

Introduction to Digital Image Steganography

David P. Holmes

Introduction

Invisible ink, secret codes, hidden messages have always been a fascinating world for most people since they were children. Little did anyone know as children that these were all forms of steganography. The word steganography is derived from the Greeks and literally means covered, "steganos" writing "graphy". [6] There are three basic forms of steganography in use on computers today. The first involves the use of text files that are hidden inside of other text files and when extracted can be read. The second form involves the use of audio files that are hidden inside of other audio files. The last, and the one that I will be discussing here, involves the use of text or digital images that are hidden inside of the other digital images.

Steganography can and will be used for both good and evil purposes. Both government and private industries need to worry about the use of steganography inside of their respective buildings. If a disgruntled employee wanted to hurt the company that he is currently employed with or to try and prove his usefulness to a competitor, it's not too hard to imagine how he could send out company plans or secrets undetected inside of digital images. The same goes for the government but maybe on even a greater scale. If an employee at the FBI, CIA or any government agency that is susceptible to spies wanted to sneak secrets to other governments what better way than to conceal them inside of harmless looking bitmap or gif files. Once extracted the secrets could reveal any number of serious threats. If on the other hand someone would like to keep personal passwords, phone numbers or any number of items readily available via the internet why not hide them in plain sight. A web page could be set up with digital images on it. By using a steganographic program, passwords, phone numbers or other confidential items could be placed inside the image and no one would ever be the wiser.

Abstract

It is my intention here to give a brief overview of the world of steganography. I will start off with the history of steganography, from its earliest uses to how it is used currently today. I will then discuss digital image steganography and how it works along with the three main characteristics. After that I will use one of the easiest steganographic programs, readily available on the internet, and walk thru the steps needed to hide one image inside another. I will briefly discuss the topic of steganalysis, the study of analyzing steganographic programs and the latest government research. Before talking about steganography I must first describe what steganography isn't. Steganography should not be confused with cryptography. Cryptography actually transforms the message that is being transmitted to make it obscure to anyone who may actually intercept the message on the Internet. Unlike cryptography where the message is enciphered steganographic programs actually hide the message within another file, whether it is a text, audio or image file. Before discussing steganography there are a few basic terms that need to be explained. The term "cover" is used to describe the original message, this could be the original digital image file, audio message or text message. The information that is hidden inside of the original message is called the "embedded" data. The term "stego" is used to describe the original data and the embedded data.

History

The history of steganography goes all the way back to the 5th Century. The earliest known writings about steganography were by the Greek historian Herodotus. The historian relates how a slave had a message tattooed on his head by Histiaeus, who was trying to get a message to his son-in-law Aristagoras. Once the slaves' hair was long enough to cover the message he was sent to Aristagoras in the city of Miletus. [5]

Steganography has been used in many different ways throughout time. The simplest was the use of invisible inks that a person could use to send a message to another without anyone else knowing. Different forms of invisible ink have been used to conceal messages throughout time. Some of the more common forms of invisible ink have been, lemon juice, milk, and urine to name a few. If someone wanted to conceal a message they would simply write a message, using one of these inks,

Introduction to Digital Image Steganography

David P. Holmes

on a sheet of paper that already had something written on it. The person receiving the message would then hold the paper over a flame and the transparent message would appear.

One of the earliest known forms of image steganography was done during the early twentieth century. During the Boer War in South Africa, the British were using Lord Robert Baden-Powell as a scout. He was scouting the Boer artillery bases mapping their positions. He took his maps and converted them into pictures of butterflies with certain markings on the wings that were actually the enemies' positions. [5]

During World War II, the Nazis introduced a new concept in espionage it was called the microdot. This simple device could conceal a full typewritten page within the size of a common period. A microdot could hold valuable information such as charts, diagrams and drawings.

Though these are fairly simple forms of steganography they have been used quite effectively throughout time to conceal messages. Computer software has taken steganography to a whole new level and with the power of most pc's today and the software readily available for free or of little cost on the internet an unprecedented rise of steganography is expected.

Digital Image Steganography

With literally millions of images moving on the internet each year it is safe to say that digital image steganography is of real concern to many in the IT security field. Digital images could be used for a number of different types of security threats. In the corporate world the sending of a harmless looking bitmap file could actually conceal the latest company secrets. JPEGs could be used in the government to conceal the latest military secrets. It is believed that the terrorists that died in the 9-11 crash in New York had aircraft configuration plans sent to them hidden inside of a digital image. It is felt that the use of steganography has allowed the terrorists cells to communicate without the fear of being caught. [2]

The use of digital images for steganography makes use of the weaknesses in the human visual system, which has a low sensitivity in random pattern changes and luminance. [7] The human eye is incapable of discerning small changes in color or patterns and because of this weakness text or graphic files can be inserted into the carrier image without being detected. Each graphic image is made up of what is called pixel elements (pixels). Each elements color is determined by the numerical value that it is assigned, ranging from 0 to 255. For example, a typical elements value could be seen as 00000000 or 00000001. The typical digital image is made up of either 8 bit (256 color) or 24 bit (true color) pixels. In a 24 bit graphic file each pixel would be represented by 3 bytes, each being 8 bits long. [4] A white pixel would look like this:

red byte	green byte	blue byte
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

An 8-bit color image could hold around 300 kilobits of hidden data and a 24-bit color image around 2 megabytes. [4] Other factors will also influence the type of image that would be used as the carrier file. Items such as compression type and color variance will need to be considered. The more gray scale that the hidden image has the easier it is to hide. This is because of the HVS's low sensitivity to random pattern changes of less than one part in 30. [7]

There are two distinct groups that steganography software would fall into: Image Domain and Transform Domain. Image domain is the more common of the two and use Least Significant Bit and noise manipulation in order to hide the embedded image within the carrier image. Transform domain are not as common and actually manipulate the carrier image by changing such things as luminance or hue. Transform domain is much less susceptible to damage than Image domain.

Introduction to Digital Image Steganography

David P. Holmes

There are three items here that I will be discussing, the insertion type (how the data is going to be inserted into the carrier file), digital watermarking, and compression types.

Least Significant Bit

Least significant bit (LSB) is the most commonly used type of insertion scheme used currently in digital steganography. It falls into the Image domain group, making it more susceptible to damage. As we already know each pixel is made up of three bytes consisting of either a 1 or a 0. If we had the following pixel:

R	G	B
10101010	01010101	11001100

The LSB would be as follows:

R	G	B
1010101 <u>0</u>	0101010 <u>1</u>	1100110 <u>0</u>

The last bit in the value is known as the least significant bit (LSB), which means that there is enough information in the seven bits preceding it to ensure that the correct color will be established. When the embedded data's bits are substituted into the least significant bits (LSB's) location it will have little to no effect on the images appearance to the human eye. [3] The software used to embed the hidden file will usually zero out the LSB's. Once these bits are zeroed out they are ready to be used by the steganography software to contain whatever data is going to be embedded. [3]

Digital Watermarking

Digital watermarking, also known as "fingerprinting" is in its infancy stage of development. Many different companies are developing different types of watermarking but they all basically work in the same manner. Most of the technology is being developed for use as a copyright protection and licensing tool. Digital watermarking can be used on any digital image, audio file or text file.

Much like the watermarking of old, digital watermarking is used to signify ownership and source authenticity. In the past, watermarks have been used on such items as paper products, letterhead and currency. The watermark is usually not plainly visible and must be held up to the light in order for it to be seen. In digital watermarking the same applies. The watermark assures ownership, authenticity and can be invisible to the human eye.

A digital watermark is a digital signal that is imposed onto the digital file. A digital watermark is basically an image laid over another image and leaves the carrier image in tact. Most watermarks are used redundantly throughout an image. An algorithm is used that inserts the watermark inside the digital image. The "author" can decide whether or not they would like the watermark to be seen. Many companies would choose to have the watermark plainly visible as a way to advertise their products. The intensity of the watermark will determine if the watermark is going to be visible or not, other things to consider would be the placement of the watermark as well as the size. One type of watermarking uses color separation that allows the watermark to only be seen in one of the color bands. In this way the watermark is not seen but if the image is printed where the colors need to be separated the image will appear making the image useless for commercial use. If an image is manipulated or altered it will usually destroy the watermark. The algorithm that is used to insert the watermark is the same one that will be needed to remove it. If the watermark is removed from the image and is completely intact then it could be assumed that the watermark is authentic. If the watermark has been destroyed then a company or "author" could use this evidence in court to show copyright infringement. There have been some problems with using watermarks on the Internet for copy protection. When a tracking

Introduction to Digital Image Steganography

David P. Holmes

service is used to try and protect against illegal copies being used on the Internet the tracking service can only access sites that are not password protected.

Compression

Digital images are either considered lossy or lossless depending upon what type of compression algorithm that is used. Bmp's and Gif's are both considered lossless while Jpegs are of the lossy type. Lossless compression cannot be compressed as much as lossy compression but also is not as susceptible to damage and usually will keep the information intact. This is the preferred method to be used for steganography. Lossy compression has a higher compression ratio but is quite susceptible to damage when the image is altered.

A Sample Program

After searching the internet I came across numerous different programs that can be used as a steganography tool. Most of these programs are freeware or shareware readily available for download. I purposely chose a program that was easy to use and had some sample bitmaps that could be downloaded. On the bitmap download page the author also states the approximate size in kilobits that the hidden file could be. I am going to describe the process step by step of how to download, extract, install and use the program. The following is the link to the authors' web site:

<http://www.blindside.co.uk/> [1]

This link takes you to a page that briefly describes how the program works and contains links to other pages. Click on the following:



And it will take you to the following page:



Blindside has been compiled across a number of platforms to give a good chance that you will be able to run it on your home, or work machine.

Please choose the platform you require from the list:

If you can't see your platform, try compiling the source code yourself.
By downloading this software you agree to the conditions of use.

Note: Blindside is a command line utility, a graphical front end for Windows is under (gradual) development. Please check back soon for a release date.

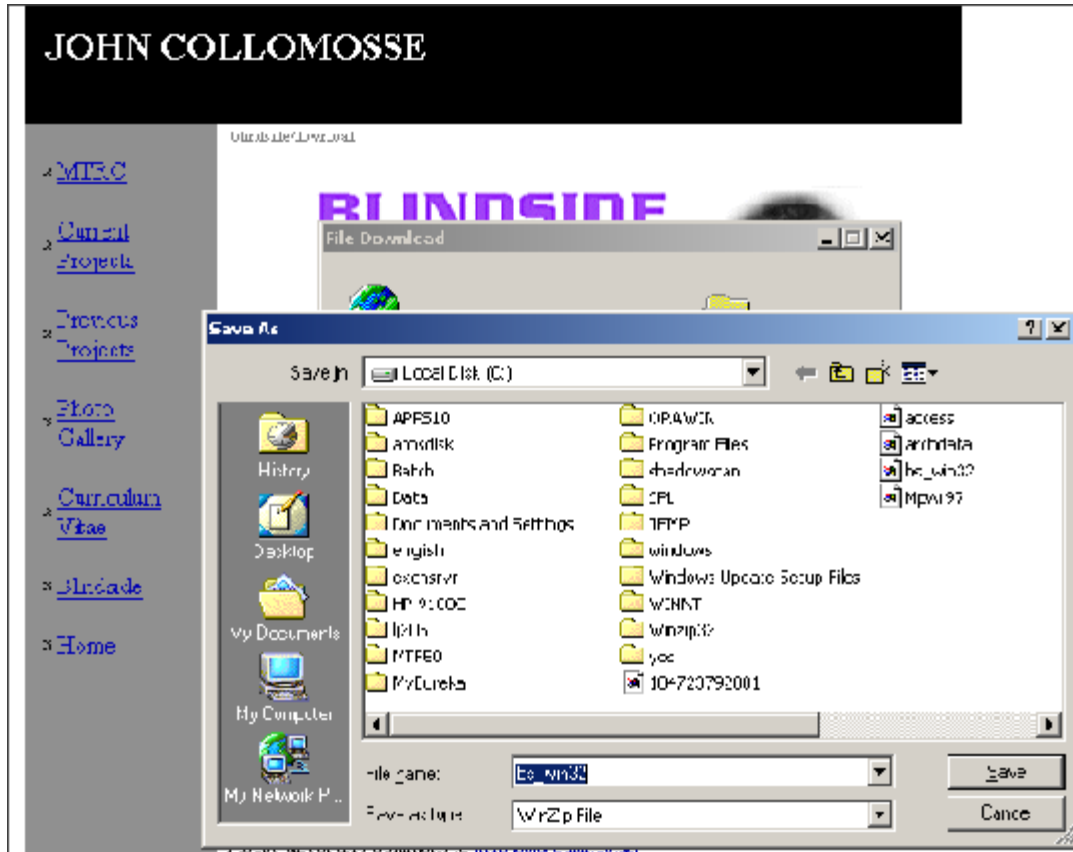
[Click here to return home](#), or [download some sample bitmap files](#)

Introduction to Digital Image Steganography

David P. Holmes

Please direct any comment to mapjpc@bath.ac.uk [1]

Choose the appropriate platform either *nix or 95/NT/2000, and choose download. The following page will pop up.



Choose save and I recommend that it be saved to your c: drive. Go to the location where you saved the program and you should see a file that looks like this.

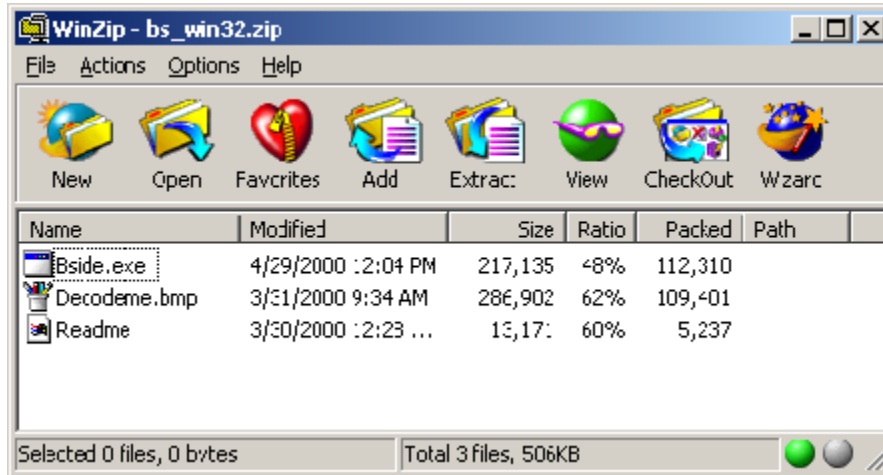


Bs_win32.zip

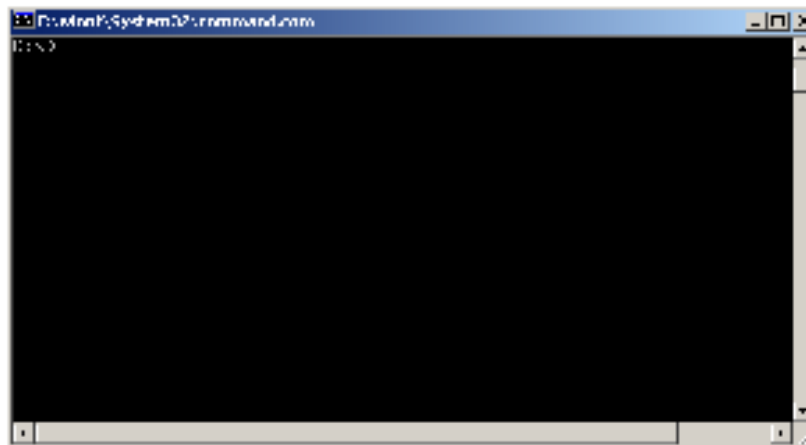
Double click on it and it and winzip will open. Click on I agree and you will see the following:

Introduction to Digital Image Steganography

David P. Holmes



Click on extract, choose new folder and type blindsided. Use windows explorer to locate the blindsided folder, once there copy the Bside.exe to the c: drive. Since this program needs to be run at the command prompt it will be easier to use if it is on the local c: drive. Click on start>run and type cmd in the open dialog box, hit ok. This should open up the command prompt window and you should see something like this.



At the command prompt type bside.exe and you will get a usage list that will explain the different syntax for the program. I have the following files that I am going to use as an example of how the program works. The carrier file is called balloon.bmp that I downloaded off of the blindsided web site. [1] It will hold approximately 11k of hidden data. The file that I am going to hide inside of the balloon.bmp is another bitmap called rushmore.gif and the output will be called sans.bmp.



balloon.bmp



rushmore.gif

The way to do this at the command prompt (assuming that all the files are on the c drive) is as follows:
C:\>bside -a balloon.bmp rushmore.gif sans.bmp

Introduction to Digital Image Steganography

David P. Holmes

```
C:\winnt\System32\command.com
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-1999.

I:\>c:

C:\>bside -a balloon.bmp rushmore.gif sans.bmp
BlindSide BMP Cryptographic Tool - (c) John Collomosse 2000
Release v0.9. All Rights Reserved, contact: ma7jpc@bath.ac.uk

■ Reading bitmap file....OK
■ Image is 1440054 bytes (800x600), 24 bits/pixel
■ Analysing Data Patterns....OK
■ Creating New Archive....OK

Adding.... rushmore.gif

■ Encoding Data....OK
■ Writing result to sans.bmp....OK

Done!

C:\>_
```

This is how it will look while the program is working and the output when it is done. This command will take the rushmore.gif and place it inside of the balloon.bmp file and save it as a file called sans.bmp. Now the sans.bmp file is ready to be sent to someone and provided that they have the bside program they could extract the rushmore.gif from the balloon.bmp. The syntax to remove the file is as follows:

Bside -x sans.bmp

This will extract any hidden files inside of the sans.bmp file. There is a readme file that comes with the bside program that explains how the program work, what can be done with it and the correct syntax to use with the program. When the two files are compared side by side you cannot see any differences, although the sans.bmp has another file imbedded within it.



balloon.bmp



sans.bmp

Steganalysis

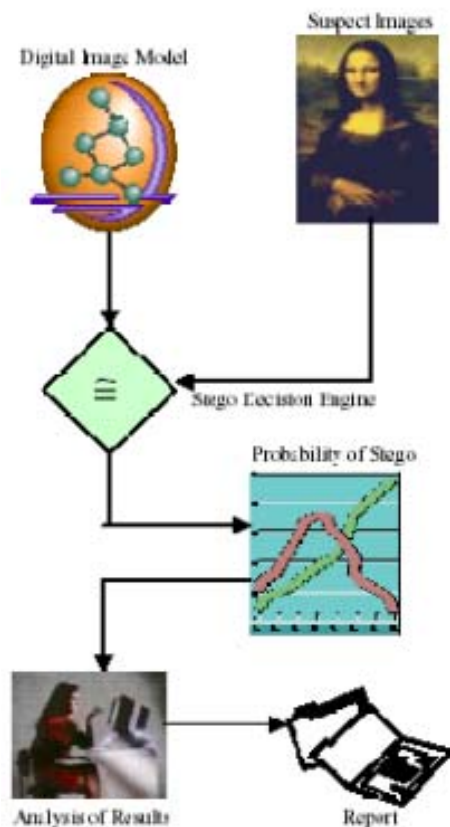
Steganalysis is used to discover and remove hidden files. The problem with current steganalysis programs is that they may remove or discover some files that are hidden that the originator of the image does not want removed, such as watermarking. Many times the same programs or algorithm used to hide a file for legitimate reasons, such as a watermark, would be the same one that is used to hide a file that a user didn't want discovered. Repetitive or obvious signs of manipulation are the easiest way to try and detect hidden files or messages, but this is usually not possible with the human eye. Images that are hidden using Image domain type of software are easier to destroy then the Transform domain type. Many Transform methods try to integrate themselves into the image so that the only way to remove it is to destroy the image itself. One easy way to destroy Image domain type images is to convert their compression type from lossless to lossy. By converting a Bitmap or Gif file

Introduction to Digital Image Steganography

David P. Holmes

to a JPEG file will many times destroy the hidden file within because of the different types of compression.

Scientists and researchers are trying new methods to try and discover ways of detecting hidden files and rendering them useless. The U. S. Government has contracted Wetstone Technologies to work with the U.S. Air Force to research algorithms that can be used to discover embedded files in digital, audio and video format. [8] Each type of steganography software has its own unique signature that is left on a file. Once this signature is known a mathematical expression can be developed and then that expression is used to compare one file with another to see what deviations occurred. Items that the researchers have concentrated on have been color characteristics, saturation, hue, frequency, tone, noise and distortion. Since each one of these items would be altered if a steganographic software program has been used on a file it would leave its digital fingerprint that an algorithm could detect. The following picture shows how Wetstone's Stego- Detection and Recovery Toolkit (S-DART) program would work. [8]



As you can see the image would run thru a steganography detector, the detector would then analyze the image determine the probability of a steganographic fingerprint and create a report. According to Wetstone the detectors can work near real time and if an image is suspected of containing a hidden file but not detected then the image would be put through a more extensive set of detectors. [8] Wetstone's technology is one of the systems that are on the forefront of the steganographic war. This technology is hopefully going to be integrated with existing systems such as firewalls and intrusion detection systems that would automatically suspect or detect embedded messages.

Conclusion

Throughout history there have been many different ways to hide messages whether it was by tattooing a message on a slave's head, hiding secrets by covering them with pictures of butterflies,

Introduction to Digital Image Steganography

David P. Holmes

using microdot technology or secret codes. Its not surprising then that in this digital age there would be new ways for people to send messages or images without others knowing that they are there.

In today's world through there are a few different ways to ensure that a file is not readily accessible to everyone. The most common methods would be encryption, steganography and watermarking. Though these are all somewhat similar technologies they are a world apart in how they do their job. When a file is encrypted it takes the information and puts it into a format that hopefully only the people who have the right keys can open the document. When the document is sent across the Internet anyone who can capture the document can see that it has been encrypted and there is that possibility that the document could be decrypted. By using steganography the file is completely hidden from anyone's eyes. This way of sending hidden documents or images is truly hidden. Steganography can be broken down into two main groups: Image domain and Transform domain. These two groups are differentiated by the way in which the steganography program manipulates and compresses the embedded data. There are two main compression types lossy and lossless. Lossy is the more common of the two but lossless is much more stable. By using a sample program I have shown how simple it can be to embed an image within another image and is basically indistinguishable to the naked eye. At the present time there are literally hundreds of freeware and shareware programs on the Internet. Discovery and destroying covert information within steganographic files is called steganalysis.

I have presented a brief overview of a very exciting and fast paced area of computer security. This technology has many in the security field worried as the possible harm that may be done to both government and private industries. As pc's become more powerful this technology will grow substantially and become much more mainstream. There are already hundreds of steganography programs available that can be used on text, audio and graphic files. The government and many private companies are researching ways to best detect the use of steganography on files. As steganalysis becomes more mature it will be implemented as a standard security tool the way firewalls, virus detection software and intrusion detection programs currently are.

References

1. Collomosse, John, Blindside, <http://www.blindside.co.uk>
2. Fixmar, Robert, "Terrorists and steganography", Zdnet News, 09/23/2001, <http://zdnet.com.com/2100-1107-530751.html>
3. Machado, Romana, "How Stego Online Works", <http://www.stego.com/howto.html>
4. Mendall, Ronald, "Steganography-Electronic Spycraft", Earthweb Networking and Communications, 09/20/2000 http://www.earthweb.com/article/0,,10456_624101,00.html
5. Sellars, Duncan, "Introduction to Steganography", <http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.html>
6. Steganography, http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213717,00.html
7. W. Bender, D. Grhul, N Morimoto, and A. Lu, "Techniques for Data Hiding", IBM Systems Journal, Vol 35, Nos.3-4, February 1996, <http://researchweb.watson.ibm.com/journal/sj/353/sectiona/bender.pdf>
8. Wetstone Technology, Inc., "What You Can't See Can Hurt You...", The Dangers of Steganography", <http://www.wetstonetech.com/technicalpapers.htm>