

Principles of Steganography

Max Weiss

Math 187: Introduction to Cryptography
Professor Kevin O'Bryant

1 Introduction

Although steganography has been a topic of discussion since pre-1995, it is only as of the new millennium that this information hiding technique has caught the eye of the privacy craving public. Established businesses have adopted steganography for covert communication; artists have done the same for intellectual property protection from consumers and advertising agencies. Several large corporations including IBM, Kodak, and NEC have identified steganography as a new market worth investing in.

2 Overview

Steganography: literally “hidden writing.” Nowadays steganography is most often associated with embedding data in some form of electronic media. The difference between steganography and the more commonly used cryptography is that while cryptography scrambles and obfuscates data that can then be accessed publicly (without consequence), steganography conceals the data altogether. Data from a “covert,” or source file is hidden by altering insignificant bits of information in an “overt,” or host file. For example, an algorithm designed to embed an audio file might replace information describing frequencies inaudible to the human ear.

3 Modern Algorithms

Modern steganography identifies two main classification schemes for the sorting of algorithms. The first distinguishes algorithms based on file type. The second, more widely used scheme categorizes based on embedding method.

3.1 Injection (Insertion), Substitution, and Generation Classifications

Insertion-based techniques hide data in sections of a file that are ignored by the processing application and do not modify those bits that determine the contents of a file that are relevant

to an end-user. For example, an insertion algorithm may write data in the comment blocks of an HTML file. Several file types and programs also establish an EOF marker to signify the end of a file. Data written after this marker is nonexistent as far as meaningful content is concerned. However, from a steganography standpoint the EOF can be used to mark the beginning of hidden data. Utilizing an insertion technique changes file size according to the amount of data hidden and therefore can be used to determine the presence of hidden information. Coming across a 5MB HTML file would arouse suspicion, for instance.

In a Substitution-based algorithm, the most insignificant bits of information that determine the meaningful content of the original file are replaced with new data in a way that causes the least amount of distortion. Although file size does not change during execution of the algorithm, the amount of data that can be hidden is limited to the amount of insignificant bits in the file. Higher “quality” files (where applicable) tend to contain more bits of insignificant information.

The injection and substitution algorithms both require a “covert” file that contains the information to be hidden, and an “overt” file that acts as the host. The generation technique requires only a covert file, as it is used to create the overt file. For instance, the covert file can be used to create a fractal image with unique colors, angles, and line lengths. A main flaw of the insertion and substitution techniques is the ability to compare a given file with another instance of the supposedly “same” file. If the file size, MD5 checksum, or anything else is different, it can be assumed that data has been embedded in the file in question. Since the result of a Generation algorithm is an “original” file, the technique is immune to comparison tests.

3.2 Embedding Data in a JPEG Image

Because the JPEG file format is compact and does not significantly degrade the quality of an

image it is in frequent use on the internet. The JPEG format uses a discrete cosine transform (DCT) to identify 64 DCT coefficients in successive 8x8 pixel blocks. Of these quantized coefficients, the least significant bits are used to embed data. Because modifications to these bits affect pixel frequency as opposed to spatial structure (as in GIF images where image structure information is present at every bit layer), no obvious distortion is present.

4 Shortcomings of Steganography

Because steganography has gained popularity only in the past decade, there are many flaws and vulnerabilities that still need to be addressed. Consequently, new steganography technologies are being released with increased frequency.

4.1 Revealing the Existence of Hidden Data

Because steganography modifies an existing file that is most likely in circulation on the internet, a bitwise comparison of a given file with the “same” file suspected of containing hidden information can reveal use of steganography. Additionally, two communicating parties can be easily identified as communicating covertly if files that normally would not be exchanged suddenly are. For example, two business executives frequently exchanging photographs of cars over a period of time could arouse suspicion.

4.2 Rendering Hidden Data Useless

Once a file is identified as possibly containing hidden data, one can either attempt to recover the information if the algorithm is known, or to destroy the data without affecting the quality of the original file. An altered bitmap converted to JPEG would compress the file and remove unnecessary bits of information, therefore removing any hidden data. Converting to any other format may not necessarily cause the image to lose information, but would change the bit composition of the data, making any hidden data unreadable.

5 Practical Steganography: Digital Watermarking

Now that the majority of information takes on a digital form, it has become increasingly necessary to provide a means by which such information can be easily identified to be under

the ownership of an entity. Digital watermarking is a means by which an image is marked such that the owner of a file can rightfully identify any instance of that file to be his own. For example, companies that sell photographs for use in websites or advertisements can embed watermarks in sample pictures to identify whether or not a photograph in use has been paid for or not. There has also been significant recent research into “fingerprinting” (hidden serial numbers or a set of characteristics that tend to distinguish an object from other similar objects). In general, fingerprints can be used to detect copyright violators while watermarks can be used to prosecute them.

5.1 Invisible Watermarks

There are two forms of digital watermarks: visible and invisible. A visible watermark simply overlays a copyright notice on the original image. An invisible watermark is the manifestation of steganography used to embed copyright information into the file itself without altering its visual representation. Steganography can be used to either embed text information into an image, or to alter a pattern of bits to form a uniformly distributed pattern in the image pixels indistinguishable by the human eye.

5.2 Steganography with a Slightly Different Goal

Watermarks do not conform entirely to the paradigms of steganography. While conventional steganography is based on the idea of hiding as much data as possible, digital watermarks tend to be small. Conventional steganography also emphasizes the secrecy of the data to be hidden and transmitted. Even if an invisible watermark cannot be visually identified, the knowledge that one exists is enough to discourage potential copyright violators.

5.3 Defeating Digital Watermarking

As with other files embedded using steganography, images containing digital watermarks can be made “clean” by simply converting the file to another file format, and back to the original format if desired. One publicly available tool written by Fabien Petitcolas (University of Cambridge, Microsoft Research) called StirMark was written to crack several watermarking schemes including PictureMarc, SysCoP, JK_PGS, SureSign,

EIKONA-mark, Echo Hiding, and the NEC method. StirMark can apply a uniformly distributed jitter pattern on an image, which confuses most watermark detecting software. A more sophisticated attack performed by StirMark introduces a slight yet significant distortion in the image emulating the digital-to-analog process on printers, and then the analog-to-digital on scanners. Another test performed by StirMark to evaluate the strength of a watermarking system calculates errors in the file relative to the original. The PSNR test uses the formula $\text{PSNR} (\text{peak signal-to-noise ratio}) = 20 \log (255/\text{RMS Error})$.

6 Steganalysis

Checking for file sizes and suspicious situations may work in detecting the use of steganography, but do not provide any solid evidence. Steganalysis compares the properties of an unaltered file to one that contains embedded information.

6.1 Statistical Analysis of JPEG Images

When one introduces random uniformly distributed noise to any kind of file, the entropy of that file increases. Because embedded data is essentially uniform noise (as far as the image is concerned, because encrypted data to be embedded will have a higher entropy than that of English text), steganography leaves its fingerprint as increased entropy.

When a JPEG image is modified as a result of steganography, certain colors will convert to another color according to the image color table. If a given color A occurs less frequently than B, A will be converted to B more often than B to A. Therefore the difference in color frequencies will decrease and an analysis of color frequency would not yield much information. Instead, an analysis of DCT coefficients should prove more fruitful. A χ^2 test on the image should show distortion from embedded data. An image with hidden data should have a similar frequency for adjacent DCT coefficients. Therefore, one can use the formula

$$y_i^* = (n_{2i} + n_{2i+1}) / 2$$

to compute the expected distribution, where n_i is the frequency of DCT coefficient i . Comparing this expected distribution to the observed distribution

$$y_i = n_{2i}$$

allows one to calculate the chi-square value

$$\chi^2 = \sum_{i=1}^{v+1} ((y_i - y_i^*)^2 / y_i^*)$$

where v is degrees of freedom.

6.2 Dictionary Attacks

In order to verify any assertions one can make from a χ^2 test, it is necessary perform a dictionary attack on the suspected file (it is necessary to perform the χ^2 test first, because when scanning a large number of files for hidden information the χ^2 test will perform exponentially faster than a dictionary attack). Because commercial software embeds data based on a user-supplied password, a brute force attack can be used to prove that hidden information exists. The dictionary attack will cycle through a known set of around 1,800,000 words, phrases, and PIN numbers until the correct one is found.

References

- [1] Eric Cole. *Hiding in Plain Sight*. Wiley Publishing. Indianapolis, Indiana, 2003.
- [2] <<http://www.digimarc.com/watermarking/>>. *Digimarc Corporation*. June 2004.
- [3] Niels Provos and Peter Honeyman. *Detecting Steganographic Content on the Internet*. ISOC NDSS'02, San Diego, CA, February 2002. [August 2001, CITI Techreport].
- [4] <<http://www.petitcolas.net/fabien/steganography/>> *Digital Watermarking and Steganography*. <June 2004>.
- [5] Fabien Petitcolas. *Attacks on Copyright Marking Systems. Vol. 1525 Lecture Notes in Computer Science* <June 2004>.