hidden file is encrypted (by some other tool) before inserted into the jpeg file. Up to 10% can be inserted into a jpeg file with minimal visual and statistical effect.

If the original jpeg file is available then a hidden file can always be detected, although to extract the information one must know the stego tool used, and the pass phrase.

Test the program

1. Download this file [jphs05.zip](http://linux01.gwdg.de/~alatham/stego.html) (180KB) if you are a Windows user. Otherwise, go to <http://linux01.gwdg.de/~alatham/stego.html> to download the Linux version.
2. Have a jpeg image and a text file ready for use. Alternatively, you can download and use my image [My jpeg](#), and text file [My message](#).
3. For Windows users, run the [Jphswin.exe](#) file. Accept the terms outlined. The following screen should display:

Introduction to Steganography



4. Select 'Open jpeg'. Choose any jpeg image.
5. Select the 'Hide' option. Enter the same pass phrase to both boxes, e.g. 12345. Choose any text file that contains the hidden message.
6. Select 'Save jpeg as'. Enter a file name for the new image file.
7. Now Select 'Open jpeg'. Choose the new jpeg file you saved in the previous step.
8. Select the 'Seek' option. Enter the pass phrase you used, e.g. 12345, in both boxes.
9. Enter a name for the recovered message file. Caution: provide the '.txt' extension in the file name.
10. Go to the folder where you saved the recovered message file and click to view. The hidden message is displayed.

4t HIT Mail Privacy LITE 1.01

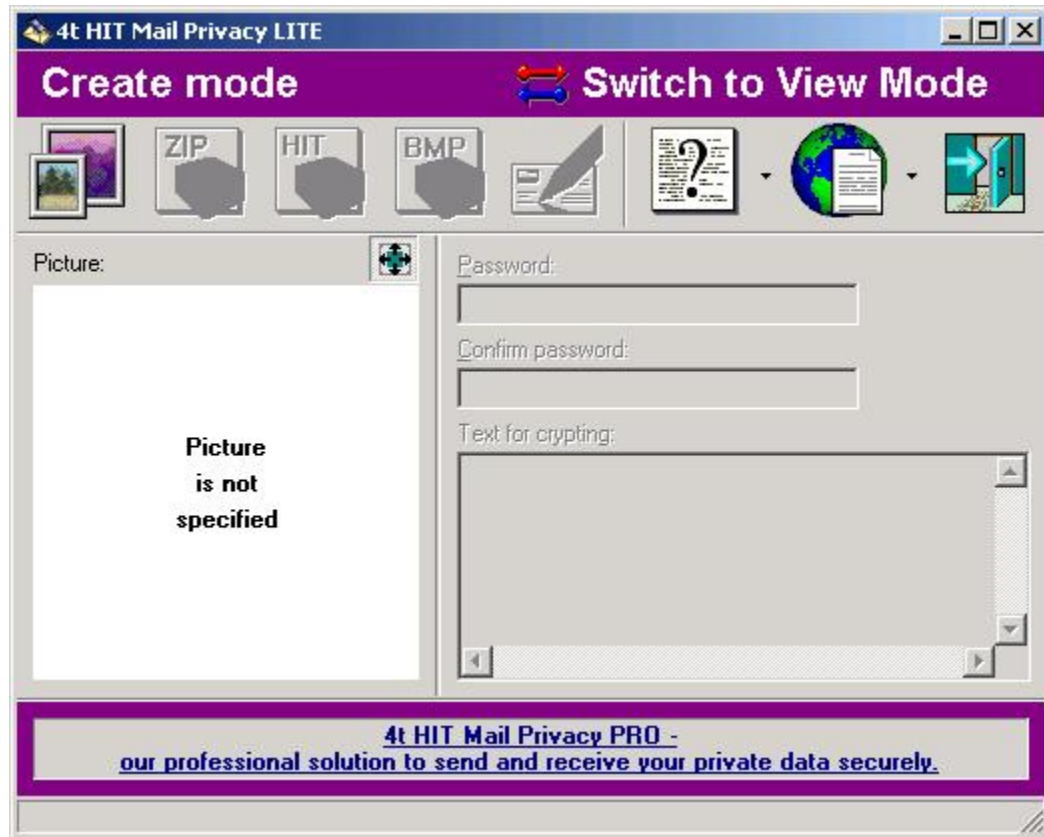
This tool is provided free of charge by 4t Niagara Software. The program is available at 1000 apps for Windows platform only.


4t HIT Mail Privacy Lite works with a number of popular image formats such as JPG, BMP, ZIP or HIT files. Data is sent and received encrypted. Unfortunately with the Lite version, there is a limit of 250 characters for the hidden message.

Introduction to Steganography

Test the program



1. Download this file 4t-hitl.zip (930KB).
2. Read the license.txt file before installing the program. If you agree to the terms of use, run the setup.exe file.
3. After the install, the following screen should display when you run the application:



4. Select an image by clicking the  icon.
5. Enter a password and confirm the password.
6. Enter the secret message.
7. You can save the encrypted message and image as a ZIP, HIT, or BMP file. (Note the significant increase in the size of the saved file.) If you save in a HIT format you can click on the file directly, enter the password to view the message (you must have the application installed). Otherwise, if the saved file is in BMP or ZIP format, you need to open the saved file from a running application, as in point 8 and 9 below.
8. To recover the message, switch to view mode.

Introduction to Steganography



9. Select the saved file by clicking the  icon and open the file.
10. Enter the appropriate password.
11. Click the  icon. The hidden message is decrypted and displayed in the text box.

Steganalysis

Steganalysis is a relatively new research discipline with few articles appearing before the late-1990s. Steganalysis is "the process of detecting steganography by looking at variances between bit patterns and unusually large file sizes" [18]. It is the art of discovering and rendering useless covert messages.

What is the goal?

The goal of steganalysis is to identify suspected information streams, determine whether or not they have hidden messages encoded into them, and, if possible, recover the hidden information.

The challenge of steganalysis is that:

- The suspect information stream, such as a signal or a file, may or may not have hidden data encoded into them.
- The hidden data, if any, may have been encrypted before inserted into the signal or file.
- Some of the suspect signal or file may have noise or irrelevant data encoded into them (which can make analysis very time consuming).

Introduction to Steganography

- Unless it is possible to fully recover, decrypt and inspect the hidden data, often one has only a suspect information stream and cannot be sure that it is being used for transporting secret information.

Unlike cryptanalysis, where it is evident that intercepted encrypted data contains a message, steganalysis generally starts with several suspect information streams but uncertainty whether any of these contain hidden message. The steganalyst starts by reducing the set of suspect information streams to a subset of most likely altered information streams. This is usually done with statistical analysis using advanced statistics techniques.

Types of Attacks

Attacks and analysis on hidden information may take several forms: detecting, extracting, and disabling or destroying hidden information. An attack approach is dependent on what information is available to the steganalyst (the person who is attempting to detect steganography-based information streams).

- Steganography-only attack: Only the steganography medium is available for analysis.
- Known-carrier attack: The carrier, that is, the original cover, and steganography media are both available for analysis.
- Known-message attack: The hidden message is known.
- Chosen-steganography attack: The steganography medium and tool (or algorithm) are both known.
- Chosen-message attack: A known message and steganography tool (or algorithm) are used to create steganography media for future analysis and comparison. The goal in this attack is to determine corresponding patterns in the steganography medium that may point to the use of specific steganography tools or algorithms.
- Known-steganography attack: The carrier and steganography medium, as well as the steganography tool or algorithm, are known.

Where can information be hidden?

Almost anywhere on the Internet.

For example, there are several places on a webpage to hide information:

- Text: Text information can be hidden by making it the same color as the background. Small shift in word and line spacing may be difficult to visually detect. To find any invisible text, do a Control-A on the page. This will select all the text on the page. To reveal distortions in the text, view it in a word processor. Hidden message can also be placed into the general context of the web page. The easiest way to detect this is to look for awkward sentence structures.
- Non-text elements: Any graphic or media clip can contain hidden links or messages.
- Links: Links can be created without it being underlined, or change color when the mouse cursor moves over them. The easiest way to find links on a page is to view the source and search for HREF=. Alternatively, one can also use the tab key to highlight all the clickable items on the page.
- Comments: The contents of a comment is viewable only in the source code of a page.

Introduction to Steganography

- Structure: Most browsers ignore information provided in the source code that is not interpretable. For example, unusual options in markup tags can possibly hide clues.
- Frames: View the source code of each frame on a web page. Sometimes a site disables the right-click or use of the menu function to find the source code. In these cases, try using the command `view-source:http://(site url)` in the address line of the browser.

Steganalysis Techniques

Hiding information within an electronic medium cause alterations of the medium properties that can result in some form of degradation or unusual characteristics.

Unusual Patterns

Unusual patterns in a stego image are suspicious. For example, there are some disk analysis utilities that can filter hidden information in unused partitions in storage devices. Filters can also be used to identify TCP/IP packets that contain hidden or invalid information in the packet headers. TCP/IP packets used to transport information across the Internet have unused or reserved space in the packet headers. Packet headers are seldom read by humans and thus makes an ideal place to hide data. The disadvantage of using this method is that firewalls can be configured to filter out packets that contain inappropriate data in the reserved fields. In addition, hiding information in packet headers is unreliable because it is possible that TCP/IP headers, and the reserved bits, are overwritten in the routing process, thus defeating the steganographic transmission.

Visual Detection

Analyzing repetitive patterns may reveal the identification of a steganography tool or hidden information. To inspect these patterns an approach is to compare the original cover image with the stego image and note visible differences. This is called a known-carrier attack. By comparing numerous images it is possible that patterns emerge as signatures to a steganography tool.

If the cover images are not available for comparison, the derived known signatures are sufficient to imply the existence of a hidden message and identify the tool used to embed the message. Detection of these signatures can be automated into tools for detecting steganography. Stegodetect takes advantage of palette patterns and signatures, and analyzes pixels that stands out from the other pixels in its area.

Another visual clue to the presence of hidden information is padding or cropping of an image. With some stego tools if an image does not fit into a fixed size it is cropped or padded with black spaces. There may also be a difference in the file size between the stego-image and the cover image. Another indicator is a large increase or decrease in the number of unique colors, or colors in a palette which increase incrementally rather than randomly (except gray scale images).

Tools To Detect Steganography

The disabling or removal of hidden information in images is dependent on the image processing techniques. For example, with LSB methods of inserting data, simply compressing the image using lossy compression is enough to disable or remove the hidden message.

There are several available steganographic detection tools such as EnCase by Guidance Software Inc., ILook Investigator by Electronic Crimes Program, Washington DC, various MD5 hashing utilities, etc. For information on available steganographic and steganalysis tools, visit the Computer Forensics, Cybercrime and Steganography Resources website at <http://www.forensics.nl/tools>.

Stegdetect, provided by Niels Provos, is a popular automated tool for detecting steganographic content in images. Provos is the author of the steganography program called OutGuess. Stegdetect is a program that detects data hidden in JPEG images using certain steganography-based applications.

Introduction to Steganography

The detectable schemes include JSteg, JPHide (unix and windows), Invisible Secrets, OutGuess 01.3b, F5 (header analysis), AppendX and Camouflage. Note that OutGuess 0.2 is undetectable by statistical analysis.

Example of a Stegdetect output:

```
$ stegdetect *.jpg
dscf0001.jpg : outguess(old)(***) jphide(*)
dscf0002.jpg : negative
dscf0003.jpg : jsteg(***)
wonder-5.jpg : jphide(**)
[...]
```

Certain types of images are more likely to show up as false positives, such as, drawings, paintings, and images with monotone backgrounds.

StegBreak is a brute force attack tool for determining the passphrase assigned to the cover file embedded with a hidden message. StegBreak is used to launch dictionary attacks against images to determine if content was hidden with JSteg-Shell, JPHide or OutGuess 0.13b. Steganographic systems embed header information in front of a hidden message. The header contains information such as the length of the message and compression methods. Stegbreak chooses a key from a dictionary and uses it to retrieve header information. If the header makes sense the guessed key is a candidate.

The Stegdetect tool can be downloaded from <http://www.outguess.org/download.php>.

Watermarking

Currently, watermarking is used for

- Copyright protection - to prevent third parties from claiming the ownership of the digital media.
- Fingerprinting - to convey information about the recipient of the digital media (rather than the owner) in order to track distributed copies of the media.
- Copy protection - to control data copying devices and prevent them from copying the digital media if the media is copy-protected.
- Image authentication - to check the authenticity of the digital media.

Watermarks are embedded directly into a file data, usually by making minor variations to pixel brightness. Watermarks are not text that is included in the file description "Comments" field. The variations in the data bits are subtle and cannot be detected by the human eye. The patterns are repeated many times, allowing the information contained in the watermark to be recovered even if the image is cropped. Some watermarks can survive a limited amount of image manipulation, such as contrast adjustments and filtering.

There are four classes of attacks on watermarking schemes: robustness attacks, presentation attacks, interpretation attacks, and legal attacks.

Robustness Attacks

Robustness attacks attempt to diminish or remove the presence of watermarks in a suspect image without rendering the image useless. These attacks can be classified into two types: signal processing attacks, and analytic and algorithmic attacks. A signal processing attack include common processing

Introduction to Steganography

operations such as compression, filtering, resizing, printing, and scanning. Analytic and algorithmic attacks involve removal or weakening of watermarks in images based on the specific methods of watermark insertion and detection. An example is the collusion attack where different watermarked versions of the same image are combined to generate a new image, reducing the strength of the watermark.

Presentation Attacks

In a presentation attack the watermarked content is manipulated so a detector cannot find it. For example, misaligning a watermarked image can sometimes fool an automated detector such as a Webcrawler, even though the underlying pixel values are not changed. Other examples of presentation attacks include rotation and enlargement. In a presentation attack the watermark does not need to be removed or diminished.

Interpretation Attacks

Interpretation attacks seek to falsify invalid or multiple interpretations of a watermark. For example, an attacker can attempt to make another watermark appear in the same watermarked image with strength equal to that of the owners watermark, creating an ownership deadlock. The pixel values may or may not change.

Legal Attacks

Legal attacks is the ability of an attacker to cast doubt on the watermarking scheme in the courts. These attacks rely on existing and future legislation on copyright laws and digital information ownership, the credibility of the owner and of the attacker, the financial strength of the owner versus that of the attacker, the expert witnesses, and the competence of the lawyers.

A truly robust watermarking scheme has to minimize an attackers ability to cast doubt on technical evidences presented in court.

As an aside, in a 2003 press release by Digimarc, a leading supplier of secure media solutions, it was reported that Corbis - a visual solutions provider (licensing images) - identified up to 50 cases of unauthorized uses of Corbis images per month using Digimarc digital watermarking solutions.

Summary

Steganography is a branch of cryptography. While most cryptography applications are used to encrypt information so that only the sender and recipient can understand it, steganography hides information that only the sender and recipient know it exist. The secret message is hidden in plain sight. The public may see the data, unaware that a hidden message is present.

Steganography is used not only to digital images but also to other media such as voice, text and binary files, and communication channels.

Steganography can be used for a variety of reasons. Legitimate purposes include watermarking images for copyright protection. Digital watermarks are similar to steganography in that they appear to be part of the original object and is not easily detectable by the casual eye. Steganography can also be used to tag notes to online images and is used to maintain the confidentiality of valuable information.

Unfortunately steganography can also be used for illegitimate purposes. For example, if someone was trying to steal data, they could conceal it in files and send it out in an innocent looking email or file transfer. Refer to the steganography article by Computerworld [18] for a real world account.

Introduction to Steganography

A comprehensive legal infrastructure needs to be established to enable copyright and ownership protection of digital content by watermarking techniques and to avoid legal attacks that diminish protection by watermarking.

Law enforcements are concerned in the trafficking of illicit materials via web page images, audio, and other files transmitted through the Internet. Steganographic detection techniques are necessary to uncover such activities which can be time consuming. Today, ongoing research in the area of Internet steganography is focused on hiding, recovering, and detecting information in TCP/IP packet headers and other network transmissions.

Glossary

Blowfish algorithm [13] - Blowfish is a 64-bit block cipher (i.e. a cryptographic key and algorithm are applied to a block of data rather than single bits) that uses a key length that can vary between 32 and 448 bits. Blowfish is available for free use and the technology is unpatented and free of license.

Byte - A unit of measure of computer memory. A byte generally represents one character and is made up of eight bits.

Coefficient - In mathematics, a coefficient is a multiplicative factor that belongs to a certain object such as a variable, a basis vector, or a basis function.

Cover image - An image containing an embedded message.

Cypher text - Refers to encrypted data.

Cryptanalysis - The art and science of breaking and decoding ciphertext, usually without prior knowledge of the secret key. Cryptanalysis reveals the secrets hidden by cryptography.

Cryptography - The art of protecting information by encrypting it into an unreadable format, called cipher text. A secret key is used to decrypt the message into plain text.

Encryption - The translation of data into a secret code.

Least significant bit (LSB) - The bit contributing the least value in a string of bits.

Lossless compression - For most types of data, lossless compression techniques can reduce the space needed by only about 50%. No data is lost in the process. For greater compression, one must use a lossy compression technique.

Lossy compression - The decompressed object is identical to the original object before the compression. Some slight shading or smooth anomalies may be observed on some of the decompressed objects.

Lossy compression - Lossy compression technologies attempt to eliminate redundant or unnecessary information. Some amount of data is lost in the process.

Microdots - Refers to text or photographic images that are reduced in size to prevent their viewing by unintended recipients.

Palette - The range of RGB colors used in a computer application.

Plain text - Refers to any message that is not encrypted - also called clear text.

Introduction to Steganography

Steganalysis - The art of discovering and rendering useless covert messages.

Steganalyst - A person whose goal is to detect and read steganography-based documents.

Steganography - A means of overlaying one set of information ("message") on another (a cover).

Stego image - The result of combining the cover image and the embedded message.

Stego text - The result of applying some steganographic process to a plain text (not necessarily encrypted).

TCP/IP - The Transmission Control Protocol / Internet Protocol is the standard protocol suite used on the Internet.

Transient - A transient is a short-lived oscillation in a system caused by a sudden change of voltage or current or load.

References and Bibliography

1. iNFOSSSEC. Cryptography, Encryption and Stenography. [online] 2000. Available at <http://www.infosyssec.org/infosyssec/cry2.htm>; Accessed on 23 June 2004.
2. Wikipedia - The Free Encyclopedia. Steganography. [online] 2004 June. Available at <http://en.wikipedia.org/wiki/Steganography>; Accessed on 23 June 2004.
3. Wikipedia - The Free Encyclopedia. Stegotext. [online] 2004 June. Available at <http://en.wikipedia.org/wiki/Stegotext>; Accessed on 23 June 2004.
4. Plunt. Steganography (hide and seek) Tutorial. [online] Astalavista Group. 2004 January. Available at http://www.astalavista.com//data/hide_and_seek.txt; Accessed on 24 June 2004.
5. Kahn, D. The Codebreakers. The Macmillan Company. New York. 1967.
6. Johnson, N. F., Duric, Z., Jajodia, S. Information Hiding: Steganography and Watermarking - Attacks and Countermeasures. Kluwer Academic Press. Norwrl, MA, New York, The Hague, London, 2000.
7. Johnson, N. F., Jajodia, S. Exploring Steganography: Seeing the Unseen. [online] 1998 February. Available at <http://www.jjtc.com/pub/r2026.pdf>; Accessed on 24 June 2004.
8. Rude, T. J. Steganography - Disappearing Cryptography. [online] CRAZYTRAIN.COM 2000. Available at <http://www.crazytrain.com/rudedude.pps>; Accessed on 25 June 2004.
9. Haldar, V. Steganography and Audio. [online] Available at <http://www.ics.uci.edu/~lopes/teaching/280ubicompW03/students%20presentations/vivek%20haldar.pdf>; Accessed on 27 June 2004.
10. hyperdictionary. Discrete cosine transform. [online] 2003. Available at <http://www.hyperdictionary.com/computing/discrete+cosine+transform>; Accessed on 27 June 2004
11. hyperdictionary. Fast Fourier transform. [online] 2003. Available at <http://www.hyperdictionary.com/dictionary/Fast+Fourier+Transform>; Accessed on 27 June 2004.

Introduction to Steganography

12. Wikipedia. Wavelet compression. [online] 2004 May. Available at http://en.wikipedia.org/wiki/Wavelet_compression; Accessed on 27 June 2004.
13. Wi-Fi Planet. Blowfish. [online] 2003 October. Available at <http://wi-fiplanet.webopedia.com/TERM/B/Blowfish.html>; Accessed on 30 June 2004.
14. Johnson, N. F., Jajodia, S. Steganalysis of Images Created Using Current Steganography Software. [online] 1998. Available at <http://www.jjtc.com/ihws98/jjgmu.html>; Accessed on 29 June 2004.
15. Johnson, N. F., Jajodia, S. Steganalysis: The Investigation of Hidden Information. [online] 1998 September. Available at <http://www.jjtc.com/pub/it98a.htm>; Accessed on 30 June 2004.
16. Wikipedia - The Free Encyclopedia. Steganalysis. [online] 2004 May. Available at <http://en.wikipedia.org/wiki/Steganalysis>; Accessed on 02 July 2004.
17. Kessler, G. An Overview of Steganography for the Computer Forensics Examiner. [online] 2004 February. Available at http://www.garykessler.net/library/fsc_stego.html; Accessed on 02 July 2004.
18. Computerworld. Steganography: Hidden Data. Quickstudy by Deborah Radcliff. [online] 2002. Available at <http://www.computerworld.com/securitytopics/security/story/0,10801,71726,00.html>; Accessed on 02 July 2004.
19. Craver, S., Yeo, B., Yeung, M. Technical trials and legal tribulations. Communications of the ACM [online] 1998 July. Volume 41. Issue 7. Available at <http://portal.acm.org/citation.cfm?id=278476.278486&dl=portal&dl=ACM&idx=278476&part=periodical&WantType=periodical&title=Communications%20of%20the%20ACM>; Accessed on 03 July 2004.
20. Gupta, M. Steganography is more than a tool for spies. [online] Eurescom 2001. Available at <http://www.eurescom.de/message/messageSep2001/stegano.asp>; Accessed on 05 July 2004.