Scott Williams

Computer Science Part II Project Dissertation

STEGANOGRAPHIC FILE SYSTEMS WITHIN VIDEO FILES

Christ's College
University of Cambridge

December 14, 2014

Performa

Name:	Scott Williams
College:	Christ's
Project Title:	Steganographic filesystems within video files
Examination:	Part II of the Computer Science Tripos
Year:	2015
Word Count:	12,000
Project Originator:	Scott Williams
Project Supervisor:	Daniel Thomas

Original Aims of the Project

To investigate appropriate steganographic embedding methods for video and to develop a practical steganographic software package to enable the embedding of arbitrary data within video files via a file system interface. Raw AVI video files should be supported and a variety of steganograhpic embedding algorithms should be available. Basic file system commands should work within the presented logical volume and embedding should occur with no perceivable impact on video quality.

Work Completed

A complete software package has been developed enabling the embedding of arbitrary files within many video formats (including MP4 and AVI) via a file system interface. A total of 9 steganographic embedding algorithms are supported, along with encryption and plausible deniability functionality. Basic file system operations work as expected within the mounted volume and the embedding process operates without any perceivable impact on video quality. Performance of the system is adequate for general use allowing high definition media content to be played directly out of the embedded volume.

Special Difficulties

None.

Declaration of Originality

I, Scott Williams of Christ's College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

I give permission for my dissertation to be made available in the archive area of the Laboratory's website.

Signed:

Date:

Contents

1 || Introduction

Steganography is the art of hiding information in apparently innocuous objects. Whereas cryptography seeks to protect only the content of information, steganography attempts to conceal the fact that the information even exists. This allows steganographic methods to be utilised in countries where encryption is illegal for example, or within the UK where keys for identified encrypted data can be forced to be handed over.

In this project I design and implement a practical steganographic software application - `Stegasis` - which enables users to embed arbitrary files within videos via a file system interface. `Stegasis` can operate with no perceivable impact on video quality and can achieve embedding capacities of upto 200% of the video size. A wide range of video formats are supported[1] along with several steganographic embedding algorithms. Standard encryption algorithms can be used to further protect embedded data and plausible deniability functionality protects users even when the presence of embedded data has been confirmed.

Steganogaphic methods operating on video have had comparatively little attention compared to images and audio. As such, there are few programs currently available which allow data to be steganographically hidden within video. `Stegasis` is the first application to enable the embedding of arbitrary files within videos via a file system interface.

1.1 Motivation

Digital media is ubiquitous on the Internet and high definition video content is now common place on video sharing and social networking websites. Video files of multiple gigabytes in size can reside on users devices without arousing suspicion, providing an ideal hiding place for large collections of sensitive files. Few programs are capitalising on this fact, and those that are, allow the user to embed only a singe chosen file into a specific single video format. As with `TrueCrypt`[2], I believe that a practical system for protecting sensitive files should present the user with a mounted logical volume allowing the use of standard file system operations to create, access and organise embedded data. Furthermore, there exist many commonly used video formats along with many more currently in development. As such, a steganographic program operating on one single video format not only greatly restricts usability, it will require constant development as new video formats inevitably become more popular. Instead, a generic solution applicable to a variety of video formats is preferred.

The many recent global surveillance disclosures show that using certain technologies related to privacy can get you “flagged” by authorities such as the `NSA` - it is no longer the case that simply encrypting data is enough to keep the owner safe.

[1] Including many modern video formats such as `MP4`, `MKV`, `FLV` and `AVI`.

[2] A successful cryptographic program providing on-the-fly encryption and full disc encryption.

2 || Preparation

2.1 Background

In this section steganographic background material, definitions and concepts are introduced. The most important property of any steganographic system is undetectability, that is, it should be impossible to differentiate between ordinary and steganographically modified objects. This requirement is famously formulated within Simmons' prisoners' problem.

Alice and Bob are imprisoned in separate cells and wish to formulate an escape plan. They are allowed to communicate, but all messages must pass through a warden Eve. If Eve suspects the prisoners of secretly communicating their escape plan, the communication channel will be severed and Alice and Bob thrown into solitary confinement. The prisoners attempt to utilise steganograhpy to exchange details of their plan undetected. The steganographic system is considered broken if Eve is able to detect the presence of hidden messages within the prisoners exchanges. It is assumed that Eve has a complete knowledge of the steganographic algorithm being used, with the exception of the stego key, which Alice and Bob have agreed upon beforehand. This is in parallel with Kerckhoff's principle used within cryptography. The warden can be considered one of three categories; *passive*, *active* and *malicious*. A *passive* warden does not modify the exchanged messages in any way, whereas an *active* warden may modify the messages whilst maintaining their original meaning. For example an *active* warden may replace words with synonyms or reorder sentences. If images are being used as a transport medium then an *active* warden may recompress or crop the images. A *malicious* warden attempts to break the steganograhpic system and impersonate the prisoners in an attempt to obtain information.

This project is concerned with only the case of the *passive* warden. As such, any modification of the video files once `stegasis` has embedded data within them, will most likely render the embedded file system corrupt[3].

2.1.1 Preliminaries

A steganographic system consists of a number of individual components:

- A *Cover object* is the original object that the message will be embedded within.
- A *Message* is an arbitrary length sequence of symbols. For this project we consider a message $\mathcal{M} \in \{0, 1\}^n$
- A *Stego key* is a secret key used within the embedded process.
- A *Stego object* is the result of embedding a message inside a cover object.

[3] This unfortunately means utilising video sharing websites such as YouTube and Facebook for distribution is not possible due to them performing compression upon video upload.

Definition 2.1. Steganographic System

Let $\mathcal{C}$ be the set of all cover objects. For a given $\boldsymbol{c} \in \mathcal{C}$, let $\mathcal{K}_{\boldsymbol{c}}$ denote the set of all stego keys for $\boldsymbol{c}$, and the set $\mathcal{M}_{\boldsymbol{c}}$ denote all messages that can be communicated in $\boldsymbol{c}$. A steganographic system[4], is then formally defined as a pair of embedding and extracting functions Emb and Ext,

$$\begin{gathered} \mathit{Emb} : \mathcal{C} \times \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C} \\ \mathit{Ext} : \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M} \end{gathered}$$

satisfying,

$$\forall\ \boldsymbol{c}, \boldsymbol{k}, \boldsymbol{m}.\ \boldsymbol{c} \in \mathcal{C}\ \wedge\ \boldsymbol{k} \in \mathcal{K}_{\boldsymbol{c}}\ \wedge\ \boldsymbol{m} \in \mathcal{M}_{\boldsymbol{c}} \Rightarrow \mathit{Ext(Emb(}\boldsymbol{c},\ \boldsymbol{k},\ \boldsymbol{m}\mathit{)},\ \boldsymbol{k}\mathit{)} = \boldsymbol{m}$$

[4] This is specifically steganography by cover modification.

2.1.2 AVI encoding

2.1.3 JPEG compression

2.2 Existing tools

2.3 Choice of Languages and Tools

2.4 Requirements Analysis

2.4.1 Core Requirements

2.4.2 Possible Extensions

3 || Implementation

3.1 Introduction

3.2 Filesystem

3.3 Steganographic Algorithms

3.4 Extensions

4 || Evaluation

4.1 Satisfaction of Requirements

4.2 Correctness

4.3 Security

4.4 Performance

5 || Conclusions

5.1 Future Project Directions

References

[1] *Steganography in Digital Media.* Jessica Fridrich, 2010.