A Java Steganography Tool

Kathryn Hempstalk

March 24, 2005

Supervisor: Eibe Frank

Abstract

Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows the existence of the message.[1] Recently, many digital techniques have been developed that perform steganography on electronic media, most notably sound and image files. The outcome of this project is to create a cross-platform tool that can effectively hide a message inside a digital image file.

Introduction

Steganography is a way of protecting information, similar to cryptography and watermarking. Whilst watermarking ensures message integrity and cryptography scrambles a message, steganography hides it. The concept of steganography has been around since Ancient Greece, yet only in the past couple of decades has it been applied to digital information.

Current digital techniques can provide reasonable security for the hidden message, but often leave marks that suggest the cover has been tampered with. Most of these marks are caused by the message being embedded into the cover without any regard to the cover's original content. Therefore, it is highly likely that if the original content was taken into account it would become significantly more difficult to discover whether steganography has been used. It would be interesting to find out whether doing this makes it more complex, if not impossible, to determine the use of steganography on a given object.

Background

Steganography has a long history, which can be traced back to Ancient Greece. In his *Histories*, Herodotus (c. 486–425 B.C.) tells how the head of a trusted slave was shaved and tattooed with a message.[2] The message was hidden after the hair had regrown, at which time the slave could travel without arousing any enemy suspicion. However, it is easy to see that once the enemy is aware of this method of hiding information, they can simply shave the head of every slave that passes through their territory, defeating the system. The same flaw is true of many quicker physical hiding methods such as invisible ink and microdots, where heating every paper and inspecting every document can extract the hidden data.

More recently, steganography has been applied electronic information, where knowledge of the technique doesn't necessarily mean the message can be retrieved. This modern steganography can be broken up into three categories: pure steganography, secret key steganography and public key steganography. Pure steganography is hiding where no information other than the hiding technique is required, as this is sufficient knowledge to be able to retrieve the hidden message. The latter two categories, secret key and public key steganography, rely on the passing of a key (or keys) without which the hidden information cannot successfully be extracted. Most steganographic software available implements the secret key approach due to the facts that pure steganography is simple to break once you know the method and public key steganography is considered to be theoretically impossible (although some models have been suggested as in [3] and [4]).

Despite the many incarnations of steganography methods and the development over time, the result is still the same. A message is transmitted inside some sort of cover in such a way that a 3rd party looking at the cover is not aware that the hidden message even exists.

Overview

There are a large number of tools freely available on the internet that implement steganography on a variety of different electronic mediums, but considering the large quantities of digital images on the internet this project will focus on using colour images as a cover. Digital images often have a large amount of redundant data, and this is what steganography uses to hide the message. Depending on the technique, the cover is typically degraded or "marked" in such a way it is easy to tell that it has been changed. The most cover detrimental way of hiding data in an image is to change the image content – i.e. the colours of the pixels. This technique, whilst crude, hides a large volume of information inside the image. Once implemented, it is not necessarily perceptable to a human eye that the image has been changed, but to a computer simple statistical analysis can pinpoint a changed image from a original one.

Part of the reason why it is so easy for a computer to notice these changes is because these techniques do not look for any features of the cover that can suggest areas that are less noticeable when changed. The algorithms blindy change colours and hence the statistical structure of the image. Ideally a scheme should first analyse the image and work on areas identified as "better to change". This analysis scheme will also have the difficulty of any information that is used must also be available after the message is embedded, so the message can be retrieved.

Once the message has been embedded, and the stego-image output, it is possible to measure the stego-image for any markings that may indicate it has been tampered with. There are many methods that can measure the likelihood of a watermark on a digital image and these processes can also be applied to steganographic images, as a watermark is simply a highly robust steganographic marking. The security of stego-images depends entirely on their ability to go unnoticed, so being able to evaluate characteristics of an image using watermark detection is a clear advantage.

This project will be an investigation into the effectiveness of taking into account the original content of the cover, and will result in a cross-platform tool that can evaluate the effectiveness of it's hiding mechanism. This tool will work on colour images, and should be able to hide a message of any type inside the image.

Objectives

This project has the following objectives:

1. To create a tool that can be used to hide data inside a 24 bit colour image.
2. The tool should be easy to use, and should use a graphical user interface.
3. The tool should work cross-platform.
4. The tool should effectively hide a message using an image degradation approach, and should be able to retrieve this message afterwards.
5. The tool should take into account the original content, to theortically more effectively hide the message.
6. The tool should be able to provide some information as to the effectiveness of the hiding, i.e. it should be able to evaluate the degradation of an image. The analysis used will consist of existing watermarking measures, reimplemented for this tool.
7. The technique should fall under the category of Secret Key Steganography – where without the key the hidden message cannot be retrieved.
8. The tool should be able to encrypt the message before embedding it.

Difficulties

The most difficult part of this project going to be designing an algorithm that is able to find areas that are better to change, and still be able to find these areas when it comes to decode the degraded stego-image. This information is lost when the message is embedded and without passing the cover for comparison it will be difficult to locate these areas again. However, there are parts of the image that will remain unchanged and these can possibly be used as a basis for analysis instead of the areas where the information has been lost. The success of the algorithm is likely to be highly dependent on the actual content of the cover – which may also be cause for concern.

The other obvious difficulty is dealing with digital images. Images come in many different flavours – 1, 2, 4, 8, 16, 24 and 32 bit images are all commonplace and the formats they are written in vary greatly. Java provides support for reading most common formats, such as GIF, JPG, PNG and BMP. The support for writing these formats is native in Java 1.5, but is an optional module in earlier versions of Java. In order to maintain the hidden information the stego-images must be stored in a lossless format, which means that an image should be maintained in a similar bit depth to it’s original format. This will cause each format to need different types of modifications – yet this should be transparent to the hiding algorithm.

Project Schedule

Class Date	Task		Assessment
8 March			
15 March	Battlesteg/Laplace		Proposal
22 March			Proposal Due
29 March	Other algorithms		
5 April	Benchmark		
12 April			
3 May			
10 May			
17 May	Encryption	Interface	Interim Report
24 May			
31 May			
7 June			Interim Report Due
19 July	Evaluation		
26 July			
2 August			
9 August			Conference Abstract
16 August			
23 August			Conference Abstract Due
13 September	Program Refinement		Final Report
20 September			
27 September			
4 October			
11 October			
18 October			Final Report Due

Evaluation

This project's success can be assessed by the following:

- Is the tool functional – i.e. can a message be hidden inside a colour image, written to disk, read in again, and the message retrieved?
- Is the employed algorithm secure?
- Can the tool encrypt data before it hides it?
- Is the program easy to use?

- Does the tool cope with reading common types of digital images?
- Can the tool self-evaluate?

Conclusion

Current electronic steganographic methods are not as necessarily secure due to the size of the data they hide and the lack of concern for the content of the message cover. Taking the cover into account is likely to increase the security of the message by hiding it in a less obvious location. This project will investigate whether taking the cover into account increases the security of the message by creating a cross-platform self-evaluating tool.

References

[1] Anonymous. (2004) "Steganography" Available at: http://en.wikipedia.org/wiki/Steganography (Nov 2004)

[2] Katzenbeisser, S. and Petitcolas, F.A.P (1999) *Information hiding techniques for steganography and digital watermarking.* Artech House, Norwood, MA 02062, USA.

[3] Backes, B. and Cachin, C. (2004) "Public key steganography with active attacks" IBM Research Report, Zurich Research Laboratory, Switzerland.

[4] Von Ahn, L. and Hopper, N.J. (2004) "Public key steganography" University Research Paper, Computer Science Department, Carnegie Mellon University, Pittsburgh, USA.