Steganography detection

Martin Šumák, Roland Cmorik

University of Pavol Jozef Šafárik in Košice
Institute of computer science
martin.sumak@netkosice.sk, rolyc@centrum.sk

Abstract. Main goal of steganography is to hide a conversation, so nobody can know that some conversation exists at all. Many of digital informations have relatively high degree of uncertainty. Images, audio and video files are stored with high precision and offer the space for steganography. Human senses are not as perfect as precision of these files. Making small changes of these files it is possible to hide a message in them, whithout damaging them. New, changed image with a hidden message is for human eye same as original image. Attacker on steganography tries to recognize whether there is a message or not. He can use statistical analysis, that measure the randomness of uncertain bits. In this paper we propose many various ways for hiding information and many important reasons for it. Next we analyse the attacker possibilities for recognizing the hidden message.

Keywords: steganography, hiding information, detection, images

1 Introduction

Steganography is a method how to communicate invisibly. It means that nobody can know that some conversation exists at all. This invisibility is the main difference between cryptography and steganography.

Both cryptography and steganography are used for exchanging a secret information between two sides. The goal of cryptography is to lock up a message in a mathematical safe using a secret key. The key is only secret information and all other information such as cypher text of message or cryptographic algorithm are public. The secrecy is based only on the secret key – the Kerckhoff's assumption. The cipher text is readable only for person who has the secret key. Everybody can know about the existence of conversation, but nobody can read the content of the messages.

On the other hand the steganography tries to hide the conversation's existence. While the cryptography needs a secret key, the steganography needs a cover medium. Cover medium is a message in which the secret message can be hidden. We will focus only on digital messages which can be broadcast by the internet. Good cover medium is an innocuous message whithout any important content. Cover medium must also contain enough amount of redundancy and ucertainty. Even if information in computers seems perfectly defined and certain, it is not true for all digital data broadcast by the internet. The data such as images, videos and sounds contain huge amount of uncertainty. This uncertainty is a space for steganography. Since the image contains an uncertain bits, it is not possible to determine their value and so these bits can be replaced with a secret message. Before

discussing the ways of hiding information, let's motivate it by discussing same reasons for secrecy.

1.1 Reasons for secrecy

People can say that secret communication is necessary for terrorists, drug dealers and many other criminals and also for a war. It is really true, but there are many humanitarian reasons for secrecy too. You can explore job possibilities without revealing where you currently work and potentially losing your job. You can protect you personal information from being exploited by terrorists or money launderers. The police can communicate with undercover agents infiltrating the gangs of bad people. So there are many reasons that protect the solid people.

Moreover, there are many reasons for steganography that have nothing to do with crimes. Some standard data structures may get older over the time and there is need for adding a new unplanned information to the format whithout breaking old software. For example we need to hide autor's name to a fotography, or doctor's and patient's name to the digitized X ray. Steganography is one solution.

Watermarks breed another reason for steganografy. Creators of digital fotos, videos, books and many others may want to add a hidden information into their files. This hidden information is used to keep indentity of author of the file and also to describe the restrictions of using the file. This type of watermarks is so-called strong watermark. Strong watermarks would be found even after the file undergoes a great deal of distortion. Ideally, the watermarks have to be detectable as possible as author want to be identifiably after distortion of the file or as possible as it has a sense. Other watermarks are deliberately made as fragile as possible. These weak watermarks have to disappear even if a minimal change of the file would occur. The presence of such fragile weak watermark is an indicator of unchanged original file. The combination of strong and weak watermarks is good protection before tampering the file.

The hidden information bundled with a document can also contain a digital signature certifying its authenticity. So it is possible to exploit an advanteges from confidentially signed document. In such way only right person can verify the authenticity of the document.

In real world there are many situations when communication between two persons is dangerous because their enemies are listening. Using a steganography a politically weak persons can communicate safety even if the internet network is controled by powerful government.

1.2 Types of steganography techniques

There are a number of different ways to hide an information. All of them exploit some bit of randomness, some bit of uncertainty, or some bit of unspecified state in a file. The simplest technique is to replace the noise in an image, sound or video file. We will focus on the image files. Image file contains the color information about its pixels. The color information is represented by three numbers of intensity of red, green and blue color component. There are eight bits for each red, green and blue

color component so the color of pixels is expressed with extra precision. This precision is not recognizable by human eye and so images offer the space for hiding information. The secret message can be stored in least significant bits of color components. It is a small change of pixel's color and such changed image is for human eye identical with original image. However it seems very secure, it is easy to recognize an existence of hidden information by visual or statistical attacks. More sophisticated way is to spread the information between many files. This way needs larger cover medium (many files), but it is then resistant against the mentioned attacks.

Another way is to mimic some innocuous data. This technique doesn't try to change a cover medium, it tries to create an innocuous message only from the secret message. For example we can create a chess game notation, that represents a random chess game. The bits of secret message can be hidden as order of black or white squares of moves in the game. With some amount of phantasy we can invent many other ways to mimic an arbitrary data.

We can also exploit the randomness of some data. It looks like replacing the noise, but there is important difference. The noise is created by sensors with extra precision and it is not such random as it seems. Randomness is used in many programs deliberately, for example to add realism to scenes of games. We can hide a high amount of information in such scenes. We can exploit anything – the number of monsters, location of the blotches and warts on their skin and so on.

Moreover these techniques can be combined in many different ways. Information can be hidden in a chess game and the game can be hidden in the noise of some file, which is then broadcast. Such combinations make the information more concealingly and resistant before revealing its content. However, only combining various techniques doesn't make hiding message more resistant before revealing its existence.

In chapter 2 of this paper we will discuss some types of attacks on steganography in general and chapter 3 is dedicated to special ways of steganalysis.

2   Attacks on steganography

Steganography algorithms provide stealth and security to information. The degree of stealth and security, is hard to measure. One way to judge the strength of a steganographic algorithm is to imagine different attacks and then assess whether the algorithm can successfully withstand them. Attacking staganographic algorithms is very similar to attacking cryptographic algorithms.

Here's a list of some possible attacks:

- File Only: The attacker has access to the file and must determine if there is a message hidden inside.

- File and original Copy: If the attacker have a copy of the file with the encoded message and a copy of the original, pre-encoded file, then detecting

the presence of some hidden message is a trivial operation. The real question is what the attacker may try to do with the data (destroy hidden information, extract the information, replace...).

- Multiple Encoded Files: The attacker gets *n* different copies of the files with *n* different messages. This situation may occur if a company is inserting different tracking information into each file. Some attackers may try to replace the tracking information with their own version of the information.

- Compression Attack: One of the simplest attacks is to compress the file. Compression algorithms try to remove the extraneous information from a file, and "hidden" is often equivalent to "extraneous".

- Destroy Everything Attack: An attacker could simply destroy the message.

- Random Tweaking Attacks: An attacker could simply add small, random tweaks to all files in the hope of destroying whatever message may be there.

- Reformat Attack: One possible attack is to change the format of the file. Different file formats don't store data in exactly same way (BMP, GIF, JPEG).

3 Steganalysis

The field of steganalysis is usually more concerned with simply identifying the existence of a message (File Only Attack) instead of actually extracting it. This is only natural because the field of steganography aims to conceal the existence of a message, not scramble it. Many of the basic tests in steganalysis will just identify the possible existence of a message. In some cases, hidden bits are spread throughout the file. Some of these algorithms can't tell you where they are, but they can tell that the hidden bits are probably there.

The basic approaches can be divided into these categories:

- Structural Attacks.
- Visual or Aural Attacks.
- Statistical Attacks.

3.1 Structural Attacks - GIF

In many cases, steganographic algorithms leave behind a characteristic structure to the data. If you know what to look for, you can often spot the effects of some algorithms very quickly.

There are many glitches and minor incompatibilities in many of the steganography software. For example Hide and Seek 4.1 for instance requires all

images to be 320x480 arrays of pixels. It is not difficult to fix, but if an attacker see that size of the picture is 320x480 he can deduce what algorithm was used and maybe he knows what attack he can apply.

Hiding information in the least significant bit is simple when each pixel is represented by 24 bits such as BMP. Often there is enough randomness in least significant bits, to make it feasible to store the data.

But, file formats such as GIF allocate 8 bits or fewer by building a palette of selected colors. Each pixel of image is represented by index of color in the palette. Hiding information in GIF format by tweaking the least significant bit can fail dramatically because the palette entries are often not close enough to each other. Entry 01001001 may be a deep blue, while 01001000 may be a hot pink.

Many of the steganographic schemes try to avoid this problem by constructing a special palette. A simple technique is to choose a smaller palette and then duplicate colors that can be used to hide information.

S-Tools creates an optimal 32-color palette for the image. Then it creates 7 near-duplicate colors for each of the 32 colors in the palette. But the final palette have 256 colors.

Palettes like these are easy to detect. When clusters of colors appear in the palette, they are often indicators of bit-twiddling schemes such as this.

In other cases, the algorithms try to order the elements of the palette to place similar colors next to each other. Flipping the least significant bit should not distort the image too much. EzStego, a program written by Romana Machado, uses this technique. The idea is on the figure 1.
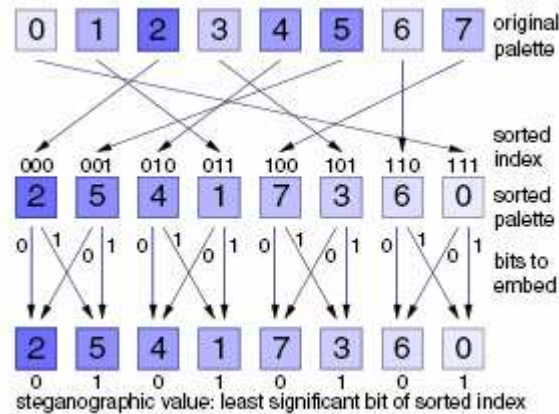

Fig.1   Embedding function of EzStego

After hiding procedure the palette must be unsorted to the original form – any attacker looking at the palette will not see that steganographic program was used. Now the message isn't hiden in the least significant bits of pixels. Receiver of the picture must run the same ordering procedure (it must be deterministic) and he can extract the message using the new ordered indexes of the palette – the least significant bits now

encode the message. But if the attacker knows the sorting algorithm then he also has access to hiden bits.

3.2 Structural attacks - GZIP

Many of the compression algorithms can be tweaked in clever ways to hide information. One of the simplest techniques was used by Andrew Brown when he approached the popular GZIP compression algorithm. The GZIP algorithm compresses data by inserting tokens that point back to a previous location where the data was found. Here's a sample of text:

„The famous Baltimore Oriole, Cal Ripken Jr., is the son of Cal Ripken Sr. who coached for the Orioles in the past."

Here's a sample section that was compressed. The tokens are shown in underlined.

„The famous Baltimore Oriole, Cal Ripken Jr., is the son of <u>(30,10)</u> Sr. who coached for the <u>(73,6)</u>s in the past."

There are two tokens. The first one, *(30,10)*, tells the algorithm to go back 30 characters and copy 10 characters to the current location. GZSteg hides information by changing the number of characters to copy. Every time it inserts a token that requires more than five characters to be copied, it will hide one bit. If the bit is 0, then the token is left unchanged. If the bit is 1, then number of characters to be copied is shortened by one. Here's the same quote with the two bits 11 encoded:

„The famous Baltimore Oriole, Cal Ripken Jr., is the son of <u>(30,9)</u>n Sr. who coached for the <u>(73,5)</u>es in the past."

Greatest advantage is that a standard GZIP program will be able to decompress the data without revealing that information was hidden in the process. A quick analysis, however, could also reveal that data was hidden in such a manner. If you scan the file, you can easily determine which tokens were just one character too small.

3.3 Visual attacks.

Some attacks strip away the significant parts of the image in a way that facilitates a human trying to search for visual anomalies. One common test displays the least significant bits of an image. Completely random noise often reveals the existance of a hidden message because imperfect digitalizers leave echoes of the large structure in the least significant bits.
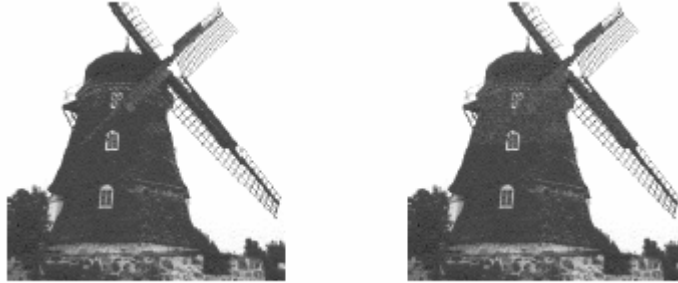
Fig. 2   On the left is the original picture, in the right one is hiden message embedded in upper half of image (EzStego used).

Fig. 3   Least significant bits of pixels (after ordering the palette), it is easy to recognize that there was a hidden message embedded in the right picture.

The disadvantage of visual attack is that they somtimes won't work if there are no objects on the picture with clear shapes – it causes random noise on the least significant bits of the original image. Next disadvantage is that it is hard to automatize the visual attacks.

3.4   Statistical attacks

Hiding messages to least significant bits of some raster image is based on the idea of replacing the noise, which is probably random. Visual attacks show that the value of last significant bits of pixels is not as random as it seems. These bits even well correspond with the entire image. Standard encrypted message that has to be hidden is more random than the bits which it replaces are. This property is the main weakness exploited by statistical attacks on steganography.

Much of the study of mathematical statistics is devoted to determining whether some phenomenon occurs at random. The simplest statistical test for detecting randomness is the χ^2 (chi-squared) test. This test is based on differences between expected number of some event occurrences and number of its real occurrences. Let k be the number of events that can occur. Assume that the events e_0, e_1, ... , e_k would occur with uniform probability (as bits in a message to hide). If there

is n events in a sample sequence at all, the expected number of each e_i event occurrences is then n/k. Let c_i be the number of real occurrences of e_i event, the amount of randomness in the sample sequence of n events is measured with this equation:

$$\chi^2 = \sum_{i=1}^{k} \frac{\left(c_i - \dfrac{n}{k}\right)^2}{\dfrac{n}{k}} \tag{1}$$

High scores indicate an unrandom condition and it is typical for any part of original picture or sound file created by an imperfect set of sensors. On the other hand low scores indicate high degree of randomness and it is typical for files with encrypted hidden message.

In χ^2 test of some image file the events can be the values of least significant bits. In such case there are only two events $e_1 = 0$ and $e_2 = 1$ ($k = 2$). c_1 is then the number of least significant bits of value 0 and c_2 the number of least significant bits of value 1. Finally n is the number of all least significant bits in selected part of the file (or in whole file, because the χ^2 test can be applied in any part of the file). We can also survey the pairs of coterminously bits. In such case there are four events described in table 1.

Event name	Event	Event count	Event probability
e_1	00	c_1	1/4
e_2	01	c_2	1/4
e_3	10	c_3	1/4
e_4	11	c_4	1/4

Tab. 1 Example of four events in χ^2 test.

We can also evaluate the randomness of triplets of bits or of squares of $p \times p$ pixels in image. The χ^2 test can be used in many final variants.

Files with a high amount of hidden information will usually have low scores in this χ^2 test. More natural, unchanged original images often have higher scores.

References

1. Peter Wayner: Disappearing Cryptography, Second Edition - Information Hiding: Steganography and Watermarking, 2002
2. Westfeld A., Pfitzmann A.: Attacks on steganographic systems. IHW 1999
3. Shawn D. Dickman: An Overview of Steganography, 2007

Steganography Detection with Stegdetect - Stegdetect is an automated tool for detecting steganographic content in images. It is capable of detecting several different steganographic methods to embed hidden information in JPEG images.Â A small matter of semantics, here: Steganography on its own isn't encryption, it's obfuscation. While the hidden data may in fact be encrypted, it is not a necessary part of steganography for it to be so. â€" Iszi Feb 14 '11 at 16:12. 1. Steganography tools are must to perform steganography efficiently. All the steganography tools listed here are free to use and you must try one to hide data.Â List of 10 Best Steganography Tools to Hide Data. Last Updated on June 11, 2020 by Amarnath Kolla 4 Comments. Steganography is derived from the Greek language which precisely means the transfer of data with high security. The steganography tools encrypt the data more effectively and efficiently. Steganography is hiding a file or a message inside of another file , there are many fun steganography CTF challenges out there where the flag is hidden in an image , audio file or even other types of files. Here is a list of the most tools I use and some other useful resources. Note : This list will be updated regularly , feel free to pm if you have any suggestions Last update : 29.1.2019. Tools. Steghide. Steganography is the field of hiding messages in apparently innocuous media (e.g. images), and steganalysis is the field of detecting these covert messages. Almost all steganalysis consists of hand-crafted tests or human visual inspection to detect whether a file contains a message hidden by a specific steganography algorithm. These approaches are very fragile â€" trivial changes in a steganography algorithm will often render a steganalysis approach useless, and human inspection does not scale.