Part II Project Proposal

STEGANOGRAPHIC FILE SYSTEMS WITHIN VIDEO FILES

Scott Williams, Christ's College
Orginiators: Scott Williams
October 13, 2014

PROJECT SUPERVISOR: Daniel Thomas

DIRECTOR OF STUDIES: Professor Ian Leslie

PROJECT OVERSEERS: Professor Peter Robinson, Dr Robert Watson

1 Introduction and Description of the Work

Steganography is the art of hiding messages within inconspicuous objects - a form of covert communication[1]. Whereas cryptography protects only the content of a message, steganography attempts to conceal the fact that the message even exists. Steganography is particularly useful in countries where encryption is illegal or not suitable, e.g. within the UK, where encryption keys can be forced to be handed over[2].

There exist many freely available programs which offer message hiding functionality within digital media[3]. However, the majority of these programs operate on single image files[4] and therefore impose a hard limit on the size of message you can embed[1]. Many programs also constrain the type of message you can embed to be a simple text string. Video files on the other hand can be several gigabytes in size without arousing suspicion[2] providing an ideal container for multiple (possibly large) sensitive files. A file system interface would enable users to hide any number of files of any type - just by copying / creating files within the mounted volume. For these reasons, the proposed project focuses on steganographic file systems within video files.

I propose to develop an application which allows a file system to be embedded within a user provided video file. The application will also enable mounting and unmounting of video files with contained file systems. As part of the project I intend to explore a number of steganographic embedding algorithms all of which will be selectable within the final application.

An example use of the final product (henceforward referred to as `Stegasis`):

```
# Prepare an existing video file
$ stegasis format -alg=lsb video.avi

# Using stegasis mount we can directly mount the video file
$ stegasis mount video.avi /mnt/volume

# Create a file inside the file system
$ echo "test" > /mnt/volume/test.txt
# Unmount the file system
$ stegasis umount /mnt/volume
```

After doing some initial research on the topic of steganographic file systems, it seems a suitable approach will be to develop a FileSystem in Userspace using the FUSE package[5]. A similar approach was taken within a paper in which a file system was embedded within multiple JPEG images[6]. For the purposes of this project I'll be focusing on uncompressed raw `AVI` video files.

I propose a staged approach to the project where each stage implements an increasingly secure scheme of embedding the file system, for example starting with naïve least

[1] JPEG images for example are typically only a few megabytes in size - limiting the size of files you can possibly embed.

[2] Raw uncompressed `AVI` files are roughly 2GB per minute of footage.

significant bit embedding, showing how this can be broken using statistical analysis and then moving on to more advanced techniques (each method selectable via the –alg flag). The main product of this project - `Stegasis` - as shown above will be a user facing application, enabling all versions of the algorithms described throughout the stages of the project to be run on user provided video files. A number of programs to analyse and break insecure schemes proposed early on during the project could also be produced. This project would tie in nicely with the Part II courses Information Theory and Coding, Digital Signal Processing, and possibly Security II.

2 Resources Required

I will be using the `C++` programming language to develop `Stegasis` of which I have a good amount of experience with. The virtual File System aspect will be implemented using the `FUSE` package. A scripting language such as `Python` or `MATLAB` may also be used to develop some of the steganalysis tools. Raw `AVI` Video files for testing purposes can be created using `VirtualDub`'s video conversion tools[7].

I intend to implement the project on my own desktop computer (running Ubuntu 14.04.1 as well as Windows 7) due to convenience and accessibility. However, there is no reason why development could not happen on the PWF machines, should this be needed. Backups will be taken at regular intervals and `Git`, a revision control system will be used (in conjunction with `GitHub`) to preserve multiple versions of the project stored both locally and in an offsite location.

3 Starting Point

Steganography shares a number of concepts with Cryptography for which an introductory course (Security I) was given last year. I have read the introductory chapters of Steganography in Digital Media by Jessica Fridrich, a number of generic steganography papers and also a few papers specific to Steganographic File Systems.

I have implemented a simple "hello world" `FUSE` virtual file system in `C++` to prove the package works as I would expect.

4 Substance and Structure of the Project

The project will consist of the following sections:

1. Research and investigation into the theoretical aspects of steganography, identifying appropriate embedding algorithms and steganalysis techniques. Investigation into developing a virtual file system and the `AVI` video format.

2. Design and implementation of `Stegasis` providing a variety of steganographic embedding algorithms, allowing raw `AVI` files to be formatted, and the mounted file system to be written to and read from. This section will follow an iterative process

wherein each iteration will propose an increasingly secure embedding algorithm and an attempt to develop a suitable steganalysis technique to break it.

3. Evaluation of `Stegasis` will be based on the following criteria:
    - Correctness: `Stegasis` correctly formats and mounts a provided video file presenting the file system as a logical volume. Files written to the volume should persist between unmounts and subsequent mounts of the same unmodified video file.
    - Usability: The `Stegasis` command line tool should be simple and intuative providing useage details for its functionality and helpful error messages.
    - Performace: The steganographic embedding process should have no noticeable impact on the file system performance i.e. writes to files should not be perceivably slower than a standard HDD[3].

5 Success Criteria

For the project to be considered a success, `Stegasis` should provide the following functionality:

- `Stegasis` should offer a number of steganographic embedding algorithms.
- Given a standard raw `AVI` video file, `Stegasis` should format[4] the video such that it can be mounted.
- Given a formatted video file, `Stegasis` should be able to mount the video and present a virtual file system at a given mount point.
- Standard file system operations including listing files, reading a file, writing to a file and deleting a file should be supported within the virtual file system.
- The above described functionality of `Stegasis` should operate without noticeable visual impact on the video content.

6 Extensions

If there is sufficient time, the following extensions may be attempted:

- Directory Structure: `Stegasis` as described only permits files to be created within the root of the virtual file system. It would be beneficial to allow users to create folder structures as you would expect from a standard file system.

[3] For example, it takes roughly 2ms to read a 1MB file from a HDD.

[4] Format in this case is referring to writing some meta data to the video e.g. which embedding algorithm is being used.

- Audio Usage: `Stegasis` as described only makes use of video image frames to embed the file system. However, a substantial part of an `AVI` file may be the audio data. It would be useful to make use of the audio data to increase the steganographic capacity of the proposed embedding algorithms.

- Video Formats: Unfortunately, raw `AVI` video is (very) uncommon compared with modern compressed video formats such as `H.264 (mp4)` it would be very beneficial for `Stegasis` to operate on a variety of video formats, rather than just raw `AVI`. However, modern video formats are intricate and complex so this may well be outside the scope of this project.

- Evaluation of video artifacts: Quantification of "noticeable visual impact on video content" via a set of human trials possibly achievable by crowd-sourcing through an online website.

7 Timetable

7.1 Michaelmas Term

- ***24th October - 6th November (weeks 3-4):*** Research on the theoretical background of steganography including reading relevant sections of textbooks and academic papers.

- ***7th November - 20th November (weeks 5-6):*** Investigation of appropriate steganographic embedding algorithms suitable for video files. Investigation of the `AVI` video format and the `FUSE` package.

- ***21st November - 4th December (weeks 7-8):*** Implementation of `Stegasis` only offering the simple LSB embedding algorithm and file system functionality.

7.2 Winter Vacation

- ***5th December - 18th December:*** Implementation of more advanced steganographic embedding techniques and integration of these into `Stegasis`. Development of steganalysis tools to break proposed embedding schemes.

- ***19th December - 1st January:*** Christmas holiday.

- ***2nd January - 16th January:*** Continuing work on more advanced steganographic embedding and steganalysis techniques.

7.3 Lent Term

- ***16th January - 22nd January (week 1):*** Polishing of `Stegasis` and source code (not yet including extension work) - core project should be finished at this point. Write progress report and prepare for the progress presentation.

- ***23rd January - 12th February (weeks 2-4):*** Evaluation of the core project. Identification and implementation of promising extension tasks.
- ***13th February - 26th March (weeks 5-10):*** Work on the dissertation write-up, completing a draft for submission to my supervisor.
- ***27th March - 23rd April (Easter Vacation):*** Revision of dissertation addressing supervisors comments. Dissertation should be ready to submit by 23rd April.

References

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

[2] *Contested UK encryption disclosure law takes effect.* Washington Post, 2009-01-05.

[3] *http://en.wikipedia.org/wiki/Steganography_tools.* Tools Comparison.

[4] *Detection of hidden information, covert channels and information flows.* N.F. Johnson and P. Sallee, April 2008.

[5] *http://fuse.sourceforge.net/* FUSE: Filesystem in Userspace.

[6] *http://www.sav.sk/journals/uploads/0124135007Jo-Kos.pdf*
Steganographic file system based on JPEG files, Matus Jokay, Martin Kosdy, 2013.

[7] *http://www.virtualdub.org/* VirtualDub video processing utility.