

1 StegAware

Copyright© University of Kent, Prof. Julio Hernandez-Castro, Thomas Sloan, and Dr. Darren Hurley-Smith. This work is funded by the Horizon 2020 RAMSES project.

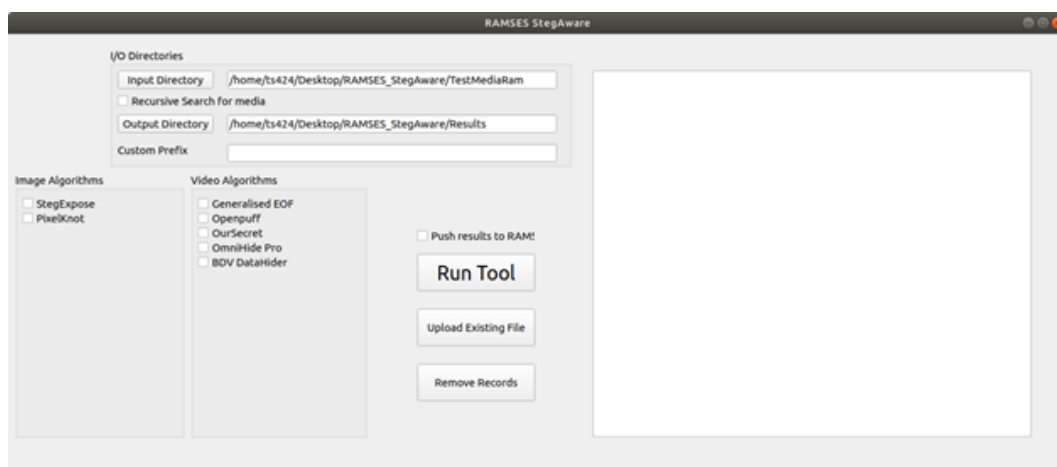
1.1 Description

This forensic and steganalytic framework provides comprehensive analysis of image and video media and is part of the RAMSES platform. StegAware is a steganalysis tool to detect the presence of steganography over multimedia content (image and video). StegAware is capable of detecting a diverse range of steganographic tools and techniques. The image steganalysis component uses StegExpose for LSB steganalysis alongside a series of signature detection methods. The video steganalysis component uses signature steganalysis and generalised detection methods to identify steganography across a range of well-known video steganography tools. In addition to this, we have added a feature for metadata forensics and have added functionality to integrate StegAware with the RAMSES platform.

1.2 Quick Start Guide

The following steps can be used for a quick setup of the tool on an installation of Ubuntu 18.04.

- 1 \$cd Desktop
- 2 \$sudo apt install git
- 3 \$git clone https://github.com/UoK424/RAMSES_StegAware.git
- 4 \$sudo apt install python3-pip
- 5 \$sudo bash ./setup.sh
- 6 \$sudo chmod -R 755 <Directory>
- 7 Run StegAware from within the directory with *\$python3 main_menu.py*



1.2.1 Prerequisites

Below is a list and explanation of prerequisites for the StegAware Tool. Most of these will be managed by the 'setup.sh' script that you must initially run as outlined in the Quick Start Guide.

The tool is available as open-source on GitHub at the following address.

https://github.com/UoK424/RAMSES_StegAware

Ubuntu – StegAware has been tested over the stable 18.04 version of Ubuntu. We, therefore, recommend using this.

Python – Many features of the tool rely on python to run. This should be installed by default on recent distributions of Ubuntu.

ExifTool – The metadata forensics feature of this tool makes use of 'ExifTool'. This is a free, open-source program for reading, writing, and manipulating file metadata. The RAMSES steganalytic tool uses these features to extract relevant metadata from video and image files. This can be installed from the following command:

```
libimage-exiftool-perl
```

MP4Reader – Used in the OpenPuff detection script for mp4 metadata extraction

```
mp4v2-utils
```

Java – The StegExpose feature of StegAware is a Java based tool. This installation will be required before StegAware can be run. There are multiple versions that can be used. For example:

```
openjdk-11-jre-headless
```

Hashing Function – StegAware uses the SHA3 hashing function to generate a unique ID of files that are processed.

```
install libdigest-sha3-perl
```

1.3 Features

The StegAware interface provides the user with a wide range of steganalytic capabilities. A full breakdown of each feature is given below.

- **Image Algorithms:** The user can perform steganalysis over PNG and JPEG image files using both statistical and signature-based attacks. These algorithms are discussed in Section 11.7.
- **Video Algorithms:** The user can perform steganalysis over MP4 video files using signature-based attacks. These algorithms are discussed in Section 11.6.
- **Input Directory:** By default, this is set to the ‘/Ramses/TestMediaRam’ subdirectory which contains separate locations to place image and video files that will be processed by StegAware.
- **Output Directory:** By default, any results are saved to the /Ramses/Results Subdirectory. There will be a .csv file containing both steganalytic and forensic metadata results.
- **Recursive Search for media:** This feature will allow the user to perform a recursive search for images and videos on the file system. With this feature, the user can change the default location from ‘/Ramses/TestMediaRam’ to any suitable location for the tool to process files located in multiple subdirectories.
- **Push Results to RAMSES Platform:** This feature will enable the user to upload results to the RAMSES platform upon running. This will require authentication using your given credentials.
- **Upload Existing File:** If you have a CSV file that you’d prefer to upload to the RAMSES platform without running any steganalytic tests, you can use this feature to do so.
- **Run Tool:** Once you have selected any suitable steganalytic scripts, simply use the “Run Tool” feature to begin processing and analysis of image/video files.
- **Live Feedback Window** This will display processing information as certain steganalytic tests are being performed.
- **Remove Records** Once authenticated, the user can remove associated records that have been submitted to the RAMSES platform. It should be noted that the user can only delete records that they have submitted
- **Upload Existing File** CSV results files can be submitted to the RAMSES platform without needing to run any steganalytic tests. This feature will allow the user to submit a chosen CSV file to the platform.

1.4 Usage

StegAware has a graphical user interface but is initially run from the Linux command line. The user should navigate to the Ramses directory and run the source script from Python.

```
$ python3 main_menu.py
```

You will be presented with an interface as shown below (Figure 1). The GUI is a PyQt5 interface, which can be used to perform steganalysis locally or push results to the RAMSES platform. StegAware boasts a strong capability for both image and video-based steganalysis using signature and statistical methods to detect the presence of steganography.

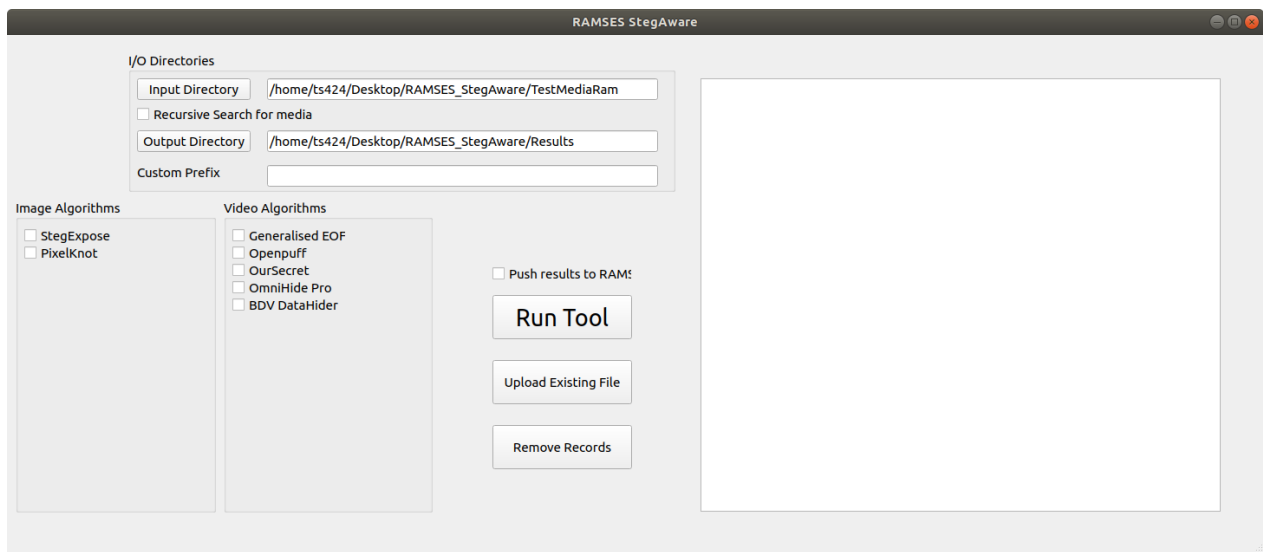


Figure 1: StegAware Interface

You can select any checkbox to carry out steganalytic tests that suit the requirements of your investigation. Using the 'Run Tool' feature will process files in a given Input Directory and provide data in the 'Live Feedback Window'. This is shown in Figure 2.

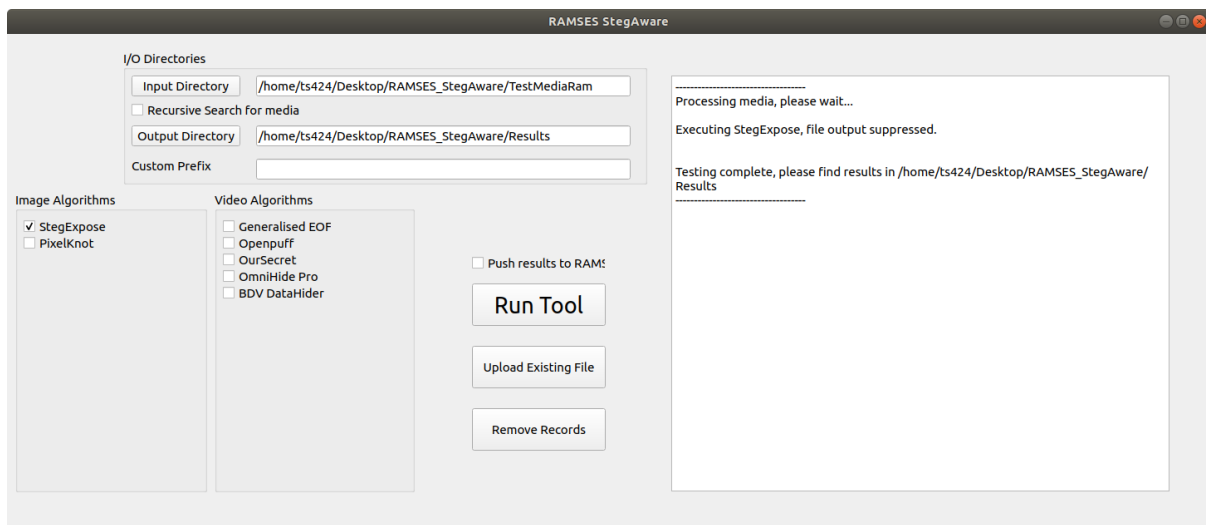
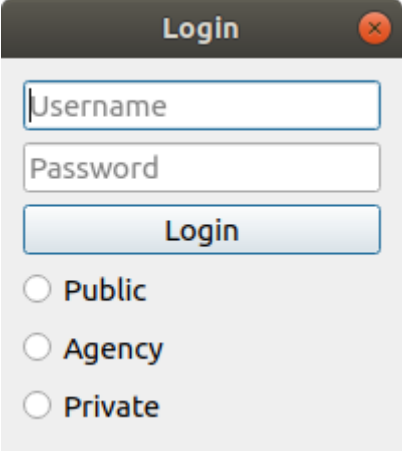


Figure 2: Live feedback window

There are two main ways in which the tool can be run. First, the user can run the tool offline. With this, the user can run steganalytic tests that will save results to the '/Results' subdirectory. The user can then decide if they wish to upload those results at a later point, or keep them offline.

The second main feature will run chosen steganalytic tests and automatically push the results to the RAMSES platform. This will first require authentication, using credentials that each user will have been given for access to the RAMSES platform. The authentication window (shown in Figure 3) will appear when running the tool using the checkbox 'push results to RAMSES platform'.



The image shows a standard login dialog box. At the top is a title bar with the word 'Login' and a red close button. Below the title bar are two text input fields, the first labeled 'Username' and the second labeled 'Password'. Underneath these fields is a button labeled 'Login'. At the bottom of the dialog are three radio buttons, each followed by a label: 'Public', 'Agency', and 'Private'. The 'Public' radio button is currently selected.

Figure 3: Authentication

1.5 Results

Any results obtained will be written to /Ramses/Results/ if they have been stored locally. In this subdirectory, a file is given that merges the results of both the steganalytic tests and the forensic metadata tests. The user will be given comprehensive details for each file scanned as shown in below.

	A	B	C	D	E	F	G	H	I	J	K	L	M
	File Name	File Size	Image Size	File Type Extension	File Access Date	File Modify Date	Duration	SHA3 512 Hash	Date	UID	Steg_Algorithm	Steg_Present	Steg_Signature
2	MP4 (1).mp4	96 MB	1920x1080	mp4	2018:08:24 14:03:2015:07:17 05:49:00:31			4a3dd07205e18823	2018:08:24 14:03:51	954084f7ba5295e	None	None	None
3	MP4 (10).mp4	118 MB	320x240	mp4	2018:08:24 14:03:2014:01:22 13:09:02:21			f06a988c3cf2175e3d	2018:08:24 14:03:51	b6517412cfa315	None	None	None
4	MP4 (11).mp4	108 MB	320x240	mp4	2018:08:24 14:03:2014:01:27 11:52:02:41			57dd569e65d5d9e42	2018:08:24 14:03:51	a223b8db33a189	None	None	None
5	MP4 (12).mp4	108 MB	320x240	mp4	2018:08:24 14:03:2014:01:23 16:19:02:48			4eace6202db71345b	2018:08:24 14:03:51	b4225346a7a0cf0	None	None	None
6	MP4 (13).mp4	109 MB	320x240	mp4	2018:08:24 14:03:2014:01:23 11:37:02:02			bfb7383efdb3a5116	2018:08:24 14:03:51	09d7d877eba8269	None	None	None
7	MP4 (14).mp4	109 MB	320x240	mp4	2018:08:24 14:03:2014:01:22 13:16:02:02			954b9e55a99dc50a8	2018:08:24 14:03:51	3c5b1b1b47e35d9	None	None	None
8	MP4 (15).mp4	116 MB	320x240	mp4	2018:08:24 14:03:2014:01:22 15:16:02:47			da88450d264cf42230	2018:08:24 14:03:51	a068c44ab1d09e	None	None	None
9	MP4 (16).mp4	115 MB	320x240	mp4	2018:08:24 14:03:2014:01:23 16:33:02:31			7ea06045fbb0c1e8b3	2018:08:24 14:03:51	4145e7118ed6db	None	None	None
10	MP4 (17).mp4	118 MB	320x240	mp4	2018:08:24 14:03:2014:01:23 15:52:02:09			cacb59f238984bd54	2018:08:24 14:03:51	21cb19d31552da0	None	None	None
11	MP4 (18).mp4	54 MB	320x240	mp4	2018:08:24 14:03:2014:01:23 13:39:02:13			a3928936b79abb68	2018:08:24 14:03:51	00f52229740756	None	None	None
12	MP4 (19).mp4	62 MB	320x240	mp4	2018:08:24 14:03:2014:01:23 13:54:02:15			88452da3ae14cc5ec	2018:08:24 14:03:51	31b66ed53ab017	None	None	None
13	MP4 (2).mp4	201 MB	320x240	mp4	2018:08:24 14:03:2014:01:23 13:51:04:46			7fb964e8b36248917	2018:08:24 14:03:51	3626bf6ba379df7	None	None	None
14	MP4 (20).mp4	47 MB	320x240	mp4	2018:08:24 14:03:2014:01:23 13:33:02:11			8a234481932559b3	2018:08:24 14:03:51	df2b0300028391e	None	None	None
15	MP4 (21).mp4	181 MB	320x240	mp4	2018:08:24 14:03:2016:12:20 18:17:04:50			4a9444108853f8b210	2018:08:24 14:03:51	b74e4e374020143	EOF Steganography	yes	EOF Injection
16	MP4 (22).mp4	92 MB	320x240	mp4	2018:08:24 14:03:2016:12:20 18:19:02:23			443bb57d5477c0e84	2018:08:24 14:03:51	293c692e5572d00	EOF Steganography	yes	EOF Injection
17	MP4 (23).mp4	235 MB	320x240	mp4	2018:08:24 14:03:2016:12:20 18:19:02:23			c44fc2cc8943370be1	2018:08:24 14:03:51	611ecc037885253	EOF Steganography	yes	EOF Injection
18	MP4 (24).mp4	122 MB	320x240	mp4	2018:08:24 14:03:2016:12:20 18:20:02:16			a1f32bad0a9e44a70e	2018:08:24 14:03:51	033f427d9e41e8e	EOF Steganography	yes	EOF Injection
19	MP4 (25).mp4	116 MB	320x240	mp4	2018:08:24 14:03:2016:12:20 18:20:02:49			4c3d8db8b0e9417b	2018:08:24 14:03:51	431ed372f04c2969	EOF Steganography	yes	EOF Injection
20	MP4 (26).mp4	223 MB	320x240	mp4	2018:08:24 14:03:2016:12:20 18:22:02:39			c315a1863812b88f	2018:08:24 14:03:51	79ab694e3b7d0cf	EOF Steganography	yes	EOF Injection

Figure 4: Results CSV

Alternatively, if the results have been uploaded to the RAMSES platform, they can be accessed by logging in to the platform through your browser. Here you will have various filtering options to easily identify user submitted data.

RAMSES						
OSINT	DARKNET	RANDOMWAVE	BTCOIN TRACKER	BANKING TROJAN ANALYZER	MULTIMEDIA FORENSICS	MALWARE
Steganalysis			Forensics			
Search...						
FILTERS						
Steganography presence:						
All						
Privacy level:						
All						
Reset						
1 2 7						
Steganography present						
Date						
User						
Agency						
Privacy level						
Format						
Campaign						
Jun 22, 2019						
Darren.smith						
Other						
private						
JPEG						
testMalware						
Jun 22, 2019						
Darren.smith						
Other						
private						
JPEG						
testMalware						
Feb 15, 2019						
Hugobom						
Policia Judicialia						
public						
2019:02:15 11:50:20+01:00						
None						
Feb 15, 2019						
Hugobom						
Policia Judicialia						
public						
2019:02:15 11:50:20+01:00						
None						
Jun 4, 2019						
Darren.smith						
Other						
public						
JPEG						
testMalware						
Jun 22, 2019						
Darren.smith						
Other						
public						
JPEG						
testMalware						
Jun 22, 2019						
Darren.smith						
Other						
public						
PNG						
testMalware						

Figure 5: Results pushed to platform

When results are submitted to the RAMSES platform, they will appear under the Multimedia forensics tab. Under this, there are two choices for either steganalysis or forensics. Results submitted by StegAware will be viewable under the Steganalysis tab. As shown in Figure 5.

Individual results can be selected which will redirect to a new page displaying a comprehensive breakdown of the steganalytic results. The breakdown can display information including: Hash, Owner, Date of Submission, File Format, Malware Campaign, Steganographic Presence, Steganographic Signature (if any), and Steganographic Algorithm (if present). These details can be seen in Figure 6.

Steganalysis result details		
Hash	a4e9ae0531324587e573e06bdffe2855eacd3897	
Owner	darren.smith from Other agency	
Date	Sun Jun 23 2019	
Format	PNG	
Duration	00:00:00	
Malware campaign	testMalware	
Steganography	Presence	Present
	Signature	Statistical Steganalysis
	Algorithm	LSB Steganography

Figure 6: Detailed breakdown of results

1.6 Overview of Video Steganalysis Scripts

The video steganalysis feature of this tool is built upon a series of detection scripts. Each detection method performs signature analysis over MP4 files and for each case, we have identified and implemented highly accurate signatures to detect the presence of steganography and trace its use to a specific tool. This tracing feature can provide useful insights for forensic investigators and practitioners. The following video steganography tools and techniques can be detected through the video steganalysis features of StegAware.

1.6.1 OpenPuff Detection

OpenPuff steganography is performed in the metadata channel of a video file. Flags, a component of atoms that describe the structure of an MP4 file are typically null values. These values are modified by the OpenPuff embedding algorithm to hide part of the secret information.

▶ Decoding Time to Sample Box		▶ Decoding Time to Sample Box	
Start offset	540 (0X0000021C)	Start offset	540 (0X0000021C)
Box size	24 (0X00000018)	Box size	24 (0X00000018)
Box type	stts (0X73747473)	Box type	stts (0X73747473)
Version	0 (0X00000000)	Version	0 (0X00000000)
Flags	0 (0X00000000)	Flags	139266 (0X00022002)

Figure 7: OpenPuff flag detection

Flags appear in many atoms throughout the MP4 file structure. However, the secret data embedded into these fields are encrypted and therefore, pseudo-random. Because of this, it is difficult to construct a consistent signature. Instead, a detection algorithm can be constructed by looking at flag fields and observing a 'null' or 'modified' value. A count can then be introduced to determine with high accuracy if a file has been modified by OpenPuff steganography. A result for this may appear as:

```
$ /filepath/filename - OpenPuff Steganography Detected! 30 5
```

The numbers in this example refer to the count of positive flags identified vs null flags. A higher weight of positive flag modifications will assume the presence of OpenPuff steganography with a higher rate of accuracy. The functionality of the OpenPuff program is shown below (Figure 8).

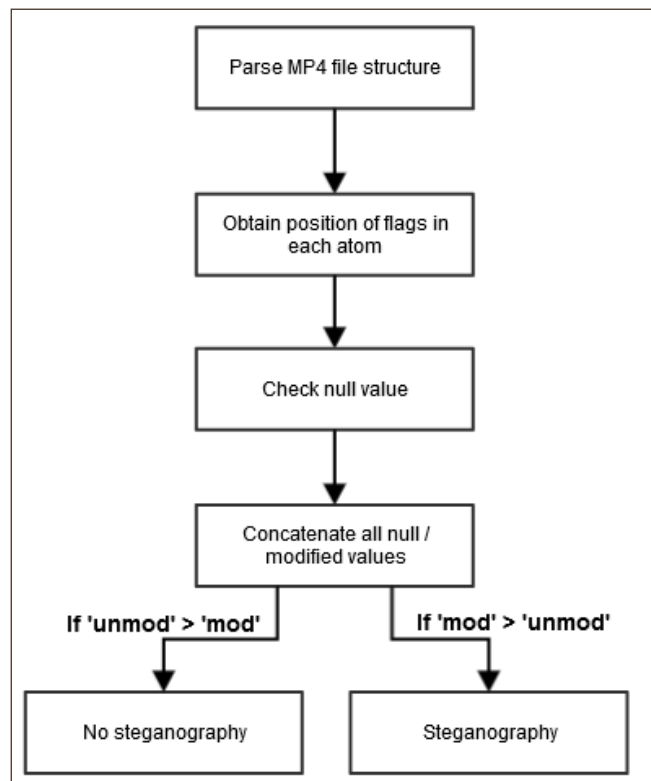


Figure 8: process of OpenPuff flag detection

1.6.2 OurSecret, BDV DataHider, and OmniHide Pro

OurSecret, BDV Datahider, and OmniHide Pro all use EOF (end of file) steganography to hide a secret message. This is achieved by creating empty space at the end of a video file and filling it with secret message data. This method is often used because it is simple to implement and does not affect the image stream or audio stream of a video file. In each case, a unique signature can be identified among the embedded data that proves the existence of steganography and traces the embedded data to the tool that performed the steganographic function.

Detection is performed by simply looking for each known signature in any given video file. This process is illustrated in Figure 9.

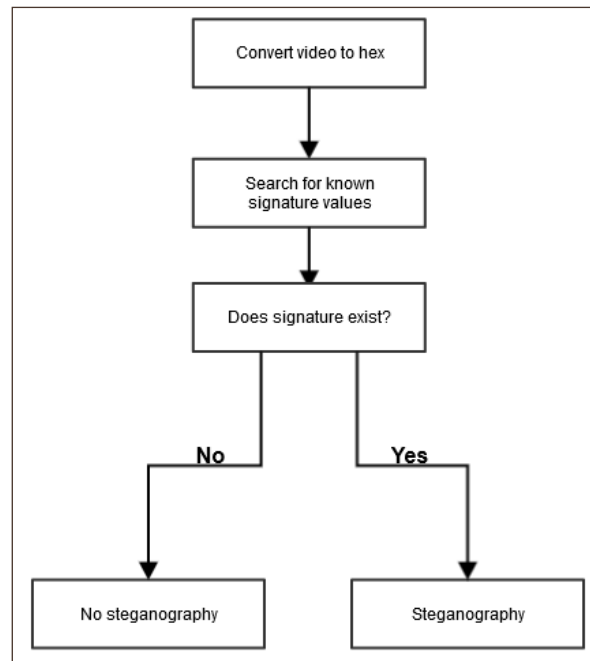


Figure 9: Process of EOF signature detection

1.6.3 Generalised EOF Detection

This feature of the SEEK framework provides multiple uses. Firstly, it can be a generalised method to detect quickly the known EOF steganography tools without giving any tool tracing. Secondly, it can be useful for detection new steganography tools that implement EOF injection that we may not have discovered. Finally, it is significantly useful when running all scripts automatically to reduce the sample size and reduce run time.

Video steganography tools that perform EOF injection modify the structure of MP4 files. The newly hidden data appears as an arbitrary atom that can be detected by error parsing, as it will not match the typical features of an atom. This process is illustrated in Figure 10.

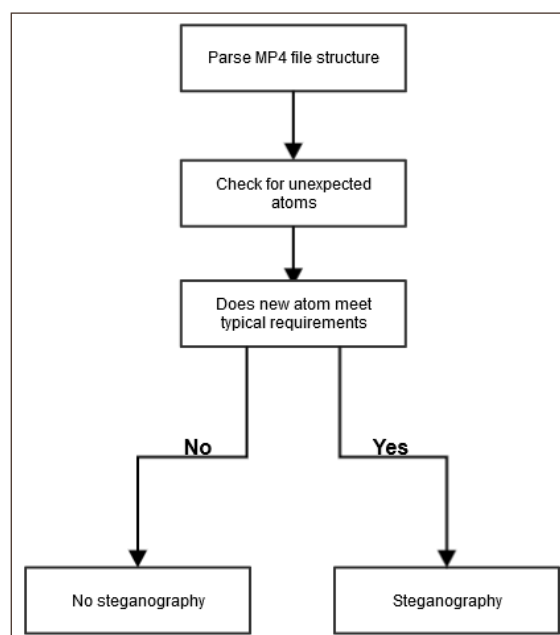


Figure 10: Process of generalised EOF detection

1.7 Overview of Image Steganalysis Scripts

The image steganalysis component of this framework has two unique detection methods. The first is StegExpose, the second is Pixelknot detection.

1.7.1 StegExpose

StegExpose is a command line tool specialised in detecting LSB (Least Significant Bit) steganography through a series of highly accurate detection methods, including RS-Analysis (Regular and Sample Group), SPA (Sample Pair Analysis), Chi-Square, and Primary Sets. This allows for detection across a significantly large number of image steganography tools that use the LSB method and estimate given size of the embedded message. However, it will not trace the use of steganography to any particular tool. The performance results are shown in Figure 11.

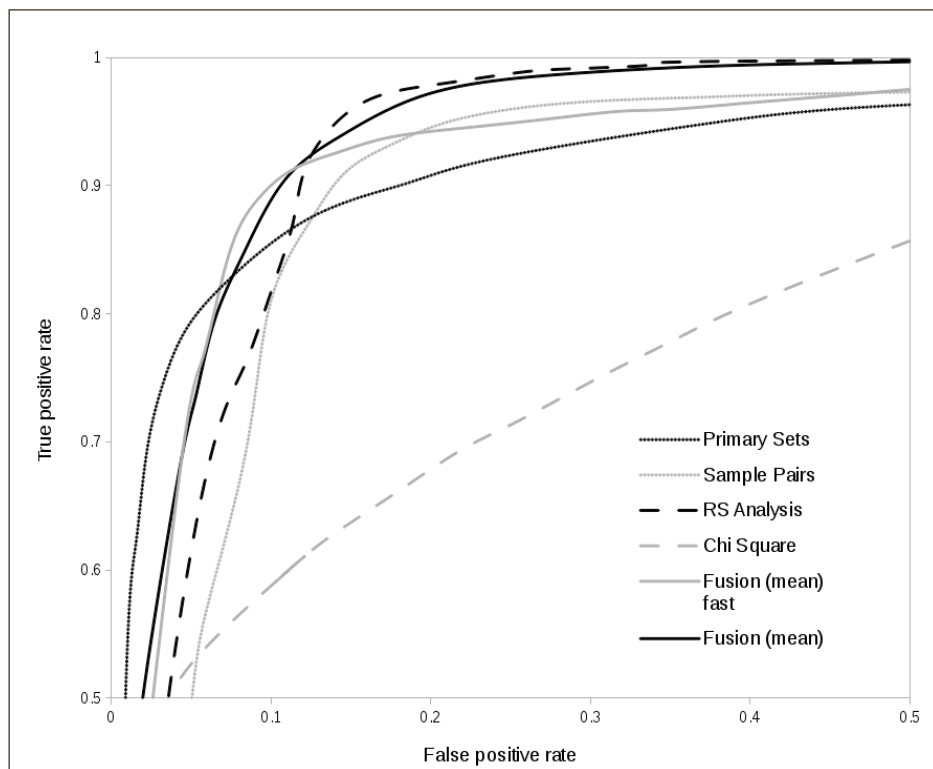


Figure 11: Results for StegExpose

1.7.2 Pixelknot Detection

Pixelknot is as well-known and highly regarded image steganography tool for mobile devices. It uses F5 steganography to hide messages in DCT (Discrete Cosine Transform) coefficients and increase the embedding efficiency via matrix encoding. Our detection feature uses signature steganalysis to identify a consistent sequence of bytes within stego-objects and trace the use of steganography to the Pixelknot tool. This process is illustrated in Figure 12.

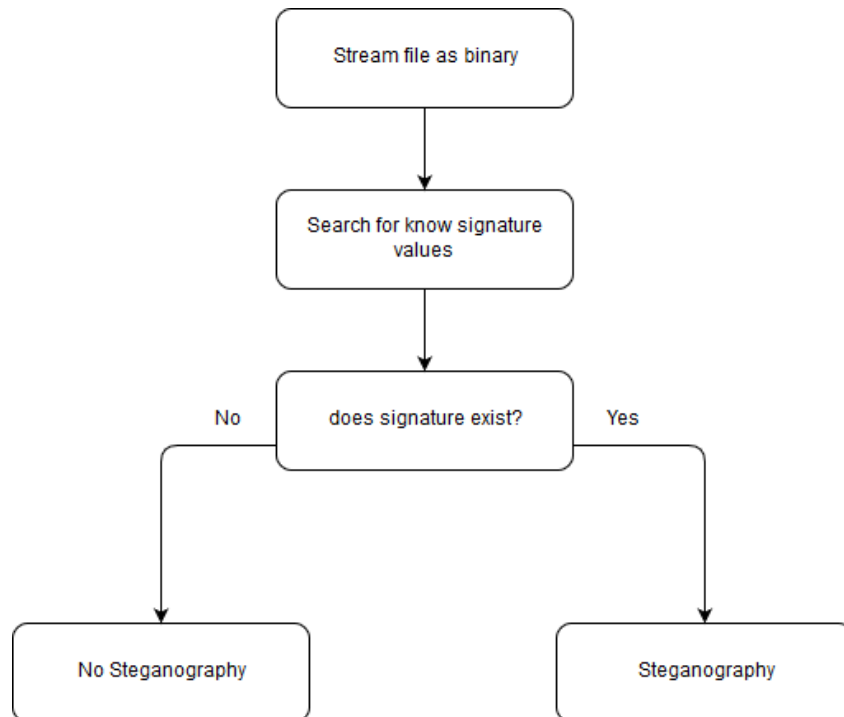


Figure 12: Process for Pixelknot detection

1.8 Forensic Metadata Analysis

Forensic metadata can be extracted using this tool. The RAMSES steganalytic framework makes use of ExifTool to extract useful metadata from any given file (PNG, JPEG, MP4). Any useful information that can be taken from a file is compiled into a report alongside a SHA3-512 hash.

The options for this are user configurable, it is simple for the user to request additional metadata outputs from each file, or limit the information given to the user. A sample of this data is shown in Figure 13.

```
ExifTool Version Number      : 10.10
File Name                    : MP4 (1).MP4
Directory                    : .
File Size                    : 96 MB
File Modification Date/Time   : 2015:07:17 06:45:54+01:00
File Access Date/Time        : 2018:02:25 23:12:54+00:00
File Inode Change Date/Time   : 2017:10:04 15:16:10+01:00
File Permissions              : rwxrwxrwx
File Type                    : MP4
File Type Extension          : mp4
MIME Type                    : video/mp4
Major Brand                   : MP4 Base w/ AVC ext [ISO 14496-12:2005]
Minor Version                 : 0.0.0
Compatible Brands             : avc1, isom
Movie Header Version          : 0
Create Date                   : 2015:07:17 06:45:23
Modify Date                   : 2015:07:17 06:45:23
Time Scale                    : 50000
Duration                      : 0:00:31
Preferred Rate                 : 1
Preferred Volume               : 100.00%
Preview Time                  : 0 s
Preview Duration              : 0 s
Poster Time                   : 0 s
Selection Time                 : 0 s
Selection Duration            : 0 s
Current Time                   : 0 s
Next Track ID                 : 4
Track Header Version          : 0
Track Create Date             : 2015:07:17 06:45:23
Track Modify Date             : 2015:07:17 06:45:23
Track ID                      : 1
Track Duration                : 0:00:31
Track Layer                   : 0
Track Volume                   : 0.00%
```

Figure 13: Sample Exiftool data