Implementation of Steganographic Techniques

Danny Friedheim

Computer Systems Lab, 2006-2007

Abstract

Steganography, the idea of hiding messages and data within other pieces of data, can be useful in many real-world applications alongside encryption and other code-writing methods. This project looks at the method of steganography known as least significant bit substitution, in which the pieces of a message are hidden within the actual binary data of a computer file. This particular application uses WAVE audio files as the carrier file, and allows messages of (practically) any length to be hidden within the sound data. The sound file itself looks identical and sounds the same to the human ear, so the existence of the message is very difficult to detect.

Introduction

This project implements a common steganographic technique known as least significant bit substitution. Steganography has many real-world applications in covert operations and communication. The development of an undetectable steganographic algorithm would be a huge breakthrough in the field and would have many implications. With this project, I propose to work towards that, starting with text messages in WAVE files.

Background

Steganography is the science of hiding data in a way that only the recipient

knows of its existence. This differs from cryptography, where the existence of the data is known, but it is not readable. The process of steganography can be achieved with various algorithms designed to undetectably doctor an image, audio file, or other type of file. There is already a diverse field of research about steganography and its various applications in communicating secret messages. Commercial and open source programs that implement steganographic techniques include Stealth, stego, Wnstorm, Snow, FFencode, and many more. One example of an individual algorithm is the F5 technique for embedding messages in JPEG images. The algorithm changes the values of randomly generated bits by a very small amount, and is virtually undetectable by statistical analysis.

This project seeks to design a new program that works flawlessly to embed messages within WAVE audio files. The implications are that messages could be sent inside a file through a monitored path, and could even be intercepted. Without this same program on the receiving end however, a third party could not easily extract the message.

Development

This project uses an implementation of the least significant bit substitution algorithm to hide text messages within WAVE audio files. The result is a command-line program in C++ that acts as both the insertion and extraction mechanism. The program is run initially with a WAVE file and a text message as input, and it outputs

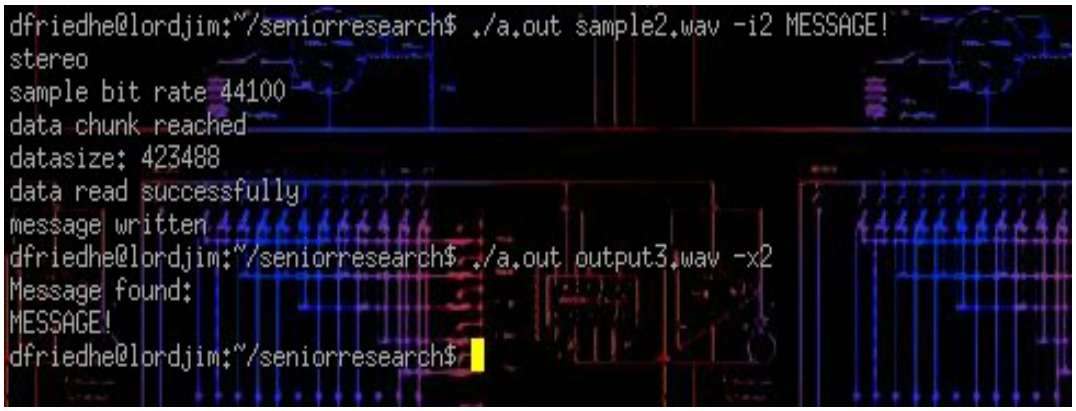
a doctored WAVE file with the message hidden inside it. Then, the program is run with just the doctored file as the input, and the message is extracted and displayed. Therefore, different users who each have a copy of the program would be able to encode and decode each other's WAVE files. However, without the program, it becomes very difficult to even determine the presence of a hidden message, let alone extracting it.

A few limitations were taken into account before implementing the algorithm. The most major restriction was time, because I knew I only had a few months to work with. This affected which goals I set and ultimately the final program, as some things had to be simplified to be finished in time. Another limitation was memory, but this ended up not coming into play. The algorithm is very fast and reasonably efficient, so memory was not a real limitation in actual implementation. However, if the program were to expand and start encoding extremely long messages within very large WAVE files, more efficient memory management might be necessary.

The success of the algorithm is confirmed in a few different ways. First of all, the existence of a message within the doctored file can simply be proven by running the extraction method. This displays the message found within the file, showing that it is indeed hidden. However, another aspect that must be confirmed is that the WAVE file still looks and sounds the same. By playing the files and comparing their sound (by ear) as well as visually inspecting the waveforms (in an audio editing program), we can ensure that the doctored files are indistinguishable, at this level, from the originals.

Also, the length of the messages allowed to be written to a given WAVE file is currently being investigated. Having added a function to insert (long) messages

from text files instead of from the command line, I will be able to determine the maximum length that a message can be before it breaks the algorithm. Then, I can write a function to calculate this maximum message length (in characters) for the given WAVE file (as all files are different).



Sample program output during insertion and extraction

Discussion

The purpose of the project was to develop a program for insertion (and extraction) of text messages within WAVE audio files. The steganographic method used is known as least significant bit substitution and has been implemented with few negative side effects on the outputted WAVE file. So far, the informal methods of confirming the success of the program have proven it to be working well. These methods include comparing the input and output files by ear as well as looking at the respective file sizes. Eventually I hope to find a statistical test that I can run on the program as a form of steganalysis to prove how effective the algorithm is at hiding the data.

Bibliography

R.J. Anderson, F.A.P. Petitcolas, "On the limits of steganography", *Selected Areas in Communications, IEEE Journal on*, vol. 16, issue 4, pp.474-481, 1998.

T. Aura, "Practical Invisibility in Digital Communication," Lecture Notes in Computer Science, Springer-Verlag, Berlin, vol. 1174, 1996, pp. 265-278.

J. Fridrich, M. Goljan, D. Hogea, ``Steganalysis of JPEG Images: Breaking the F5 Algorithm'', *Lecture Notes In Computer Science*, vol. 2578, pp. 310-323, 2002.

N. F. Johnson, S. Jajodia, "Exploring Steganography: Seeing the Unseen", *IEEE Computer*, 1998.