Investigating Steganography in Audio Stream for Network Forensic Investigations: Detection & Extraction

Yao Lu

a thesis submitted to the graduate faculty of design and creative technologies
AUT University
In partial fulfillment of the
Requirements for the degree of
Master of Forensic Information Technology

School of Computer and Mathematical Sciences

Auckland, New Zealand
2014

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a University or other institution of higher learning, except where due acknowledgement is made in the acknowledgements.

…………………………….

Yao Lu

Acknowledgements

This thesis was completed at the Faculty of Design and Creative Technologies in the school of Computing and Mathematical Sciences at Auckland University of Technology, New Zealand. Support was received from many people through the 2 years of study. Firstly, I would like to thank my family; my mother Xiangyan Xiao and my father Jianhua Lu who provided financial support. Also special thanks to my fiancée Natalie Ai who took care of my daily life so that I can focus on the thesis research.

I would also like to thank my thesis supervisor, Prof. Brian O. Cusack who has provided valuable support and inspiration in the thesis project. With his support the project progressed smoothly. There are many other staff members at AUT that deserve thanking for their support. Jung Son for setting up EnCase for the research and Alastair Nisbet for managing the laboratory environment for the research testing. Other classmates supported the research with their knowledge and encouragement. Ting and Wei Li were always providing ideas in regards to the research project.

Last but not least, thanks to the members of various open source communities. Specifically, thanks to Backbone Security for providing free trial version of StegAlyzerAS and StegAlyzerSS. Thanks to Chad Davis from Backbone Security who has answered my queries regarding to technical issues and thanks to the proof reader.

Abstract

There are more than 200 steganography tools have been developed by software developers for use in digital media. Multimedia technologies are readily available in the steganography field. They have complex encoding algorithms and compression methods to hide information. Audio is a common multimedia format that is widely used in the Internet. The uploading, downloading and transmission of audio files through the Internet is done through many different audio streams. Audio steganography activities could happen in any of those audio streams. The possibility puts challenges to traditional digital forensic investigation. In order to investigate audio streaming steganography, several steganography detection tools and network stream capture tools need to be involved and evaluated. The current academic literature has little support in this topic area for forensic investigation knowledge. Therefore, the main focus of the research project is to investigate steganography in audio streams as a forensic investigation.

The research asked the question; What are the procedures and challenges when conducting digital forensic investigations for audio steganography?

Five testing phases were designed. In phase 1, three audio steganography tools and two steganography detection tools were tested in order to determine suitable tools for conducting case scenario testing. Openpuff and StegAlyzerAS were found to be the best audio steganography embedding tool and detection tool respectively. WireShark was tested to evaluate audio streaming capture capability as well as packets analysis capability. Phase two involved conducting the case scenario simulating a criminal activity using audio steganography in which Openpuff was used to create four audio steganography files containing evidence. Phase 3 and phase 4 then used combinations of digital forensic tools and steganalysis tools to detect and extract the secret contents from audio streams using standard forensic procedures.

The research findings showed that Openpuff was better than Mp3Stegz and S-Tools both in audio steganography embedding processes and audio steganography extraction processes. StegAlyzerAS was capable of detecting audio steganography tools after scanning while StegAlyzerSS was incapable of detection audio steganography contents during scanning. Additionally, WireShark successfully captured audio streaming packets. The analysis on these captured packets indicated that WireShark would not identify any steganography activities but the analysis could show information such as the type of files in the packet. This information was then used to reduce the scope of the forensic investigation and to better target audio steganography on a suspect's hard drive. After comparative analysis on original evidence and extracted evidence, the extraction rate of audio steganography achieved 75% in the case simulation.

The research has provided knowledge on audio steganography investigation methods and indicates that current forensic tools could cause problems for investigation unless correctly applied. Current steganalysis tools are designed for looking at particular algorithms in steganography but there are other algorithms that are used in audio steganography. This is the challenge for forensic investigators. Therefore, a possible design of an audio steganography tool within a logical flow chart diagram was proposed for future research (see Figure 6.1).

Table of Contents

Chapter 1: INTRODUCTION

Chapter 2: LITERATURE REVIEW

Chapter 3: RESEARCH METHODOLOGY

Chapter 4: RESEARCH FINDINGS

Chapter 5: RESEARCH DISCUSSION

Chapter 6: CONCLUSION

List of Tables

List of Figures

List of Abbreviations

3DES	Triple Data Encryption Standard
3GP	Third Generation Partnership Project
AAC	Advanced Audio Coding
AIFF	Audio Interchange File Format
AIU	Audio Identification Unit
AQMs	Audio Quality Metrics
ASCII	American Standard Code for Information Interchange
ATD	Automatic Target Detection
AUT	Auckland University of Technology
AVIS	Adaptive VoIP Steganography
BFD	Binary File Descriptor
BMP	Bitmap
BRR	Bit Recovery Rate
CPU	Centre Processing Unit
CRR	Character Recovery Rate
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
FLV	Flash Video
FTK	Forensic Toolkit
GA	Genetic Algorithm
GB	Gigabytes
GIF	Graphics Interchange Format
GIF	Graphics Interchange Format
GUI	Graphic User Interface
IBM	International Business Machines Corporation

ICPO	International Criminal Police Organization
IDS	Intrusion Detection System
IM	Instant Messenger
IP	Internet Protocol
JPEG	Joint Photographic Experts Group
KB	Kilobytes
LSB	Least Significant Bits
MB	Megabyte
MD5	Message Digest
MFIT	Master of Forensic Information Technology
MOS	Mean Opinion Score
MP3	MPEG Audio Layer III
MPEG	Moving Picture Experts Group
MS	Micro Soft
NAT	Network Address Translation
OLE2	Object Linking and Embedding 2
P2P	peer-to-peer
PC	Personal Computer
PCX	Personal Computer Exchange Image File Format
PDF	Portable Document Format
PNG	Portable Network Graphics
RAM	Random Access Memory
SAAS	Steganography Analyzer Artefacts Scanner
SARC	Steganography Analysis and Research Centre
SARTS	Steganography Analyzer Real Time Scanner
SASS	Steganography Analyzer Signature Scanner
SHA1	Secure Hash Algorithm 1
SST	Spread Spectrum Techniques

Stego	Steganography
SWF	Small Web Format
TCP	Transmission Control Protocol
TDT	Transform Domain Techniques
TGA	True vision Graphics Adapter
UDP	User Datagram Protocol
VADDI	Voice Activity Detection Dynamic Insertion
VAMI	Value-based Multiple Insertion
VOB	Voice Over Broadband
VoIP	Voice Over Internet Protocol
WAV	Waveform Audio File Format

Chapter 1

INTRODUCTION

1.0 BACKGROUND

Steganography is another term for convert communication. It works by hiding messages in inconspicuous objects (cover objects) that are then sent to the intended recipient. The most important requirement of any steganography system is that it should be impossible for an eavesdropper to distinguish between ordinary objects and objects that contain secret data (Fridrich, 2010). The technology applications for steganography have expanded rapidly in recent years. “The number of steganography software has reached more than 200 at present” (Chen et al., 2006). The number and complexity of application for digital use has kept pace with the growth of the Internet and people requiring tools for multimedia use, including digital audio technologies. Steganography for audio has grown as a new technology that involves different algorithms such as echo encoding and phase encoding that are different from the algorithms used for image steganography.

As audio techniques have been developed for audio streaming on the Internet for radio stations for example but then incorporated into social networking and communication applications such as Skype. Online gaming is also big user of audio channels on the Internet. Using the TCP/IP protocol, audio files can be uploaded, downloaded, and transmitted through the Internet. This benefit of transmission makes “the interest in using audio data as cover object in steganography” become much stronger (Nugraha, 2011). There are many types of audio formats differentiated by encoding algorithms, proprietary design, compression algorithms, and standardized formats. Each type of audio format makes data embedding and steganalysis with different requirements. The different formats raise challenges when investigating and analyzing steganography in forensic investigations (Poisel & Tjoa, 2011).

The types of crimes perpetrated using audio channels on the Internet range from fraud and computer security breaches to distribution of illegal content. One essential piece of electronic evidence in multimedia environments is audio evidence. The main challenges arising when analyzing this kind of data are the limited amount of time to investigate content, the huge amount of data and different multimedia codes and formats (Poisel & Tjoa, 2011). The use of audio steganography as an anti-forensic method raises the challenges of detection and analysis. A research reports shows that the probability of detection for LSB audio steganography is 20% while for AVIS audio steganography is only 8% (Xu, et al., 2011). Hence successful extraction rates for the audio steganographic content are lower than the detection rate during forensic investigation. Thus, if criminals are technology minded and apply audio steganography to protect content, it is hard for forensic investigators to extract the evidence with support from current digital tools. The scope of audio steganography investigation is outlined in Figure 1.1.

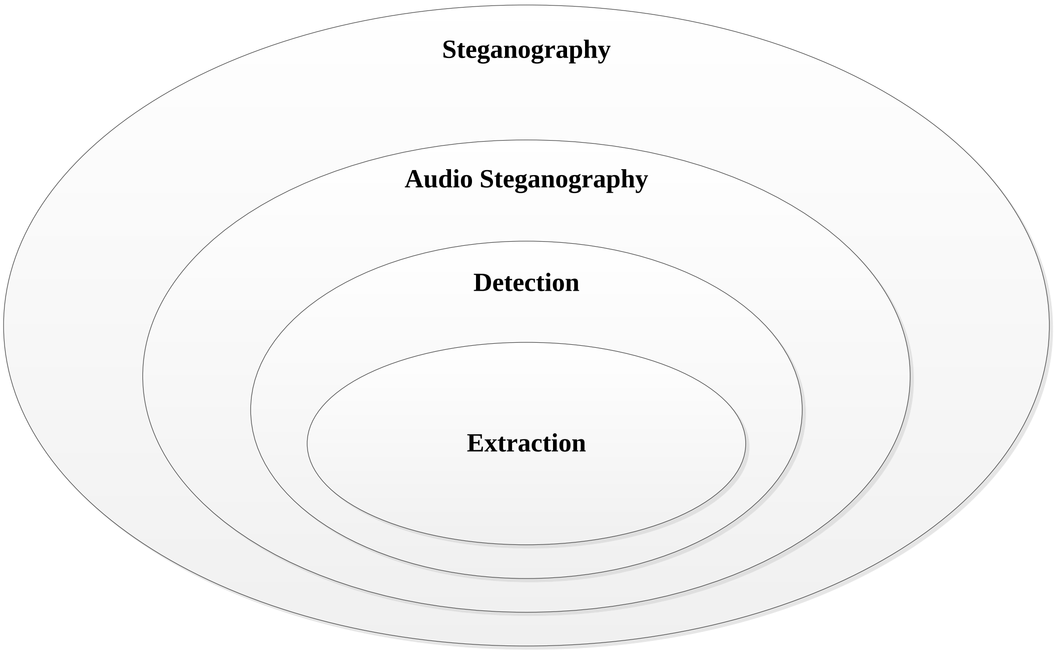


Figure 1.1: Scope of audio steganography challenges

Accordingly, the main research questions in regards to the topic are:

What are the procedures and challenges when conducting digital forensic investigation for audio steganography?

How credible is the extracted content from audio steganography in relation to a forensic evidence purpose?

1.1 MOTIVATION

Section 1.0 briefly introduced the background to the chosen research area of forensic investigation on audio steganography. In order to understand the reasoning for the chosen research area, the motivations of conducting this research will be presented and discussed. The driving motivations include the growth of audio steganography technologies, the increasing of criminal activities in the research area, and the fact that audio steganography is hard to detect.

The concept of steganography dates back many millennia when messages used to be hidden on things of everyday use such as watermarks on letters, carvings on bottom sides of tables, and other objects. The more recent use of this concept emerged with the new digital world. Experiments have shown that data can be hidden in many ways inside different types of digital files. The main benefit of steganography is that the payload is not expected by the investigators who get to examine the computer data (Abboud et al., 2010). More than 200 steganography tools can be used from the Internet and this number increases rapidly each year. Furthermore, many researchers have proposed their own steganography algorithms in order to make it undetectable for forensic investigation.

A group of researchers proposed an AVIS approach for hiding information within network audio streams to drop the detection rate from 20% to 8% (Xu, et al., 2011). Another group of researchers proposed a novel phase coding technique for steganography in audio media that drops the quality of audio file without notice by human ears (Nutzinger & Wurzer, 2011). Moreover, an audio steganography method called controlling bitrate was reported by a group of researchers that no perceived sound quality degradation during data embedding.

All those novel steganography techniques on audio channels raise challenges in forensic investigation.
On the other hand, differentiating anomalous audio document (Stego audio) from pure audio document (cover audio) is difficult and tedious (Geetha et al., 2006). Currently available automated steganography tools have limited performance. From a web search, only three automated steganography tools can be found which are StegAlyzerAS, StegAlyzerSS and StegoSuite. Most approaches are looking for solutions in one particular audio steganography algorithm. Also, there are few publications reporting the evaluation of steganographic content extraction.

Additionally, criminals are getting familiar with steganography concepts and tools use for covert communication purposes. After the events of September 11, 2001, there was immediate concern regarding the possible use of steganography by the al Queda network running up to the Terror Attacks. There were concerns voiced regarding the television broadcasts by the network and potential covert communications. Many organizations are using steganography technology to hide information communications. Watermarking is a visible use to protect properties but steganography is an invisible use. The determining issue is the legitimacy of the property ownerships and the legitimacy of the intended communications (Judge, 2001).

To conclude, there are challenges in chosen research area and there are further challenges to forensic investigators in audio steganography as well. Therefore, it is critical for a researcher to research audio steganography and to find out the current states of tools and techniques for audio steganography investigation. It is also important for forensic investigators to be well prepared for the procedures before conducting audio steganography investigation. Consequently, one of the outcomes of this research is a flow diagram. It brings together the experience and learning gained during the literature review and the laboratory tests and is presented as a suggested audio steganography investigation flowchart (see figure 1.2). Another outcome is the flow diagram for developing a better audio steganography detection tool (see figure 6.1).

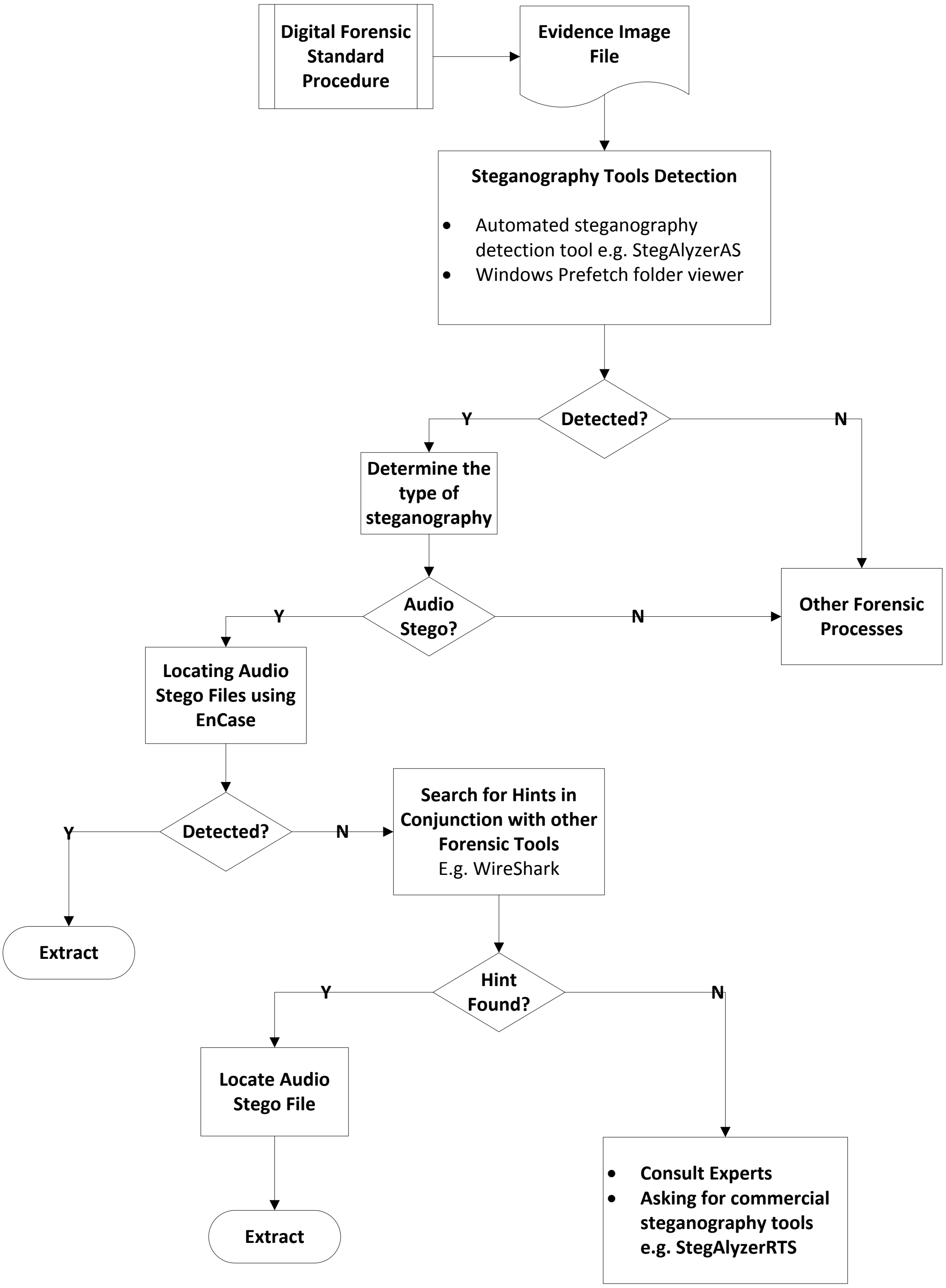

Digital Forensic Standard Procedure
Evidence Image File
Steganography Tools Detection
Automated steganography detection tool e.g. StegAlyzerAS
Windows Prefetch folder viewer
Detected?
Y
N
Determine the type of steganography
Audio Stego?
Y
N
Other Forensic Processes
Locating Audio Stego Files using EnCase
Detected?
Y
N
Search for Hints in Conjunction with other Forensic Tools E.g. WireShark
Extract
Hint Found?
Y
N
Locate Audio Stego File
Extract
Consult Experts
Asking for commercial steganography tools e.g. StegAlyzerRTS

Figure 1.2 and 5.1: Suggested audio steganography investigation flow chart diagram

1.2 STUCTURE OF THE THESIS

The thesis is laid out in six chapters. It is preceded by the formalities and is followed by the references and the Appendix.

Chapter 1 introduces the background for the steganography concept and audio streaming as well as audio steganography detection. The scope of audio steganography detection and extraction is also presented. The motivations for the research project identify gaps and the need for research in the chosen area.

Chapter 2 provides an extensive review and discussion of the current literature within the topic area in order to build theoretical knowledge for the research project. A summary of current issues are discussed as well. Chapter 3 then designs a research methodology to conduct experimental testing for the research project. Five similar studies are reviewed and evaluated in order to support building a research methodology. The identification of data requirements and limitations of the proposed research methodology are also defined and discussed.

Chapter 4 presents the findings from experiments on each testing phase. Several variations made during the actual testing are outlined in order to announce the changes between design and the actual experiment. The chapter then presents the findings in the order from data collection, data treatment, and data analysis. The results from different testing phases will fill in these three data requirement categories. Phase 1 present findings on 3 audio steganography tools and 2 stego detection tools as well as network stream capture tools – WireShark. Phase 2 combined with phase 3 and phase 4 generates findings from a simulated case scenario. The findings of comparative analysis are also presented. All the findings are in table forms or screen shot figures in the Appendix.

Chapter 5 presents the research findings in comparison with the reviewed literature in Chapter 2 and the similar studies in Chapter 3. The main research questions and sub research questions are answered in table form. Asserted hypotheses are tested using the findings from the previous chapter. The discussions reflect on the testing phases, and the reported findings are reconciled with reviewed literature. A brief recommendation on audio steganography investigation as well as a guideline in a flow chart diagram is included.

Chapter 6 is the conclusion section that summarizes the entire research project. The findings from testing phases have been reviewed. The limitations of the research are proposed as well as possible future research ideas in the chosen or related research areas. References and the appendix are included at the end of the thesis. The research appendix contains the findings and collected data in screen shot figures.

Chapter 2

LITERATURE REVIEW

2.0 INTRODUCTION

The research objective of chapter 2 is to critically review the current literature which are relevant to the study areas of this thesis; namely steganography in digital media, audio streams and digital forensic principles. Firstly, it is vital to understand all the aspects of steganography in digital world, the history, current development, security (secure stego-systems and steganography increases security), steganalysis, potential threats and attacks, and the growing concern of involving steganography in audio streams. This then will link to digital forensic investigations for steganography in audio streams, and the need for investigating the present knowledge for the acquisition, preservation, analysis and reporting of audio stream steganography evidence.

This chapter is not only to find present-day facts of the target technology, but also will review the prospective problems and issues exist for feasibility analysis and adoption for research. Chapter 2 is structured into the following sections. In Section 2.1, the definition of steganography will be introduced. Section 2.2 will briefly outline the history through the developing of steganography techniques and tools. Then, the following Section 2.3 will review the main steganography technologies. Section 2.4 will review the current issues in steganography followed by a review of the tools of steganography in Section 2.5. Section 2.6 will introduce and define audio steganography. Section 2.7 will review the current developments in steganography detection and steganalysis areas. Section 2.8 will review the forensic investigations processes relevant to steganography and audio streams. Then an evaluation of current issues and problems in the proposed research area will be presented in Section 2.9.

2.1 STEGANOGRAPHY DEFINITION

Steganography is another term for convert communication. It works by hiding messages in inconspicuous (cover) objects that are then sent to the intended recipient. The most important requirement of any steganography system is that it should be impossible for an eavesdropper to distinguish between ordinary objects and objects that contain secret data (Fridrich, 2010). For example watermarks on letters is a typical use of steganography, but the most recent use of this concept emerged with digital media such as mp3 files, mp4 files, JPEG files, and so on. The following section will review the history of steganography, the technology background, specific references to steganography, steganography channels and tools.

2.2 STEGANOGRAPHY THROUGHOUT HISTORY

The term steganography was first used by Johannes Trithemius (1462-1516) in his trilogy polygraphia and in steganographia. Scholars then suspected that the book was a code and attempted to decipher the mystery which was not solved until 1996 (Fridrich, 2010). The first written evidence about steganography being used to send messages is due to Herodotus. In his story, he tells of a slave sent by his master to the Ionian city of Miletus with a secret message tattooed on his scalp. When the slave grew his hair, the message was concealed and could be delivered safely (Fridrich, 2010).

It seems early steganography methods were based on hiding messages in text which is also called linguistic steganography or acrostics. A famous linguistic steganography scheme is called Cardan’s Grille, which was originally conceived in China and reinvented by Cardan (1501-1576). It simply places a mask on the text then the letters of the secret message can be accessed.

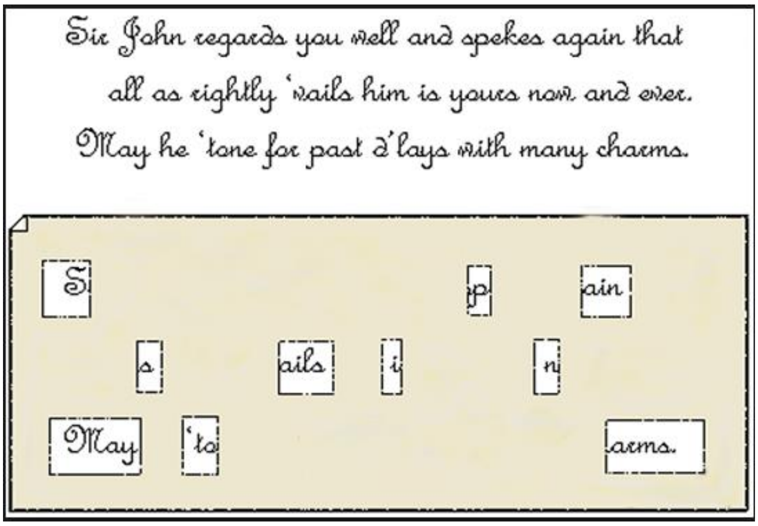

Sir John regards you well and spekes again that
all as rightly 'vails him is yours now and ever.
May he 'tone for past d'lays with many charms.

Figure 2.1: The Cardan's Grille (Fridrich, 2010)

An enhanced method from Cardan's Grille which called Turning Grille was used during World War I. The Turning Grille is looked like a normal grille. But when using the Turning Grille, the encoder wrote the first sequence of letters, then rotate the grille 90 degrees and wrote the second sequence of letters, and so on, rotating the grille after each sequence. Therefore the secret message had been created (Kipper, 2004). Another case of steganography during World War I was writing secret message using invisible ink. Spies used milk, vinegar and fruit juice to write messages on a blank paper and it cannot be seen by human eyes. A Receiver can see the message by simply heating the paper (Jamil, 1999).

During World War II, steganography techniques were developed and applied. Nazis created a method called microdot. A microdot was a tiny photograph the size of a typed period. It could reproduce to a standard sized typewritten page with marvelous clarity. Extremely difficult to detect, therefore the Germans used microdots to transmit large amounts of printed data during the war (Jamil, 1999).

In 1980s, invisible watermarking had been used by former British Prime Minister, Margaret Thatcher. After several cabinet documents had been leaked, Thatcher asked all document processors encode their identity in the spacing of the document. This allowed for disloyal ministers to be quickly found out (Kipper, 2004).

Currently, the emphasis has been on various forms of digital steganography. Commonly there are a number of digital technologies that the community is concerned with, namely text files, still images, movie images, and audio. After the events of September 11, 2001, there was immediate concern regarding the possible use of steganography (Judge, 2001). But, there is no conclusive evidence that shows terrorist organization hiding message using steganography technology. However, there are organizations that have been found putting steganography materials on websites. A web site posting belonging to the Ulster Loyalist Volunteer Force was found had hidden messages found. The majority of organizations using steganography technologies are protecting intellectual property. Other organizations that are using steganography technology are involved with digital watermarks for the protection of properties. Using a variety of techniques, images, music, and movies can be imprinted with digital watermarks (Judge, 2001). Xerox has developed a digital watermarking system for printed images which involves generating half tone of screen with a key image. When the screen is superimposed over the original image, with a slight offset, the key image will appear.

Steganography is developing rapidly as a security technology. The next section will discuss the steganography technology is detail.

2.3 STEGANOGRAPHY TECHNOLOGY

Generally, in digital steganography, there are only three principles in hiding a digital message in a digital cover: injection, substitution, and generation of new files (Kipper, 2004). Injection means directly embed secret message into host medium. The problem is that it often makes the host file larger, and then makes it easier to detect. Substitution means replace normal data with secret data. This will keep the size of host file but may drop the quality of the original host file. Generation of new files, as it says, is generating a cover for the sole purpose of hiding a secret message. According to Kipper (Kipper, 2004), the three principles can be split into six categories.

2.3.1 Categories of Steganography

The six categories of steganography are:

1. Substitution System techniques
2. Transform domain techniques
3. Spread spectrum techniques
4. Statistical method techniques
5. Distortion techniques
6. Cover generation techniques (Kipper, 2004).

2.3.1.1 Substitution System Techniques

Substitution system steganography replaces redundant or unneeded bits of a cover with the bits from the secret message (Kipper, 2004). A typical substitution system technique is named Least-Significant Bit (LSB) method. Several steganography tools use LSB method to encode the secret message in digital cover (picture, audio, or video file) because there is an enormous amount of redundant or unneeded space. It is this space that the steganography program will take advantage of and use it to hide a secret message on the bit level within the digital cover (Kipper, 2004).

LSB method works in the following way. For example, the following string of bytes represents part of a cover (a picture file).

1000100 10000110 100001001 10001101

01111001 01100101 01001010 00100110

These bits make up a color value in the picture file. If we change the first bit in the first string from 1 (1000100) to 0 (0000100) will dramatically change the color of the picture and can be easily detected. As opposed changing the last bit from 0 (1000100) to 1 (1000101) will have little effect on the information the byte representing. We call the last bit the least-significant bit. Now, assume we want to hide a secret number 214 into the cover. In binary number 214 are 1101010. Using the least-significant bit method, the 214 message will be blended into cover as shown in table 2.1.

Table 2.1: The substitution system of LSB method

Original bits in cover	Secret message (214)	Steganography message
10000100	1	10000101
10000110	1	10000111
10001001	0	10001000
10001101 01111001	1 0	10001101 01111001
01100101	1	01100101
01001010	1	01001010
00100110	0	00100110

From the above table, we can see that after embedding the secret message, there are only 5 strings of bytes have been changed and in a picture file there are 500 kilobytes to 1 megabyte of redundant information. In other words, there are 1 megabyte of least-significant bits can be changed with little noticeable difference to the cover picture. Therefore, LSB method is commonly used in steganography as the method is quick and easy to use. The LSB method has some drawbacks as well. According to Sathyal et al., (Sathyal et al., 2012) using the LSB method only a few messages can be embedded in the cover image when the change affects the pixel's properties. The LSB method can also cause distortion detectable by steganalysis methods. If the image is cropped or rotated during transmission, the method will not be able to find which least-significant bits is the secret message.

2.3.1.2 Transform Domain Techniques

Transform Domain techniques are very effective especially dealing with image steganography. Basically, transform domain techniques hide secret message data in the transform space of a signal (Kipper, 2004). So what does it mean? Let's first look at how we send images in real life. Normally, we are using JPEG format when sending images through internet. The Joint Photographic Experts Group

(JPEG) adopted the discrete cosine transform (DCT) as its power level transform mainly targeting image compression for communication which means JPEGs compress themselves when they close. According to Munirajan et al. (Munirajan et al., 2004) “The DCT based image compression technique is very effective in terms of attaining image compression suitable for transmission and makes such a technique based images highly populated on the web. Also it will not display any significant degradation that could be visually noticed when used for steganography”. During the DCT based image compression, JEPGs will reduce its size to be smaller. So the change from compression process is called transform space and this transform space is where we can hide secret messages in data. Therefore, the transform domain techniques are usually used in JEPG image steganography.

2.3.1.3 Spread Spectrum Techniques

Direct Sequence

In direct sequence spread spectrum, the stream of information to be transmitted is divided into small pieces. Each of the pieces is allocated to a frequency channel of the spectrum (Kipper, 2004). During the data transmission, the signal is combined with higher data bit sequence that divides the data related to a predetermined spread ratio. If any data bits are damaged during the transmission the redundant data rate bit sequence code will enables the original data to be recovered.

Frequency Hopping

In frequency hopping spread spectrum, the bandwidth will be divided into many possible broadcast frequencies. The signal will switch carriers quickly among these frequencies follow a pseudorandom sequence. Compare to frequency hopping, direct sequence spread spectrum is usually better and more reliable.

2.3.1.4 Statistical Methods Techniques

In statistical methods, only 1-bit of information embeds in a digital carrier to create a statistical change. So it is also called 1-bit steganography scheme. A statistical change in the cover represent a “1”, an unchanged cover represent a “0”.

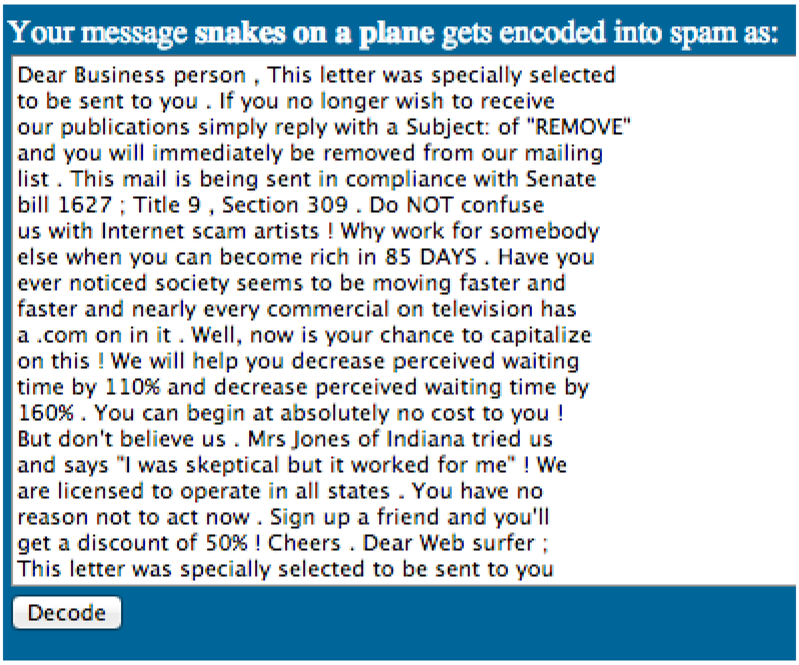
This steganography technique works based on the ability of a receiver to distinguish between modified and unmodified covers. Those 1s and 0s can be recovered into the secret messages.

2.3.1.5 Distortion Techniques

Normally, in steganography, a good practice is hiding the secret message in the cover without any detection being noticed by an observer. To ensure this it is better to keep a modified file same as the original cover as far as possible. However, a change in the cover file can hide information as well (Kipper, 2004). In this method, receiver calculates the hidden secret message by comparing the distorted cover with the original file.

2.3.1.6 Cover Generation Methods Techniques

The cover generation method in steganography is unique and different from other steganography methods. Typically, people choose a cover object to hide secret message in. But using cover generation method, people actually creates a cover for the sole purpose of hiding information (Kipper, 2004). The spam mimic is the best example of a cover generation method.



Your message snakes on a plane gets encoded into spam as:

Dear Business person , This letter was specially selected
to be sent to you . If you no longer wish to receive
our publications simply reply with a Subject: of "REMOVE"
and you will immediately be removed from our mailing
list . This mail is being sent in compliance with Senate
bill 1627 ; Title 9 , Section 309 . Do NOT confuse
us with Internet scam artists ! Why work for somebody
else when you can become rich in 85 DAYS . Have you
ever noticed society seems to be moving faster and
faster and nearly every commercial on television has
a .com on in it . Well, now is your chance to capitalize
on this ! We will help you decrease perceived waiting
time by 110% and decrease perceived waiting time by
160% . You can begin at absolutely no cost to you !
But don't believe us . Mrs Jones of Indiana tried us
and says "I was skeptical but it worked for me" ! We
are licensed to operate in all states . You have no
reason not to act now . Sign up a friend and you'll
get a discount of 50% ! Cheers . Dear Web surfer ;
This letter was specially selected to be sent to you

Figure 2.2: Spam mimic. (Fridrich, 2010)

Figure2 shows a result from spam mimic. The hidden secret message “snakes on a plane” was embedded into the spam email. The observer obviously cannot read the secret message when they look at the spam email so the hidden secret message can be delivered safely to receiver. After decoding, the receiver should read the secret message.

2.3.2 Types of Steganography

From the techniques involved in, steganography can be separated into the above categories. Also, it can be separated into two types – linguistic and technical depends on the different ways and methods to conceal information (Figure 2.3).

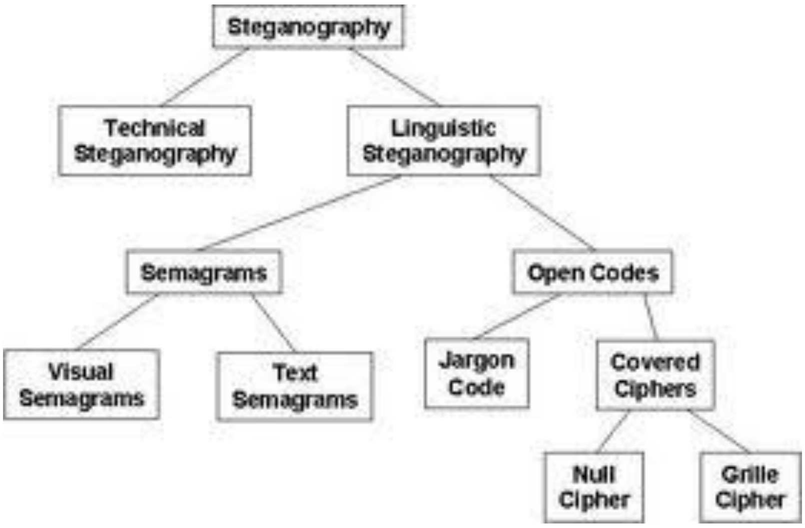


Figure 2.3: Types of steganography. (Kipper, 2004)

2.3.2.1 Linguistic Steganography

As it is said, linguistic steganography is any form of steganography that uses language in the cover (Kipper, 2004). Linguistic steganography is a broad area and also it is the earliest method in steganography. There are a number of forms of linguistic steganography that have been invented.

NICETEXT

A program called NICETEXT uses linguistic steganography in a very inventive way. The goal of NICETEXT is to provide a program that can transform secret text into text that looks like natural language while still providing a cover for the original secret text (Kipper, 2004). Thus the secret text is hidden inside the text cover. The advantage of this program is that it can be applied to many different languages. NICETEXT has large code dictionaries consisting of different types of words. It transforms secret text into natural text by selecting words using the matching codes for the proper types of word in dictionary table. The reverse process simply parses individual words from the text and uses codes from dictionary table to recover the secret text.

Masking

It is well known that encryption provides secure channels for communicating entities. However, due to lack of covertness on these channels, an eavesdropper can identify encrypted streams through statistical tests and capture them for further cryptanalysis. Hence, the communicating entities can use steganography to achieve covertness. Radhakrishnan et al. (2005) proposed a new form of multimedia steganography called data masking. Instead of embedding a secret message into a multimedia object, as in traditional multimedia steganography, they processed the entire secret message to make it appear statistically similar to a multimedia object itself. Thereby they foiled an eavesdropper who is primarily applying statistical tests to detect encrypted communication channels. It is showed that their approach can potentially give a covert channel capacity which is an order of magnitude higher than traditional steganography. Their experiments also showed that steganalyzer trained with stego objects of known steganographic have low detection accuracy for data masked multimedia objects (Radhakrishnan et al., 2005).

Grilles

A grille is simply a stiff piece of paper or cardboard with holes positioned around it. The secret message is written in the holes, and then the rest of the message is filled in around it. The only way the message is readable is by the recipient who has the correct grille (Kipper, 2004).

Text Semagrams

Text semagrams can be described as graphical modifications of the text. They concern details are tiny but visible. There are methods works without text as well, called real Semagrams. Some examples of text semagrams are type spacing, tiny spaces and offsetting. Type spacing uses the white space in a document to denote binary values. It can be between the individual words, the sentences, or even between the paragraphs (Kipper, 2004). Furthermore, there are not only spaces between words, sentences and paragraphs but also tiny spaces between letters. These tiny spaces either refer to binary code or the letter before/after it is part of the secret message.

Real semagrams is usually an indicator of previously agreed message. For example, Bob wants to tell Alice that everything is prepared for Friday activity. A real semagrams could be Bob sends a postcard with a picture of city center to Alice as agreed previously that only Alice knows the meaning. Therefore the secret message has been delivered safely.

2.3.2.2 Technical Steganography

As defined by Kipper (Kipper, 2004), technical steganography is the method of steganography where a tool, device, or method is used to conceal the message. Some examples of technical steganography can be invisible ink, microdots, and computer-based methods. As these examples were mentioned at the beginning of this chapter I will not go further in this section.

2.4 ISSUES IN STEGANOGRAPHY

Although there are many advantages and modern technologies in steganography, it is always limited by problems. The first issue in steganography turned out to be the level of visibility. It means whether the embedding process distort the cover to the point where it is visually or audibly noticeable or not. If the cover is unacceptably changed then it is not sufficient for the payload. The second issue is robustness and payload. Every things in the world require trade-offs includes information hiding. Kipper (Kipper, 2004) believes that to have a robust method of embedding the message means you must have redundancy to resist changes made to the cover. Obviously, this redundancy lowers the payload. In general the relationship between redundancy (robustness) and the payload is below.

- More robust = lower payload
- Less robust = higher payload

File dependence is another issue in steganography. For example, in digital steganography, some sound files are either lossy or lossless. The conversion of lossless file to the compressed lossy file can destroy the hidden information in the cover. Some compressed MP3 file cut out some bits to increase the ability of transmission. Those bits may be the part of the embedded message.

2.5 STEGANOGRAPHY TOOLS

Steganography in digital use is still a young technology, however, it is also an interesting area that many steganography tools can be found and used from the Internet. The number of steganography software and tools has reached more than 200 at present (Chen et al., 2006). In this section, I will introduce some steganography tools either famous or useful.

2.5.1 BackYard

BackYard (Kipper, 2004) is a steganography file system program that can hide and protect files, folders and drives by making them completely inaccessible from operating system. It also has a built-in search function to find and protect files. Features include:

- Make any file or directory invisible
- Write protect any file or directory
- Make any file or directory attributes unchangeable
- Make any file or directory attributes impossible to delete or read
- Make any directory deny a file or sub-folder creation
- Protect all files with a specific extension or an entire drive

2.5.2 BMP Secrets

BMP Secrets is a powerful steganography program allows user to store any information in a bitmap file. It is useful because of its large hiding capacity. Some features include:

- The program uses an original steganography method developed by Parallel Worlds that allows users to replace up to 65% of the true-color BMP file with data. User can convert the result image only to lossless format; lossy format will destroy the information as I have mentioned in previous section. If users try to make any changes to the result image, information will also be lost.
- Users can choose hiding rate. Notice that the higher the hiding rate, the lower the quality.
- A built-in to encoding compressor that allows the storage of much more text files than binary.
- A User can hide information in any part of the image. Sometimes, a user can hide two files in two areas in one image.
- A User can set an automatic quality option. The program will search for the best quality when the whole file can be hidden.
- Hiding spread data all over the image when user provides a password. To withdraw an encoded file and to decode it is very difficult, because nobody except the user knows the data-spreading order.
- A User can view results of hiding and compare the original with the result. (Kipper, 2004)

2.5.3 Info Stego

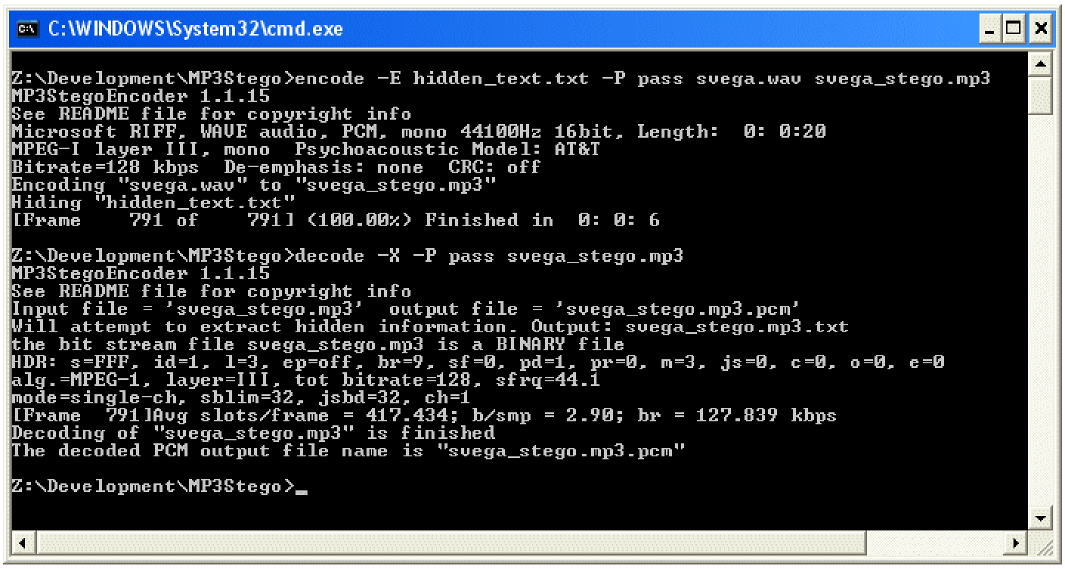
Info Stego is a steganography and watermarking tool that allows user to protect their private information, secret communication, and legal copyright information and data encryption technology. Info Stego can hide information and copyright mark inside another file, which can be picture, sound, video, and so on. The advantage of Info Stego is that people can barely notice the change of cover file with their eyes and ears. (Kipper, 2004)

2.5.4 MP3Stego

When looking at the steganography tools available on the Net, it occurred to Fabien (Petitcolas, 2012) that nothing had been done to hide information in MP3 files which are sound tracks compressed using the MPEG Audio Layer III format. There is a growing interest world-wide in MP3 files because they offer near-CD quality at compression ratio of 11 to 1 (128 kilobits per second). This gives a very good opportunity for information hiding. So Fabien developed a tool called MP3Stego in order to hide information in MP3 file.

MP3Stego will hide information in MP3 files during the compression process. The data is first compressed, encrypted and then hidden in the MP3 bit stream. Although MP3Stego has been written with steganography applications in mind it might be used as a copyright marking system for MP3 files (weak but still much better than the MPEG copyright flag defined by the standard). Any opponent can uncompress the bit stream and recompress it; this will delete the hidden information at the expense of severe quality loss.

In MP3Stego, the hiding process takes place at the heart of the Layer III encoding process namely in the inner loop. The inner loop quantizes the input data and increases the quantiser step size until the quantized data can be coded with the available number of bits. Another loop checks that the distortions introduced by the quantization do not exceed the threshold defined by the psycho acoustic model. The part2_3_length variable contains the number of main data bits used for scale factors and Huffman code data in the MP3 bit stream. It encodes the bits as its parity by changing the end loop condition of the inner loop. Only randomly chosen part2_3_length values are modified; the selection is done using a pseudo random bit generator based on SHA-1 (Petitcolas, 2012). Figure 2.4 shows the command view of using MP3Stego.



```
Z:\Development\MP3Stego>encode -E hidden_text.txt -P pass svega.wav svega_stego.mp3
MP3StegoEncoder 1.1.15
See README file for copyright info
Microsoft RIFF, WAVE audio, PCM, mono 44100Hz 16bit, Length:  0: 0:20
MPEG-I layer III, mono  Psychoacoustic Model: AT&T
Bitrate=128 kbps  De-emphasis: none  CRC: off
Encoding "svega.wav" to "svega_stego.mp3"
Hiding "hidden_text.txt"
[Frame    791 of    791] (100.00%) Finished in  0: 0: 6

Z:\Development\MP3Stego>decode -X -P pass svega_stego.mp3
MP3StegoEncoder 1.1.15
See README file for copyright info
Input file = 'svega_stego.mp3'  output file = 'svega_stego.mp3.pcm'
Will attempt to extract hidden information. Output: svega_stego.mp3.txt
the bit stream file svega_stego.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=3, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=single-ch, sblim=32, jsbd=32, ch=1
[Frame  791]Avg slots/frame = 417.434; b/smp = 2.90; br = 127.839 kbps
Decoding of "svega_stego.mp3" is finished
The decoded PCM output file name is "svega_stego.mp3.pcm"

Z:\Development\MP3Stego>_
```

Figure 2.4: MP3Stego command view. (Petitcolas , 2012)

2.5.5 F5

Many steganography systems are weak against visual and statistical attacks. Systems without these weaknesses offer only a relatively small capacity for steganography messages. However, the newly developed algorithm F5 withstands visual and statistical attacks effectively.

Visual attacks on steganography systems are based on essential information in the cover medium that steganography algorithms overwrite. Lossy compressed carrier media such as JPEG and MP3 are originally adaptive and immune against visual and auditory respective attacks. Steganography tools Jsteg and MP3Stego are withstanding visual and auditory attacks respectively. But these two steganography tools offer only a small capacity for steganography message (less than 1%). Unlike Jsteg and MP3Stego, steganography tool F5 uses a Java runtime environment and new steganography algorithm that allows it to embed files into true-color BMP, GIF, or JPEG images in a high capacity up to 13% of the carrier image size as well as withstand visual and statistic attacks (Westfeld, 2001).

2.6 AUDIO STEGANOGRAPHY

Audio steganography as it is said is a form of technology that uses audio files as cover media to hide secret information for communication purpose. The key function in audio steganography is funding a proper method to embed information either a text file or another audio file into the cover media. The major assessment criteria of embedding methods are the trade-off between payload and visibility. In section 1.6.1 I will introduce four well used embedding methods in audio steganography.

2.6.1 Embedding Methods in Audio Steganography

There are huge challenges for data hiding in technical applications. For example, any embedded data in the host signal are likely to be removed or modify by compression algorithms. The key of successfully embedding data in audio steganography is to find the holes that are not possible for exploitation by compression algorithms. Consequently, the experts in steganography use embedding methods that are commonly found in everyday logic algorithms.

2.6.1.1 Least Significant bit Encoding

The simplest embedding method in audio steganography is low bit encoding, commonly known as the least significant bit encoding. In image steganography the LSB of grey value of each cover image pixel is replaced with corresponding message bit to generate the stego image (Sarreshtedari et al., 2009). In LSB audio steganography an appropriate bit is modified at least significant bit .The remaining bits will be used but it may be cause noise (Kumar & Anuradha, 2012). This method can be attacked by several steganalysis methods, because it detectably changes statistical and perceptual characteristics of the cover signal. A typical method for steganalysis of the LSB encoding is the histogram attack that attempts to diagnose anomalies in the cover’s histogram. A well-known method to withstand the histogram attack is the LSB+ steganography that intentionally embeds some extra bits to make the histogram look natural. However, the LSB+

method still affects the perceptual and statistical characteristics of the cover signal. Therefore, Ghazanfari et al. (Ghazanfari et al., 2011) proposed a new method called LSB++, which improved over the LSB+ steganography by decreasing the amount of changes made to the perceptual and statistical attributes of the cover media. They identified some sensitive pixels affecting the signal characteristics, and then lock and keep them from the extra bit embedding process of the LSB+ method, by introducing a new embedding key. Evaluation results showed that, without reducing the embedding capacity, their method can decrease potentially detectable changes caused by the embedding process.

2.6.1.2 Phase Encoding

Phase encoding embedding algorithm is based on the phase encoding technique which embeds data in the phase spectrum of the frequency domain signal of the audio media. Nutzinger & Wurzer (Nutzinger & Wurzer, 2011) enhanced a novel phase encoding algorithm from the previous researchers. Differing from previous works they retained the original phase values in order to best keep the quality of the cover audio signal. Secret bits were embedded by introducing a configurable phase difference between selected chunks of blocks from the cover medium instead of discarding the original phase values and they also introduced a random phase like previous researchers had.

The phase encoding algorithm was first introduced in “Techniques for data hiding” written by Bender et al. (Bender et al., 1996) in IBM Systems Journal. In this system the phase spectrum of the cover audio signal was split into several blocks. The original phase from the first block was discarded and replaced with values representing the bits of the secret message. All other phase values were further modified in such a way that the phase difference was the same for the original as well as the modified phase spectra (Nutzinger & Wurzer, 2011). Unlikely, Nutzinger & Wurzer developed the new phase encoding algorithm beginning with splitting the original audio cover into blocks and transferred every block into the frequency domain by FFT algorithm. The two variables depict the frequency interval was used for embedding and extraction. Figure 2.5 shows the embedding process.

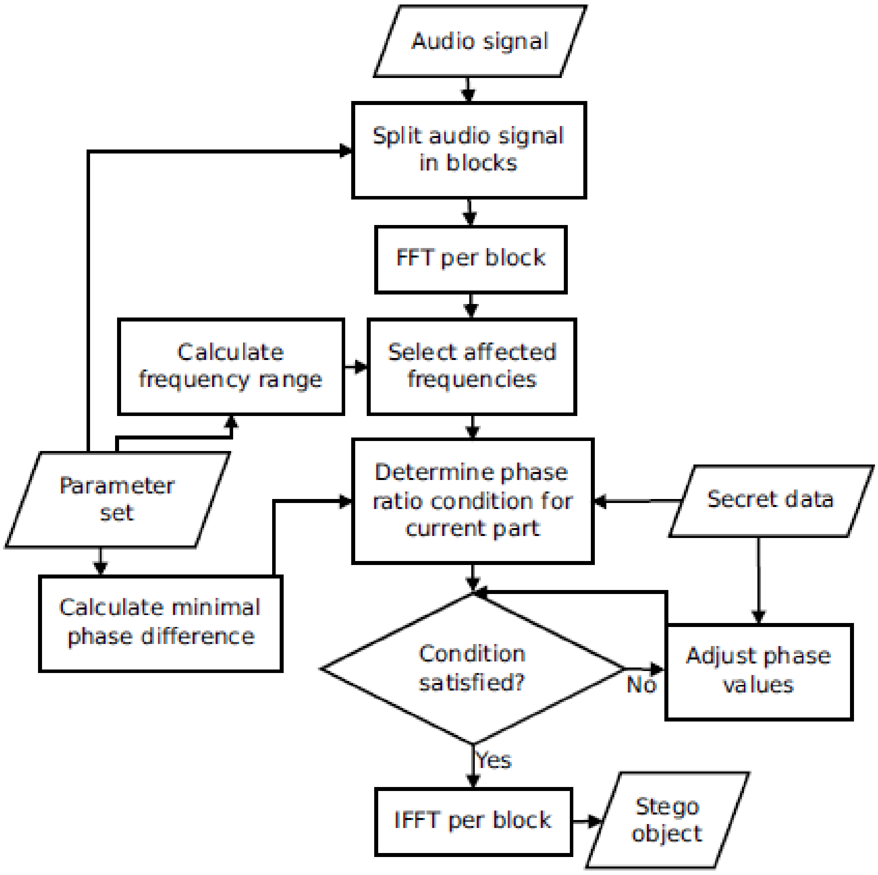

Audio signal
Split audio signal in blocks
FFT per block
Calculate frequency range
Select affected frequencies
Parameter set
Determine phase ratio condition for current part
Secret data
Calculate minimal phase difference
Condition satisfied?
No
Adjust phase values
Yes
IFFT per block
Stego object

Figure 2.5: The phase encoding embedding process. (Nutzinger & Wurzer, 2011, p.93)

The extraction process is a backward play of the embedding process. To evaluation the phase encoding algorithm, Nutzinger & Wurzer (Nutzinger & Wurzer, 2011) evaluated their steganography algorithm with four tests which were security through parameterization tests, robustness tests, listening test, and statistical analysis tests. The result showed that the variable parameters guaranteed the security of their system. The robustness of their algorithm was not related to the audio cover used with the exception of the noise addition. The listening test showed that the original audio quality was not noticeable reduced. Finally, the statistical analysis could exploit a slight modification in the audio media after using their algorithm.

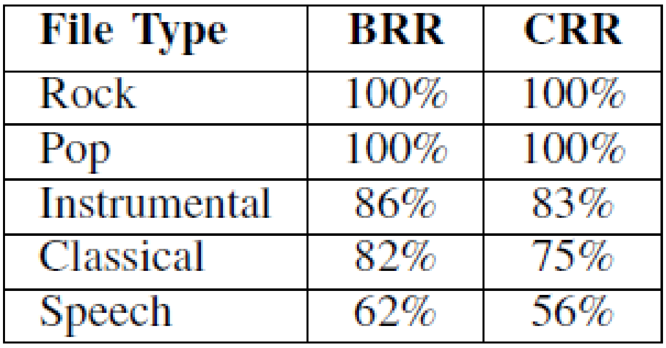
In general, the phase encoding embedding method creates a new field in audio steganography. The outcome of this method makes audio steganography more practical and less complicated.

2.6.1.3 Echo Encoding

Echo encoding is a steganography embedding technology to hide information in audio media. Unlike least significant bit encoding that could give 100% recovery; echo encoding algorithms usually had a lower recovery rate. But Mitra & Manoharan (Mitra & Manoharan, 2009) introduced a new echo encoding algorithm in 2009 that can significantly increase the recovery rate.

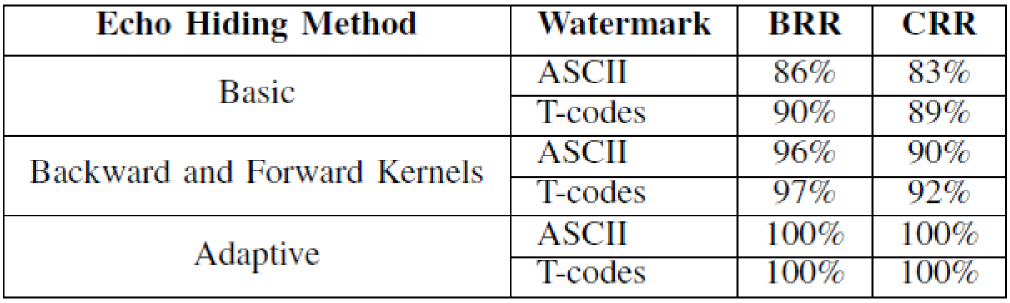
In basic echo encoding, the cover audio was divided into small segments and echoes were introduced to each sample of segments. The echo delay rate and echo decay rate were used to control the hiding message. After information embedding, all of the encoded segments were re-combined and written to an output audio. But the recovery rate of this basic echo encoding algorithm is low. Then Mitra & Manoharan (Mitra & Manoharan, 2009) introduced T-code in their echo encoding algorithm. T-code is a subset of all Huffman code sets. The advantage of using T-code in echo encoding is when some bits were lost or corrupt in a T-code encoded stream during decoding, the decoder regains synchronization automatically. So it is more tolerant to media transformation that results in bit loss, inversion, or bit additions.

During their performance evaluation, the measurements of recovery rate were divided into Bit Recovery Rate (BRR) and Character Recovery Rate (CRR). The result was shown in Figure 2.6, Figure 2.7, and Figure 2.8.



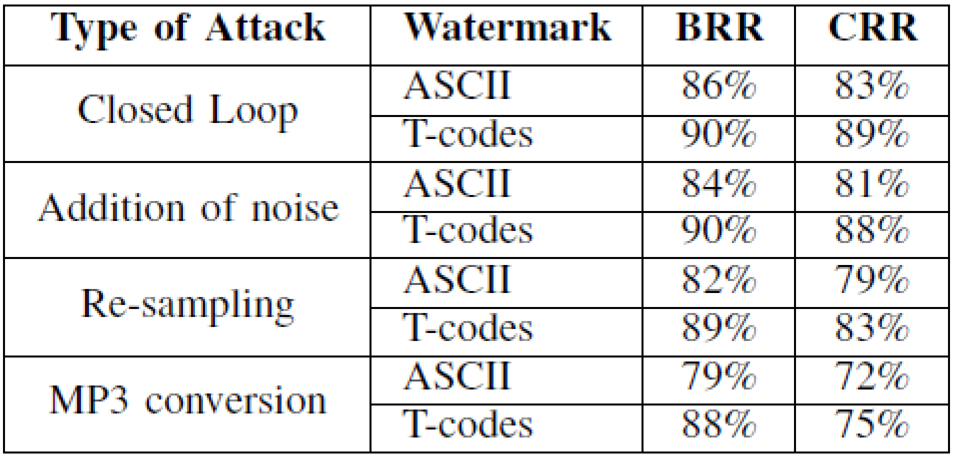
File Type	BRR	CRR
Rock	100%	100%
Pop	100%	100%
Instrumental	86%	83%
Classical	82%	75%
Speech	62%	56%

Figure 2.6: Effect of file type on recovery rate. (Mitra & Manoharan, 2009, p.122)



Echo Hiding Method	Watermark	BRR	CRR
Basic	ASCII	86%	83%
	T-codes	90%	89%
Backward and Forward Kernels	ASCII	96%	90%
	T-codes	97%	92%
Adaptive	ASCII	100%	100%
	T-codes	100%	100%

Figure 2.7: Effect of echo encoding method and T-Codes on recovery rate. (Mitra & Manoharan, 2009, p.124)



Type of Attack	Watermark	BRR	CRR
Closed Loop	ASCII	86%	83%
	T-codes	90%	89%
Addition of noise	ASCII	84%	81%
	T-codes	90%	88%
Re-sampling	ASCII	82%	79%
	T-codes	89%	83%
MP3 conversion	ASCII	79%	72%
	T-codes	88%	75%

Figure 2.8: Effect of type of attack and T-Codes on recovery rate. (Mitra & Manoharan, 2009, p.124)

The results showed a significant improvement of using their echo encoding algorithm relates to recovery rate. Also, their algorithm increased audio steganography capacity. For instance, the number of bits required to hide the text “helloworld!” reduces from 77 to 66 when using T-codes with a fixed dictionary for ASCII character set.

2.6.1.4 Bitrate Control on AAC

Wei et al. (Wei et al., 2010) has proposed a steganography method on AAC (Advanced Audio Coding) file which the audio format was MPEG-2 and MPEG-4. The hidden data were first encrypted and then embedded into the AAC bit stream. The data embedding took places at the core of the AAC encoder, and each bit of the hidden data was encoded as the parity of the number of bits used for Huffman and differential coding in a frame of the AAC bit stream.

AAC is a lossy compression encoding scheme for digital audio which gains better sound quality than MP3 at the same bit-rate. It has been standardized by ISO as MPEG-2 part 7 and MPEG-4 part 4. As the successor of MP3, AAC has the similar encoding process compare to MP3. In order to learn steganography method on AAC I need to understand how raw audio are compressed into an AAC format frames. Firstly, 1024 samples are read from the raw data file, and then these samples are processed with analysis filterbank. Later, a Modified Discrete Cosine Transform is performed on the sub-band values obtained in the last step. On the other hand, a psychoacoustic model is employed to calculate the mask threshold for each sub-band. In the quantization process, two nested iteration loops are utilized. The inner loop ensures the audio samples could be quantized within the pre-calculated number of bits; while the outer loop ensures the distortions resulted from the quantization do not exceed the mask threshold defined by the psychoacoustics model. Finally, the encoded bits are stored in the bit stream. Figure.2.9 shows the block diagram of an AAC encoder.

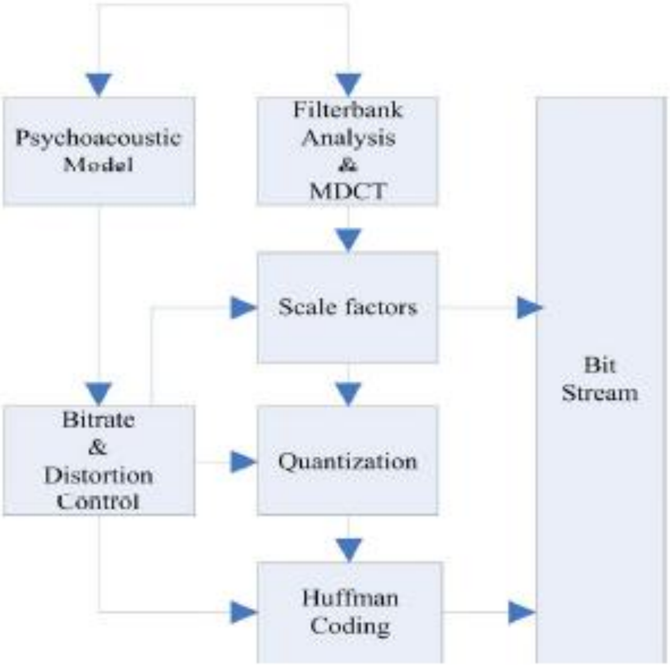


Figure 2.9: An AAC Encoder. (Wei et al., 2010, p.4374)

Author embedded the hidden data in the process of compressing raw audio into AAC format, the proposed steganography is implemented as a new AAC encoder and a new AAC decoder is designed with the function of extracting hidden data. Figure 2.10 shows the steps of the hidden process.

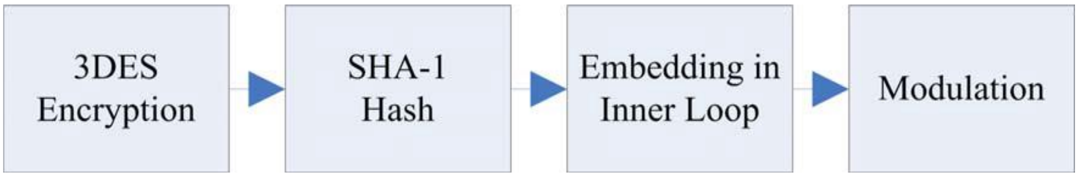


Figure 2.10: Steganography AAC Encoder (Wei et al., 2010, p.4374)

Firstly, 3DES was employed to encrypt the original secret data. Then the encrypted data were embedded bit by bit. Secondly, each bit of encrypted data is embedded into randomly selected AAC frames while the selection is done with a pseudo random bit generator based on SHA-1. Thirdly, controlling the bit-rate in the inner loop with a hidden bit had been proposed. The method has a similar principle with MP3 stego in inner loop and with a minor difference. The difference is that the decreasing speed of length of the encoded bits for AAC files is fairly large; it is possible that the length of encoded bits could never have the same parity with the bit to be hidden. Therefore the program would leave the loop via the exit when the number of times the inner loop has already executed exceeds a threshold. Finally, when program leaves the inner loop via the extra exit, the length of encoded bits does not have the same parity with the hidden bit. A modulation to the length of encoded bits should be conducted to ensure the length has the same parity with the hidden bit in order to avoid the notice of the steganalysis process. To gain the hidden data is simple. Hidden data extraction is by conducting the process backwards.

The drawback of the method is that the modulation will bring extra sound quality degradation. But the modulation could be used in practice therefore authors showed that no perceived sound quality degradation is introduced by this steganography method. (Wei et al., 2010)

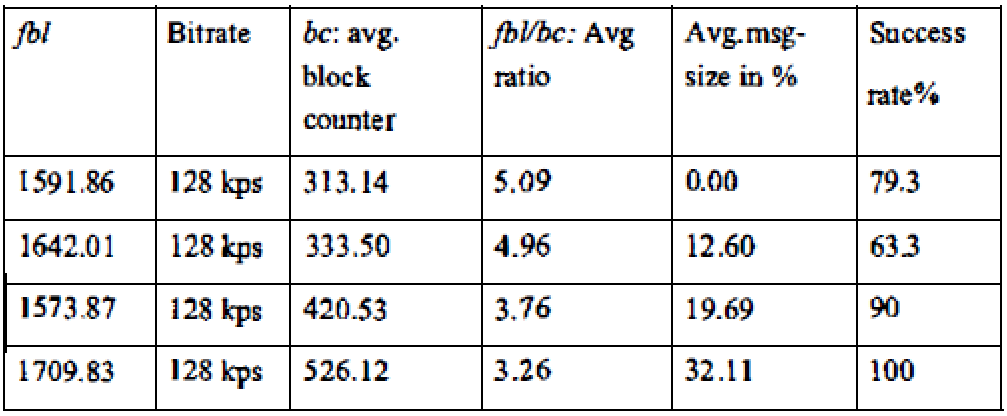
After understanding the way the secret message is embedded into audio cover media we can find out efficient ways to detect and analysis the stego in audio.

2.7 STEGANOGRAPHY DETECTION

Though steganography aims at transmitting images without visual degradation or changes for a naked observer, it cannot dispense with altering spatial and transform level details in order to embed the data. Even though these alterations may not be captured by visual observation, they do manifest themselves for detailed analysis. As it is said by Munirajan et al. (Munirajan et al., 2004) steganography detection and steganalysis could be possible. On other words, steganalysis refers to the detection of embedded information which has been hidden by using steganography techniques (Poisel & Tjoa, 2011). They proposed a fuzzy logic based technique that could be applied to detain images with steganography. The image was JPEG file and the information was hidden on the compressed images' discrete cosine transform coefficients. Their purpose was to establish new techniques and for achieving universal stego detection. The results with their fuzzy inference system for frequency domain stego-detection were convincing enough to acclaim it is a ‘Universal’ stego detection algorithm. Their tests were conducted on random images both from the Internet and from their own database (test images embedded with J-Steg and JP-Hide and Seek) validated their assertions. They found around 85% positive detection with less than 10% false negatives as find stego detection result.

Another steganalysis algorithm was proposed to estimate 2LSB steganography in image by Niu et al. (Niu et al., 2009) in 2009. Their purpose was to establish a novel steganalysis focus on 2LSB embedding method as many technology has developed from LSB to 2LSB in order to withstand previous LSB steganalysis algorithm. They created a masked estimation function which could set weighted value to an image. After weighted a suspected image, they performed message length test which is a set of mathematics formula to deduce whether an image has embedded information or not. Result shows a high accuracy and quick detection speed using their steganalysis algorithm but still have a small estimation error for some tested steganography images.

Due to the improvements of audio steganography, concealing of information in audio media becomes possible for a lot of applications (Dittmann & Hesse, 2004). Differentiating anomalous audio document (Stego audio) from pure audio document (cover audio) is difficult and tedious. Steganalysis techniques strive to detect whether an audio contains a hidden message or not (Geetha et al., 2006). Dittmann & Hesse (Dittmann & Hesse, 2004) proposed an Intrusion Detection System (IDS) to detect hidden channels in Voice over IP (VoIP) applications. They introduced two steganalysis classifiers for MP3Stego and Steghide. After experimental tests, they had positive results that the IDS captured every packet through the VoIP communication channel. For the stego audio detection, they found that the higher the capacity the better the accuracy in detection. Figure 2.11 shows their findings.



fbl	Bitrate	*bc*: avg. block counter	*fbl/bc:* Avg ratio	Avg.msg-size in %	Success rate%
1591.86	128 kps	313.14	5.09	0.00	79.3
1642.01	128 kps	333.50	4.96	12.60	63.3
1573.87	128 kps	420.53	3.76	19.69	90
1709.83	128 kps	526.12	3.26	32.11	100

Figure 2.11: Test result for MP3Stego detection (Dittmann & Hesse, 2004, p.345)

Other researchers have a contribution to audio steganalysis as well. Huang et al. (Huang et al., 2011) introduced a novel steganalysis method that employs a second detection and regression analysis in their study. The proposed method can not only detect the hidden message embedded in a compressed voice over Internet protocol (VoIP) speech, but also accurately estimate the embedded message length. The method was based on the two statistics, which is, doing a second steganography run (embedding information in a sampled speech at an embedding rate followed by embedding another information at a different level of data embedding) in order to estimate the hidden message length. In their second statistics-based steganalysis algorithm, they first ran a poker test for speech parameters. Based on the poker test result, they conducted the second statistical detection and regression analysis. The experimental results show their proposed steganalysis method has great precision in determining the embedding rate in most circumstances, and acceptable errors occur at low embedding rates.

In general, steganalysis procedure starts from packet capture and steganography detection. The process needs support from different steganalysis tools and techniques against different steganography embedding techniques and different forms of cover media. The steganalysis techniques are always improving with new ways of doing steganography processes.

2.8 STEGANOGRAPHY AND NETWORK AUDIO STREAMING FORENSICS

Computer Forensic science protects digital evidence from possible alterations, damage, data corruption, or infection by design or carelessness. It uncovers all relevant files on suspect systems, including overt, hidden, password-protected files (Britz, 2008). Therefore, it is important for forensic investigators to disclose steganography files during evidence acquisition, extraction and analysis.

Erbacher et al. (Erbacher et al., 2009) announced that new and improved data hiding techniques pose a problem for forensic analysts investigating computer crime. Computer criminals are able to hide information using stego-channels available in commonly used document formats, thereby hindering an investigator from acquiring possibly important evidence. They focused on detecting the use of stego-channels in the unused or dead space regions in the Object Linking and Embedding 2 (OLE2) specification used primarily by Microsoft's Office. The detection algorithm which they called OleDetection was focused on detecting the use of stego-channels using a three-step process comprising the detection of dead regions in a document, the extraction of binary data and the generation of appropriate statistics using byte-frequency distribution, and the comparison of the calculated statistic with threshold values to determine whether or not the document contains hidden data. There experimental results showed that their algorithm can correctly identify 99.97% of documents with previous stego-channel techniques with a false positive rate of only 0.65%. Regretfully, the authors didn't mention the extraction of hidden date that has been detected.

Abboud et al. (Abboud et al., 2010) proposed a research on computer forensics on steganography and visual cryptography. They pointed out that numerous novel algorithms had been proposed in the fields of steganography and visual cryptography with the goals of improving security, reliability, and efficiency. They compared two methodologies in their research with similarities and differences. They also proposed an idea for possible algorithm which combines the use of both steganography and visual cryptography. Steganography and visual cryptography are somewhat similar in concept. Ultimately they both are ways of hiding data from prying eyes and in many cases from forensic and security investigators. Some claim that visual cryptography is another type of steganography and some claim the inverse. Although in their basic purpose of hiding information they are indeed similar, when it comes to the data transformation algorithms steganography and visual cryptography take advantage of different methodologies in order to protect their respective payload. In steganography, only the sender and receiver are aware of the hidden data and typically if the loaded file falls into the hands of anyone else they wouldn’t suspect the hidden data. Whereas in cryptography, when someone receives data that is encrypted the first thing that comes to their mind is the question of what is encrypted and how they can decrypt the hidden message. Unfortunately, steganography and visual cryptography tools were used exclusively, it is almost impossible for investigators to uncover hidden or encrypted data. On the other hand, if the detection tools are used in conjunction with other tools, then it makes the lives of investigators much easier and gives them a better chance of detecting suspicious data.

In fact, there are fewer available research reports in the field of forensics on audio steganography which motivates the proposed research. Furthermore, most articles in forensics on steganography were more focused on the detection. None of them gave a guide for typical forensic procedures in steganography.

2.9 SUMMARY ISSUES AND PROBLEMS

Originally, steganography is aimed to be created for enhancing security in order to achieve information confidentiality. With the developing of computer technology, steganography is widely used in the form of digital media. Ironically, in many situations, steganography has been misused and become a threat to security. Digital forensics is a science of investigation of evidence; steganography is considered as a threat. Steganography technology can hide evidence effectively and remain silent unless a forensic investigator means to look for it. Although a forensic investigator tries to look for steganography information, it is still an arduous task because there are many types and thousands of tools that can embed and hide information in cover media. An Investigator with less experience and poor technical support can easily overlook the potential areas and files for evidence. Therefore, to identify steganography, the investigator will need the knowledge of steganography, powerful steganography detection and analysis tools, and effective guidelines.

Generally, seizing steganography files in a computer means a lot of work because nowadays the size of a hard drives has become much bigger. Thousands of files in a computer are all needed to be investigated for steganographic information during forensic investigation. Accordingly, some researchers have developed methods for developing steganography tools to detect steganography in files. Additionally, some websites provide a steganography service online which means there could be no steganography tools installed on computer but only steganography files. Thus, in most cases, searching steganography tools on a suspect computer is not efficient for forensic investigation. Furthermore, online chat software becomes more and more popular, audio stream as a cover media is widely used to transfer steganography. There are many types of encoder in audio streaming. The encryption makes steganalysis more difficult during investigation. Fortunately, there are only three main embedding techniques in audio steganography which means most audio steganography can be recognized and extracted. On the other hand, some researchers started to develop more complex approaches and embedding methods. With these novel methods, audio steganography is unlikely to be extracted even when it has been detected and found. With the availability of audio steganography, some intelligent criminals manipulate the available algorithms with cryptography algorithms for use. Some researchers had experiments for steganography plus visual cryptography that makes the hidden information hard to be detected by forensic investigators. Moreover, opposite to steganography detection tools, it is easy to obtain free audio steganography tools on the Internet that has also raised the misuse of steganography. Therefore, a forensic examination in relation to audio steganography and detection is needed to evaluate the impact of audio steganography tools on forensic investigation.

Guidelines had been developed by some researchers in their steganalysis research to detect steganography files. But most of them were focused on technology and none of them were focused on audio steganography extraction to obtain hidden information. Most researches were from security point of view but not forensic. Similarly, some forensic researches were focused on steganography detection on suspect area of hard drive and applications. Audio steganography forensics is also a new research area with few contributors.

Obviously, there is lack of research on steganography when conducting digital forensic investigation related to audio streaming forensic. As many steganography tools have the ability to hide information in audio streaming through live network environment, it becomes necessary for forensic investigators to consider audio steganography as a breeding ground for illegal activities. Thus, an audio steganography evaluation for digital forensic investigation and a guideline on how to conduct forensic investigations on audio steganography in streaming is needed for forensic investigators when they conduct similar investigations.

2.10 CONCLUSION

Chapter 2 provided the literature review of the knowledge in related to the research area of the thesis. Firstly, it presented an overview background knowledge for the definition of steganography and history of steganography. Then the technology of steganography has been discussed from reviewing the current publications and research reports. Issues of steganography have been identified. The Literature of steganography detection and steganalysis have been reviewed and discussed to gain knowledge and guidance for conducting an experimental study.

Audio steganography and forensic investigation of audio steganography has been reviewed. The current state of audio steganography in forensic investigation has been evaluated. Therefore, the determined proposed research will focus on advancing the body of investigating on audio steganography. Moreover, issues and problems from current studies have also been outlined to identify and determine possible research questions.

Chapter 3 will therefore designed the research methodology based on the understandings from Chapter 2. A review of similar studies will be conducted to evaluate the methodologies that have been implemented by others. Research questions and hypotheses will be derived. Expected outcomes will be discussed and methods for the collection, treatment, and analysis of data established. Also the limitations will be discussed after designing the research.

Chapter 3

RESEARCH METHODOLOGY

3.0 INTRODUCTION

Chapter 2 has reviewed the literature which are related to the topic area of this thesis. The reviewed areas include steganography techniques, audio steganography embedding method, steganalysis, and forensic investigation of digital steganography. This information links with chapter3 as a researchable question must be selected in relation to the identified problem areas and the research methodology decided.

A number of similar studies in the proposed research area will be sourced and evaluated in Section 3.1 to learn from the experience of other researchers' working within similar areas of study of how to do research. Different from the literature review in Chapter 2, the review in Chapter 3 will focus on the methodology, research question, and hypothesis of the related work. Section 3.2 will outline the main research question and hypotheses as well as the design of the proposed research procedures. Section 3.3 will cover the data requirements for all the data that can be generated from the structured experimental tests. Data requirements consist of data collection, data treatment, and data analysis.

In Section 3.4, the limitations of the research will be discussed to identify the restrictions that may influence the transfer of findings and potential biases.

3.1 REVIEW OF SIMILAR STUDIES

In order to develop the methodology for this research, five independent and relevant studies have been evaluated and reviewed. There have been a number of references of audio steganography in Chapter 2 to address the process of audio steganalysis, detection and forensic investigation. The following studies have been chosen for relevance to the research area, methodology, and experiments regarding forensic audio steganography detection and analysis.

The first study by Zheng et al. (Zheng et al., 2012) in Section 3.1.1 proposed a novel method using Feature Matching to identify Least Significant Bit (LSB) steganography software. It gives a new thinking in steganography detection and the steganalysis field. The second and third studies in Section 3.1.2 and 3.1.3 are related to audio steganography and audio steganalysis respectively. The second study by (Xu, et al., 2011) presented an adaptive method for steganography in audio streams to enhance LSB embedding method. The third study by (Geetha et al., 2006) investigated the use of Genetic Algorithm (GA) to aid autonomous intelligent software agents capable of detecting any hidden information in audio files.

The fourth study by (Erbacher et al., 2009) in Section 3.1.4 is related to forensic investigation in the use of steganography channels in Microsoft Office applications. An capability anti-forensic of steganography has also been identified in the research. The final study by (Leung & Chan, 2007) discusses their experience in network forensic analysis on Skype audio traffic with blocking and detecting the communication.

3.1.1 Feature Matching to Identify Steganography Software

Zheng et al. (Zheng et al., 2012) in the article - *A Method Based on Feature Matching to Identify Steganography Software* proposed an approach to identify steganography software by feature matching based on the principles of Least Significant Bit (LSB) steganography algorithm. The findings of the research showed the feature matching method can reliably identify a variety of common steganography software based on LSB algorithm. Furthermore, authors are trying to develop the algorithm to identify steganography software using other embedding algorithms.

The Authors argue that the current steganography detection systems prefer to distinguish whether an image is a stego image or not by comparing the statistical properties of the image before and after message hiding. The drawback of those systems is that after detection it is nearly impossible to extract the hidden message from stego image for forensic purposes, because those systems did not understand which steganography algorithm was used. But Zheng et al. (Zheng et al., 2012) developed the method to identify the steganography software which can raise the degree of attention to the suspicious targets, and provide critical clues and basis for judgment of stego images and extraction of the hidden message.

Authors used experiment methodology in their research with support from relevant studies, and previously developed steganography and steganalysis algorithms such as understanding binary software, identifying the cryptography algorithm by matching the semantic of the algorithm, and the malware detection algorithm. They developed the framework of the methodology into four phases.

In the first phase, the authors briefly researched the LSB embedding algorithm. LSB embeds a message into the cover image by changing the LSBs of the cover image according to the message bits to get the stego image. Thus, they evaluated the common techniques of LSB steganography software and created the core instruction template.

Moving to the second phase, authors presented an algorithm to detect whether the software satisfy the core instruction template or not. The algorithm was implemented based on the IDApro. The framework of algorithm of software identification is show in figure 3.1. The algorithm identifies steganography software by matching all the instructions in the template with the binary intermediate code created by IDApro in the binary program. After that, they built the instruction dependencies and templates to process the matching which is the detection part of the algorithm. The Stegtool Detector was created and used to match the style of each instruction and the style of each code in the template, and check whether the value of the match variables have the same relationship. If the relationship is the same then the two instructions were matched. Thus the algorithm determined that the software is LSB steganography software. Otherwise, the software might not be the steganography software or steganography software based on other embedding algorithms.

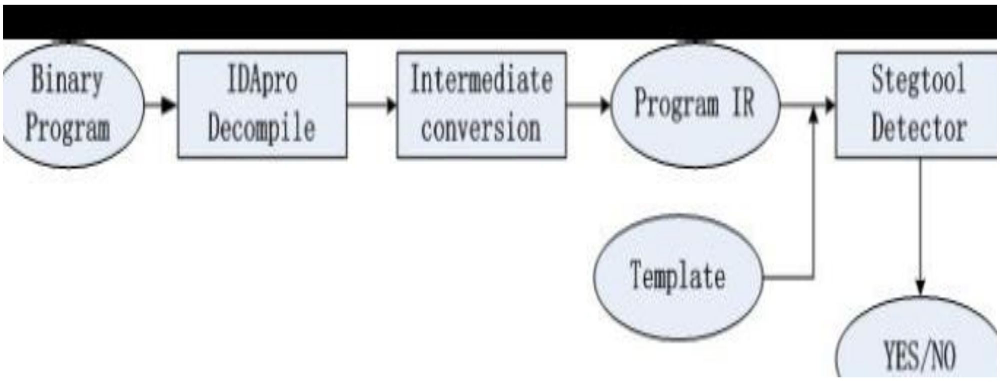


Figure 3.1: The framework of algorithm of software identification (Zheng et al., 2012, p.991)

After developing the identification algorithm, Zheng et al (Zheng et al., 2012) started experimental tests on Windows XP system in the third phase. They ran two test sets respectively. In the first test set, they used the proposed algorithm to detect 14 chosen LSB steganography software types. The results showed that 9 of 14 steganography software can be detected effectively. In the second test set, they selected 15 steganography software types based on other embedding algorithms. The result showed that no steganography software was detected.

In the final phase, they evaluated the experimental results and found out their algorithm can detect LSB steganography software effectively. Their experiment could have improvement in the future by having an algorithm that is able to detect more steganography software.

3.1.2 Information Hiding within Network Audio Streams

Xu et al. (Xu, et al., 2011) realized that the existing VoIP steganography only focus on information hiding in the LSB bits of network audio streams, yet this algorithm raised some security threats, where the hidden information could be easily detected and removed. Thus, they proposed an adaptive VoIP steganography (AVIS) approach to hide information in network audio streams. The AVIS consists of two parts which are value-based multiple insertion (VAMI) and voice activity detection dynamic insertion (VADDI). VAMI works by dynamically select multiple bits based on the VoIP vector value, while VADDI dynamically changes embedding intervals to avoid detection and attacking.

The preliminaries had been reviewed by authors includes VoIP information embedding process which is shown in figure 3.2, LSB steganography, and audio cover quality measurement. The audio quality is measured using mean opinion score (MOS) standard which is shown in table 3.1.

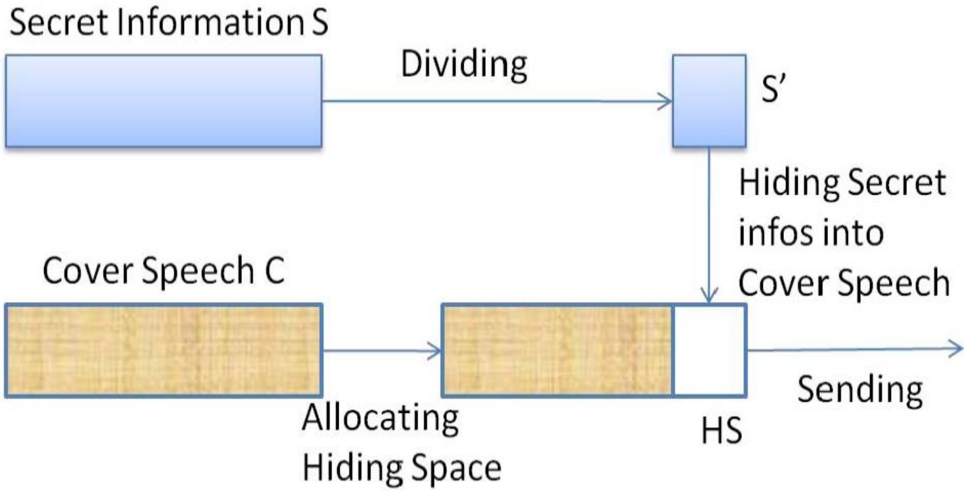


Figure 3.2: VoIP information embedding process (Xu, et al., 2011, p.612)

Table 3.1: Mean Opinion Score standard (Xu, et al., 2011, p.613)

Mean Opinion Score (MOS)		
MOS	**Quality**	**Impairment**
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

From researching the LSB embedding method, the authors realized that LSB actually has potential threats due to its vulnerabilities in detecting. Then they proposed the AVIS embedding algorithm. The VAMI and VADDI were introduced respectively. In VAMI, the algorithm works by selecting the smaller vector which has smaller bit-weight of the potential embedding bits to hide information. Because whether the potential embedding places available was decided by sound quality, authors used MOS to estimate sound quality to judge the availability of potential embedding bits. Although the embedding place was dynamically decided, the embedding intervals were still fixed. Authors introduced VADDI algorithm to judge whether the VoIP is silent or in conversation, then the algorithm dynamically decided the embedding intervals based on frame as the unit.

To evaluate the performance of AVIS steganography algorithm Xu et al (Xu, et al., 2011) implemented the algorithm in Linphone which is a famous open-source VoIP platform. 25 audio samples were sent between 2 platforms. Emiprix Call Analyzer and the Pesq tool were used to intercept the VoIP stream in the receiver end to evaluate the AVIS steganography performance. The anti-detecting performance, voice quality, and latency were evaluated. To be specific, the probability of detection for LSB steganography is 20% while for AVIS steganography is only 8%. The MOS standard for 25 samples is above grade 4. The only problem is with the latency test. The AVIS steganography algorithm occupies one third of the process time. However, with regard to latency of the Internet, up to 20ms is the cost and such increasing in encoding processes will not be perceivable by humans (Xu, et al., 2011).

3.1.3 Audio Steganalysis using GA

"Steganography is used to avoid drawing suspicion to the transmission of a hidden message in multimedia. This creates a potential problem when this technology is misused for planning criminal activities." Geetha et al. (Geetha et al., 2006). Thus, they had investigated the use of Genetic Algorithm (GA) to aid autonomous intelligent software agents capable of detecting any hidden information in audio files. GA employed various Audio Quality Metrics (AQMs) to calculate on audio signals. Those AQMs operated as an Automatic Target Detection (ATD) system. They designed the ATD audio steganalysis system relied on the choice of audio quality measures and the construction of GA based rule generator.

First, the authors started a learning stage that teaches AQMs agents how a stego object looks like. They called it the training which is accomplished using GA and utilize a database containing cover and stego audio samples. The learning stage consisted of a de-noiser, feature selector, AQM evaluator, and data set formation. This stage is to generate a set of rules that will be used in classification of marked and non-marked signals for detecting stage. In detecting stage, the network traffic was monitored constantly. AIU (Audio Identification Unit) was used to identify any audio file and the information was passed to the detector agent which is individual software working autonomously. Figure 3.3 shows the learning stage and detecting stage.

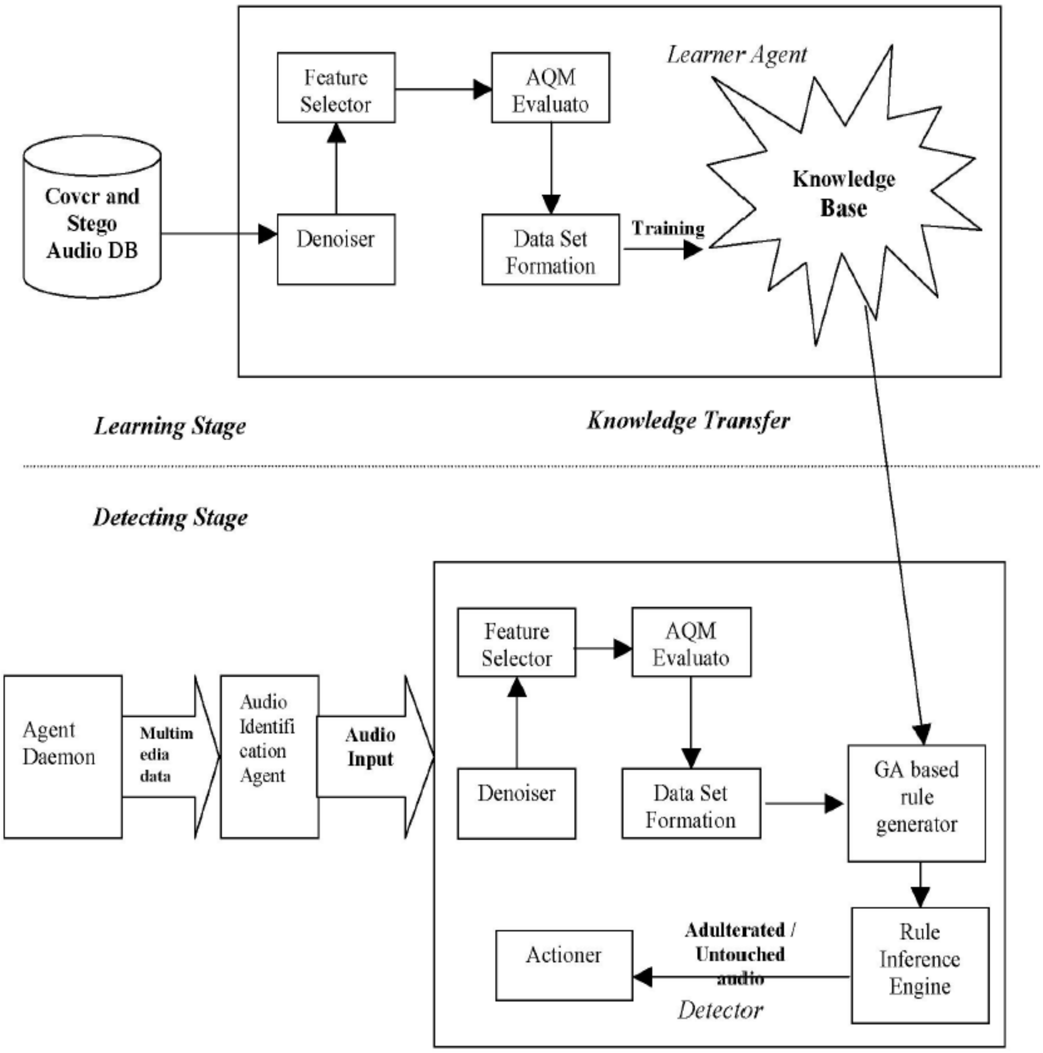
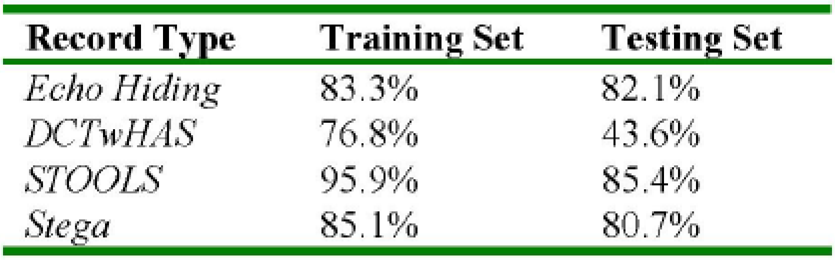


Figure 3.3: Architecture of GA based autonomous multi-agent audio steganalyzer

(Geetha et al., 2006, p.3)

In experimental tests, four audio information hiding techniques were accomplished. Therefore, authors created GA based rules and interconnected to each other. These four techniques were frequency masking with DCT, echo watermarking, Steganos, and Stools. 600 audio records such as music, male speech, female speech, silence file, and instrumental audio were implemented into a training database. Among those audio records, 300 were stego file and 300 were original file. 100 independent audio records containing 50 stego file and 50 original file were kept in test database. The Author conducted two experiments. In the first experiment, the training database was analyzed through the proposed steganalyzer. The second experiment, authors analyzed the test database using the proposed steganalyzer. The result of both experiments was shown in figure 3.4.



Record Type	Training Set	Testing Set
Echo Hiding	83.3%	82.1%
DCTwHAS	76.8%	43.6%
STOOLS	95.9%	85.4%
Stega	85.1%	80.7%

Figure 3.4: The detection rate of the steganalyzer (Geetha et al., 2006, p.5)

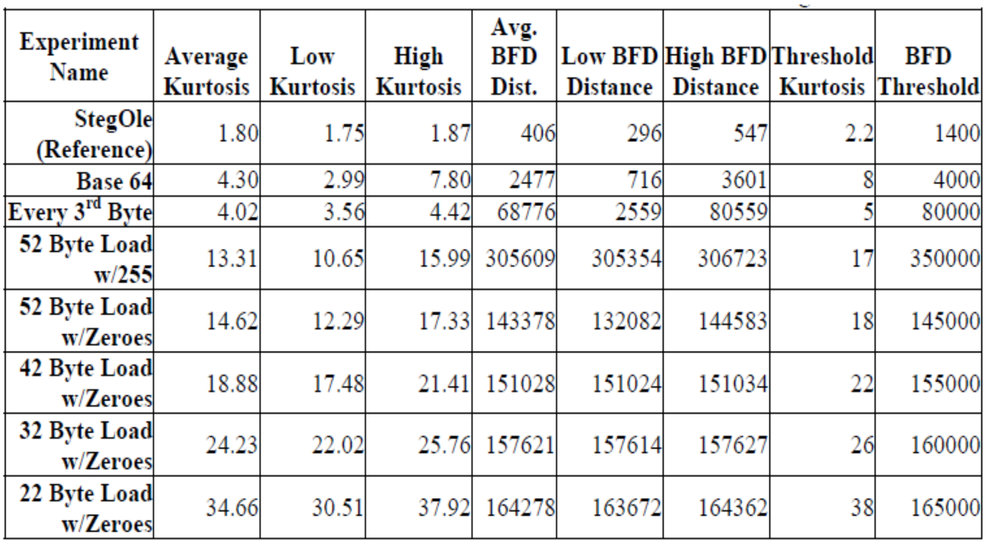
3.1.4 Forensic and Anti-Forensic of Steganography

Erbacher et al. (Erbacher et al., 2009) indicated that digital forensic is the science of collecting, discovering, and preserving digital data for use in court. Acquiring digital data from a target system is not an easy task because criminals are able to hide information using stego channels available in commonly used document formats such as Microsoft Word, Excel, and PowerPoint. Thereby, authors proposed the OleDetection (Object Linking and Embedding Detection) algorithm to detect stego channels using three-step process. This three-step process included the detection of dead regions in document, the extraction of binary data and the generation of statistics, and the comparison of the calculated statistics with threshold values.to determine whether or not the document contains hidden data.

Since common office documents created by Microsoft Office programs are implementations of the OLE2 specification (OLE2-formatted documents), authors used MS Word, Excel, and PowerPoint documents for their experiments. Each MS Word document was collected in an adhoc fashion from random websites. Eventually, 293 MS Word documents were acquired, ranging in size from 20.5 KB to 4858 KB. The Excel data set contained 109 files ranging in size from 10 to 8108 KB while the PowerPoint data set contained 99 files ranging in size from 26.5 to 31148 KB. Among those documents, 50% were randomly modified using StegOle to hide message. The OleDetection tool is a console-based program that used the kurtosis statistic in combination with the BFD to identify documents that were modified to contain a stego channel.

Several trial experimental runs were performed to determine the best threshold. Based on these runs, the first experiment used thresholds set to a single standard deviation from the average. The kurtosis value was set to 1.81 and the BFD distance was set to 490. The results showed on average a 69.2% detection rate. After another 4 experiments the author found the most effective threshold which kurtosis values was 2.2 and BFD distance was less than 1400 gave an average 99.97% detection rate. Then an experimental validation test was performed using Excel and PowerPoint data set which gave 100% of detection rate for Excel and 90.5% of detection rate for PowerPoint. After detection, the stego data were extracted by OleDetection for using in forensic purpose.

Finally, the authors examined a method of encoding the stego channels in OLE2 documents that the stego document is statistically similar to the original document from an anti-forensic perspective. They discovered that diluting the payload density is a positive way to reduce the probability of been detected. Therefore, a series of experiments were performed with different payload size and different hiding space. Figure 3.5 shows the performance of different payload size.



Experiment Name	Average Kurtosis	Low Kurtosis	High Kurtosis	Avg. BFD Dist.	Low BFD Distance	High BFD Distance	Threshold Kurtosis	BFD Threshold
StegOle (Reference)	1.80	1.75	1.87	406	296	547	2.2	1400
Base 64	4.30	2.99	7.80	2477	716	3601	8	4000
Every 3rd Byte	4.02	3.56	4.42	68776	2559	80559	5	80000
52 Byte Load w/255	13.31	10.65	15.99	305609	305354	306723	17	350000
52 Byte Load w/Zeroes	14.62	12.29	17.33	143378	132082	144583	18	145000
42 Byte Load w/Zeroes	18.88	17.48	21.41	151028	151024	151034	22	155000
32 Byte Load w/Zeroes	24.23	22.02	25.76	157621	157614	157627	26	160000
22 Byte Load w/Zeroes	34.66	30.51	37.92	164278	163672	164362	38	165000

Figure 3.5: Detection thresholds for different payload (Erbacher et al., 2009, p.93)

Their experimental result showed that limiting the payload to 32 bytes per 512 regions is probably sufficient to avoid most detection attempts which offer a possible anti-forensic opportunity in OLE2 documents steganography. This result showed that steganography can hide a reasonable amount of information and is hard to be detected.

3.1.5 Network Forensic on Encrypted Skype Traffics

The fifth article I reviewed is related to audio network traffic detection and forensics. Leung & Chan (Leung & Chan, 2007) shared their experience in detecting and blocking Skype traffic as well as forensic analysis in an article – *Network Forensic on Encrypted Peer-to-Peer VoIP Traffics and The Detection, Blocking, and Prioritization of Skype Traffics.* Skype is a popular P2P voice over IP application. However, the ability to traverse network address translation (NAT) and bypass firewalls makes Skype a possible threat to enterprise level network security.

Motivated by the above reason, the authors conducted a lab experiment method for investigating Skype. Firstly, they started with **model construction** which was formulating an application layer event graph for the Skype activities.

Secondly, authors began **collection of evidence**. They executed the experiment procedures to perform the Skype phone call activities at application layer. The tool WireShark was used to capture all incoming and outgoing packets during the experiments. Thirdly, authors **analyze the evidence** by performed detail investigation and analysis on the digital evidence collected and compared the timestamps against those activities. Finally, they **presented the evidence**. The analysis results were arranged and presented into a Skype communication framework including the entities involved and the 15 stages in a Skype conversation. The Skype communication framework is shown in figure 3.6.

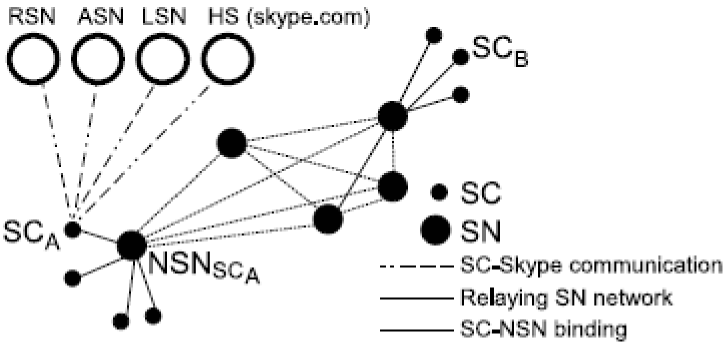


Figure 3.6: Skype Communication Frameworks (Leung & Chan, 2007, p.3)

In the Skype communication framework, HS stands for Skype HTTP Server. SC is Skype client. SN is super node. RSN is registration super node. ASN is authentication super node. LSN is location super node. NSN is neighbor super node. With the detailed transport layer communication framework of Skype, authors can draw policies for Skype traffics detection. In practical, they identified the UDP communication port which enables controlling the Skype traffics by blocking the corresponding sockets.

From their experiments, they were able to identify the sockets associated with Skype applications, formulate the detection rules, and block the socket completely by performing network forensic analysis to generate a communication framework of Skype to the transport layer.

3.2 RESEARCH DESIGN

Five related research projects have been identified and analyzed in Section 3.1 to develop an adaptive research methodology for the proposed research. Therefore, an experimental methodology is chosen to conduct the proposed research. Experimental methodology tries to isolate and control every relevant condition which determines the events investigated, so as to observe the effects when the conditions are manipulated (Clarke, 2005). In the proposed research, several audio steganography tools with different steganography techniques will be experimented to test the reliability, efficiency, and veracity. Several audio steganography detection and extraction tools will also be tested for the same purpose. A selection of detection tools will be tested in an online environment to test the live detecting ability. The most efficient tools will be chosen and used in an organized forensic examination based on a case scenario to learn what evidence can be found in the computer system after creation and receiving of the audio steganography files. The result from the experiment will be analysed in order to answer the research question. The procedure of the investigation during the experiments will be used as a guideline for audio steganography evaluation.

In Section 3.2.1, a brief discussion of the five published research examples in Section 3.1 will be reviewed. Some selected problems from Section 2.9 in the areas of forensic investigation of audio steganography in a live environment will also be discussed in Section 3.2.2. As a well-developed hidden technology, audio steganography is hard to detect during forensic investigation unless it is a specific evidence requirement. Therefore, evaluation of steganography in forensic investigation is important. The development of the research question will be constructed in Section 3.3.3 based on the literature reviewed in Chapter 2, and the review of similar research studies in Section 3.2.1. Research hypothesis will be established after determining the research question and sub question. Research model will be presented in Section 3.2.4. In the end, the research data map will be presented to show the logical connection among each stage of the research processes as well as the research question and hypotheses.

3.2.1 Summary of Similar Studies

In Section 3.1, five similar research studies in the research area of information hiding and detection, audio steganography, steganography forensics and anti-forensics, and audio stream detection were reviewed. The summary of learning from those studies was presented in this section to provide guidance for the proposed research.

Zheng et al. (Zheng et al., 2012) presented a feature matching method to identify steganography software in their studies which is an important research direction of steganography forensics. They created a matching template with popular steganography tools based on LSB hiding algorithms and tried to detect these tools with their feature matching method. Although their study was not focusing on audio steganography detection and forensic it created new thinking in steganography forensics that is detecting steganography tools instead of detecting steganography files. Their research has a positive impact on this proposed research as it will evaluate the possibility of detecting the audio steganography tools using current popular steganography detection tools. On the other hand, the process of Xu et al.'s study (Xu, et al., 2011) has some similarity with the proposed research. They researched a novel steganography technique that was better than a LSB algorithm to hide information within network audio streams. In the early stage, more than two steganography algorithms will be used to hide information with a group of network audio streams to evaluate the performance of different steganography algorithms and tools.

Geetha et al. (Geetha et al., 2006) presented an algorithm that used multi agent architecture as an automatic target detection (ATD) system to detect hidden information in audio files. Their experimental results provided promising detection rates. It is an important support to the proposed research that detection of audio steganography is able to be achieved. The proposed research must be valuable, meaningful, and feasible from technical point of view. Then, Erbacher et al. (Erbacher et al., 2009) raised the topic from a technical to a steganography forensic level. In their study, forensic and anti-forensics of steganography in OLE2-formatted documents (Microsoft word, excel and access) was presented. Computer forensics can be considered a series of analytic and investigative methods applied to evidence that is magnetically stored or encoded that must be identified, collected, examined, and preserved. However, acquiring digital evidence from a target system is not an easy task. The situation is getting harder because of the development of steganography and information hiding techniques. The proposed research will evaluate a created case scenario that audio steganography files need to be detected and the hidden information has to be extracted as digital evidence in a series of forensic investigation processes.

Leung and Chan (Leung & Chan, 2007) conducted an empirical study to detect and block Skype Traffic (network audio streams) in a network forensic investigation. Although their research was not relevant to steganography, it is supportive to parts of the proposed research. Their research showed the experience in network audio streams detection, block, and forensic which will be done in the proposed research with adding steganography in network audio streams. The knowledge and experience from those five research studies suggests that this proposed research is significant and achievable in the named research areas.

3.2.2 Review of Problem Areas

In chapter 2, section 2.9 the issues and problems that steganography poses to network security and challenges to forensic investigation has been summarized. As discussed in section 2.8 forensic investigations of steganography has always been considered as a difficult job as the developing of tools and techniques is challenging. Zheng et al. (Zheng et al., 2012) posted a feature matching method to match the most popular steganography tools with data in order to conduct steganography detection. As discussed in section 2.9, detecting steganography tools in digital forensics has limitations. The use of steganography tools cannot be evidence that prove criminal activities during an investigation. A Forensic investigator needs to find steganography files as direct evidence to present in the court.

As presented in section 2.9, many researchers have developed more advanced audio steganography algorithms. Xu et al. (Xu et al., 2011) developed a modern LSB audio steganography embedding method in order to achieve less detection rate. Their work proves that advanced steganography techniques and tools improves the level of forensic investorgators' work. On the other hand, some researchers conducted research on steganography detection. Geetha et al. (Geetha et al., 2006) presented an ensemble of autonomous multi-agents and a genetic algorithm to detect audio steganography in their experiments. Although their method can detect audio steganography files, the detection rate is still not perfect as below 80%. Refer to real life, it means over 20% of criminals could possibly be ignored after the evaluation of steganography using current technologies.

The amount of evidence has always been a big issue for forensic investigators. Also anti-forensics in steganography are noted by Erbacher et al. (Erbacher et al., 2009). They provided a method that encoding the steganography files statistically to be similar to the original files prevents detection by forensic investigators. Therefore, a systematic evaluation of challenges in related to conducting forensic investigation (detection and analysis) on audio steganography is necessary.

Additionally, most researchers were focused on steganography and steganalysis techniques as reported in section 2.9. Current audio steganography forensic research were neither mentioning the procedures to conduct forensic investigation on audio steganography nor try to extract steganography content from detected audio cover files. Leung and Chan (Leung & Chan, 2007) proposed a method of detecting network audio streams in their research without capturing and extracting audio files. Thus, understand how to capture audio steganography is not enough in forensic examination. Extraction of credible content from audio steganography files is also an important signal to determine the success of a steganography forensic investigation.

Finally, a guideline can be helpful for forensic investigators during audio steganography investigation. With no guideline in related to audio steganography forensics currently, the development of such a guideline after systematic experimenting of how to conduct a forensic investigation on audio steganography is a useful outcome for research.

3.2.3 The Research Question

The literature review in Chapter 2 has provided the foundation knowledge in the proposed research area of evidence acquisition preservation and analysis of conducting digital forensic investigation in audio steganography. Some of the literature has covered steganography embedding algorithms, steganography detection and analysis, steganography attacks and misuses, and evaluation of steganography tools. The processes of digital forensics were presented and specific literature was reviewed regarding digital forensics in steganography. Furthermore, the preceding review of similar studies in section 3.1 has provided a technical background into audio steganography creation, detection, and analysis as well as forensic perspective of investigating steganography. All of these similar studies have provided related information into the realm of digital forensic in audio steganography.

The research question was developed based on the literature review in Chapter 2, and the review of similar studies in Section 3.1. The misuse of audio steganography is a potential risk to forensic investigations has been discovered. Additionally, it has been discovered that detection of audio steganography is a major issue in forensic investigation. The extraction of audio steganography content is another issue in presenting evidence in digital forensic investigation. Such issues have addressed serious question into digital forensic investigation.

Table 3.2 displays the main research question and the related hypothesis which has been developed from the preceding information.

Table 3.2: Main Research Question and Related Hypothesis

Main Question 1: What are the procedures and challenges when conducting

digital forensic investigation for audio steganography?
Asserted Main Hypothesis 1: The current state of digital forensic tools for steganography detection causes problems on digital forensic investigation for audio steganography.
Main Question 2: How credible is the extracted content from audio steganography in relation to the forensic evidence purpose?
Asserted Main Hypothesis 2: The detected audio steganography file can be extracted with the support from selected tools.

According to main research question and asserted main hypothesis, the sub questions have been devised in order to answer various linked components to the main research question and are set out in Table 3.3.

Table 3.3: Sub Research Questions

Sub Question 1: Does network monitor and packet capture tool WireShark be able to capture potential audio steganography packets?
Sub Question 2: Is it easy to embed various types of contents into audio streams with support from Mp3Stego, Openpuff, and S-Tools?
Sub Question 3: How efficient is StegAlyzerAS and StegAlyzerSS in order to identify audio steganography?
Sub Question 4: Can StegAlyzerSS extract the steganography content that has embedded into audio streams? If it can’t, can Mp3Stego, Openpuff, and S-Tools extract the steganography content?

Hypotheses have also been developed for each sub question that proposed above. Table 3.4 shows the hypotheses accordingly.

Table 3.4: Sub Research Questions Associated Hypotheses

Hypothesis 1: WireShark can capture network packets which have passed the network card; it cannot restore information of the packets for steganalysis purpose.

Hypothesis 2: The chosen three steganography tools are quick and efficient for everyone to create their own audio steganography files.
Hypothesis 3: Using StegAlyzerAS and StegAlyzerSS, over 90% of audio steganography tools and files can be detected and identified.
Hypothesis 4: StegAlyzerSS is able to extract the steganography content from audio steganography files created by Mp3Stego, Openpuff, and S-Tools.

3.2.4 Research Model

The aim of the research is to advise on audio steganography detection and analysis for the digital forensic investigation. The experimental method (Clarke, 2005) is conducted to evaluate the outcomes to answer proposed research questions. The experimental network is client-server based. The two clients machines are installed with windows 7 operation system with all security features turned on. Audio streams are going to be transmitted between two clients.

The Figure 3.7 shows the research model that consists of five phases in order to construct the experimental research to find out answers to the research question.

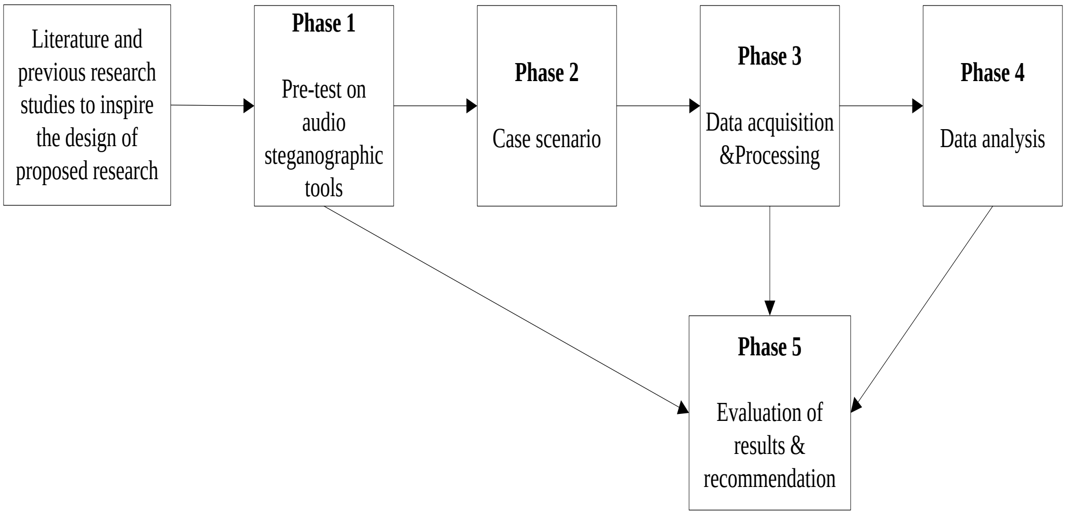

Literature and previous research studies to inspire the design of proposed research
Phase 1
Pre-test on audio steganographic tools
Phase 2
Case scenario
Phase 3
Data acquisition &Processing
Phase 4
Data analysis
Phase 5
Evaluation of results & recommendation

Figure 3.7: Research Model

Phase 1 is a preliminary test to experience three chosen audio steganography tools

(Mp3Stego, Openpuff, and S-Tools) and two steganography detection tools (StegAlyzerAS and StegAlyzerSS) as well as network capture tool (WireShark). The experimental result will directly determine which tools are going to be used to conduct the case scenario in Phase 2.

Phase 2 is to develop a case scenario which is very similar to a real life audio steganography misuse case for digital forensic investigation. The purpose of Phase 2 is to generate the raw data and evidence data for acquisition & processing in Phase 3 as well as for analysis in Phase 4.

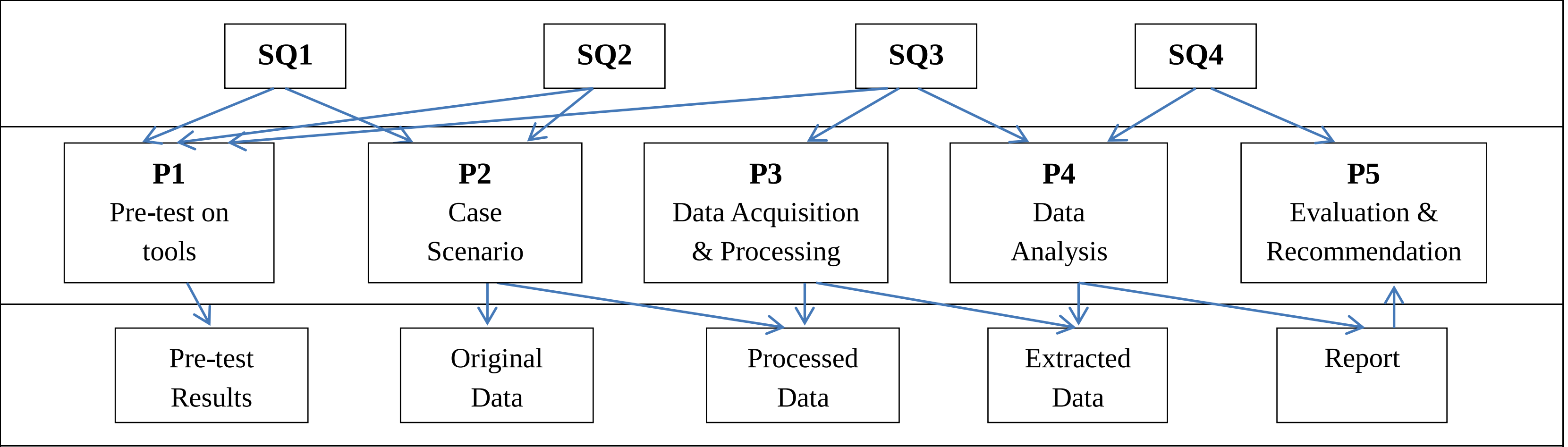
In Phase 3, the evidence that created from case scenario is going to be acquired and processed follow general digital forensic procedure. Forensic tools write blocker and EnCase are going to be used. In Phase 4, the processed data is going to be extracted and restored. Forensic analysis and steganalysis of the evidence will be conducted. The result is going to be compared with the original data in order to answer the research questions.

Phase 5 is to generate recommendations of initiating digital forensic investigation on audio steganography after recoding the entire investigation in Phase 3 and Phase 4 step by step.

3.2.5 Data Map

Table 3.5: Research Data Map

Main Research Question	1. What are the procedures and challenges when conducting digital forensic investigation for audio steganography? 2. How credible the extracted content from audio steganography is in relation to forensic evidence purpose?
Sub Questions	**SQ1** **SQ2** **SQ3** **SQ4**
Research Phases	**P1** Pre-test on tools **P2** Case Scenario **P3** Data Acquisition & Processing **P4** Data Analysis **P5** Evaluation & Recommendation
Data Collection	Pre-test Results Original Data Processed Data Extracted Data Report
Main Research Hypotheses	1. The current state of digital forensic tools for steganography detection causes problems on digital forensic investigation for audio steganography. 2. The detected audio steganography file can be extracted with the support from selected tools.



3.3 DATA REQUIREMENTS

During the testing phases there are a number of requirements for different aspects of data handling. The tests will be conducted in a laboratory environment consisting of fully secured network and two desktops with windows 7 installed. There are five sets of data that will be generated in the test including pre-test results, original data, processed data, extracted data, and report. In order to evaluate the procedures and challenges which audio steganography has addressed to digital forensic investigation, all these five data will be collected then be analyzed and the results will be reported in Chapter 4.

The data requirements that needs to be addressed in the research phases are separated into three categories: data collection, data treatment, and data analysis. All these data requirement categories will be discussed detail to ensure the data is collected, treated and analyzed accurately.

3.3.1 Case Scenario

Case Scenario is designed for simulating the digital forensic investigation on audio steganography after pre-test phase. In the case scenario, audio steganographic is misused to commit crime activities.

Antonio Braga is the drug lord in America. He controls 80% of drug dealers and drug trading activities. The International Criminal Police Organization (ICPO) and local police are keeping eyes on his daily activity as well as communication channels. Braga knows about it and manipulates several methods to deal with other drug dealers in order to finish drug trading. One of his favorite methods is sending several drivers in fast racing cars to cross the board between America and Mexico to trade drugs. He does not contact with those drivers face to face but though Tencent QQ which is instant messenger software. All the conversation is follow one way structure which is Braga to the drivers.

The greatest driver for Braga is called Dominic Toretto. Therefore, most time Braga sends time and the route to Toretto to deliver the drugs. ICPO and local police eavesdrop on the communication between Braga and Toretto but could not find any evidence. Then the ICPO and local police suspect that Braga uses steganography techniques to hide drug dealing information in the audio streams during communication with Toretto.

After ten months observation on Toretto, ICPO caught him during drug trading in Mexico with the help of Mexican police. Then Toretto was extradited from Mexico to American. During the interrogation, Toretto testifies that all his criminal activities are following the orders from Braga. The orders consist of time, addresses, and routes were hiding in the audio streams sent by Braga. ICPO and local police arrested Braga. A law enforcement team is sent to seize evidence from Braga's hard drive to extract and analyze the potential steganography tools and files. Forensic investigation tools EnCase will be used to analyze the evidence.

3.3.2 Data collection

Data collection is an important process in the proposed research. In order to make the proposed research rigorous, the data generated from tests needs to be collected in a correct form and accurately.

The first data needed to be collected are the pre-test results from three different steganography tools (Mp3Stego, Openpuff, and S-Tools). The cover files will be MP3 files and WAV files. The secret information will be in text format and image format (JPEG). The chosen steganography tools will be used to embed those text file and image file into the MP3 and WAV files seriatim to create audio steganography files. The steganography files and cover files will be evaluated by WinMD5 to generate hash values of each. The comparison result will be stored in the excel form. Then the created steganography files will be sent between two windows 7 PC using Tencent QQ through local area network. WireShark will be used to capture and analyze all the packets between the two PCs in order to find out whether it can detect audio steganography files or not. WireShark will generate a report after capturing the transmissions between two PCs.

In addition, the collection of original data, processed data and extracted data will be generated during the simulation of case scenario in the laboratory environment. During this process, the simulated activities, the cover files, the steganography content files, hash values, the WireShark capturing results, and the date and time will be collected as original data that is input to an excel file.

The processed data of the experiment will be collected using a forensically acceptable method that the write blocker is going to be used to acquire the evidence from the suspect’s PC in order to maintain data integrity. Then the forensic software EnCase will be used to create an image of the case for data analysis. After data acquisition, the collection of processed data will be conducted. A Forensic examiner from law enforcement team will determine the data to be extracted and identified on to make the steganography content visible.

Furthermore, the restored data will be collected during forensic analysis processes during Phase 4 of the experimental test. In this step, the steganography content will be extracted so that forensic examiner could find out the critical evidence of the criminal activities such as what happened, who was involved, when does it happened, where does it happen, and how does it happened.

Finally, all the steps from experimental Phase 1 to Phase 4 of the investigation on audio steganography will be collected and recorded in a report. This report acts as an observation report that to guide others who will conduct a similar forensic investigation in future. It also can ensure that all the processes during the proposed experimental test are repeatable for similar results.

3.3.3 Data Treatment

Data treatment is considered as the preparation stage among the data requirement categories. All the collected data include pre-test results, original data, processed data, and extracted data will be processed in Windows Excel for comparative evaluation and storage purpose.

For the pre-test results, the data will be reviewed and stored in separated folders by generated from different tools. Hash value and comparison results will be recorded in Windows Excel in order to determine the suitable tools for conducting the case scenario test in Phase 2. The original data generated in case scenario will be processed using two PCs installed with a Windows 7 operating system. One PC will be used as a sender to send audio steganography files while the other PC will be the receiver to receive the audio steganography files. All the data transmission activities will be monitored using WireShark. WireShark will generate a report that will also be stored. Furthermore, all the activities include evidence creation, tools selection, and network communication will be recorded in order to compare with the restored data from Phase 4.

To ensure the integrity of digital evidences is the critical part of digital forensic investigation. Thus, all the extracted and restored data will be processed for MD5 hash value that stored and compared in Windows Excel. In Phase 3 and Phase 4, several tools consist of write blocker and EnCase will be used to analyze the evidence image. Several reports will be generated by those tools and stored for further evaluation.

The reports from the investigation will be the critical document for recommendation of the best practice of audio steganography forensic investigation. The observation steps will be also stored in a flow chart for simple interpretation.

3.3.4 Data analysis

In the proposed experimental research, data analysis is the core part among data requirement categories. As designed, there are four data analysis procedures during the investigation. They are the analysis of pre-test results, forensic analysis on evidence data, comparative analysis on original data and extracted data, and analysis on observation reports.

In pre-test results analysis, the possibility of successfully embedding steganography content into audio streams by three steganography tools will be examined. The possibility of detecting live audio steganography by using WireShark will also be evaluated. StegAlyzerAS and StegAlyzerSS will be used to examine the detection and extraction of audio steganography files in the hard drive. A comparison among different tools will be analyzed in Windows Excel. Consequently, analysis on pre-test results will provide a technical foundation for tests in the following Phases.

The second data analysis procedure is so called forensic analysis. The processed data and extracted data will be evaluated during forensic investigation. For the particular investigation, forensic investigation means to search for the hidden messages during the investigation. Therefore, file analysis will be involved. After analysis on extracted and restored data, evidence should be found and prepared in the form that could be presented to the court.

The third data analysis procedure is important for the proposed research which is the comparative analysis between original data and restored data. Because it is the way to determine how successful that the proposed digital forensic investigation in relate to audio steganography detection and extraction.

Finally, an analysis on observation reports will be conducted. The aim of this analysis is to generate and conclude a recommendation to forensic investigators when conducting similar investigation. A flow chart diagram will be created after the analysis.

3.4 LIMITATIONS

It is important that the limitations of the proposed research methodology are outlined and discussed. This is so the restrictions (for example transfer and biase) in the proposed research can be recognized. Therefore, the research results can be objectively evaluated for further development.

The first main limitation of the proposed research is the steganography tools and techniques that are used to implement the experimental test. There are over 50 steganography tools freely or commercially available using different embedding techniques that can create audio steganography files. In the proposed research, only three most famous tools have been chosen. They are all using the LSB embedding method and are free downloadable from the Internet. It is to expect that commercial audio steganography tools are functionally better than these three chosen tools. Furthermore, some experts hackers may create their own audio steganography tools which is more advanced then published tools from the Internet.

Secondly, the chosen steganography detection tools are both trial version and not the full versions. There are limitations on functions and performance using trial versions compared with the full version. This situation limits the scope of the test results. In general, the lack of detection tools on audio steganography in market could cause problems not only to the proposed research but also to the real life environment.

Thirdly, the proposed research is conducted in a laboratory environment. The live capturing test is conducted on local area network within controlled conditions. Therefore, the network condition in real life may be more complex than the experimental test. The effect of real life investigation may be different than the proposed research as the network condition may be different. Furthermore, live data capture is usually be done using wireless technology with many more interferences.

Lastly, every forensic tool has its own limitations and drawbacks. The experimental findings are based on the evaluation generated by the chosen tools such as EnCase, MD5, StegAlyzerAS, and StegAlyzerSS. Therefore, the results of the proposed research may only be transferable to the investigations which are involved with using same or similar tools.

3.5 CONCLUSION

Chapter 3 is focused on developing a feasible research methodology for testing audio steganography detection and extraction in a digital forensic investigation. Similar studies in related to the research area were reviewed in detail to learn the relevant methodologies. Tools and techniques had also been evaluated and tested from relevant studies in order to develop proper test environment for proposed research. Then the review of problem areas from Chapter 2 were conducted and used to develop the research questions as well as asserted hypotheses for each question. The research model consists of five phases was outlined to provide a logical progress of how the experimental test will be conducted. Those five phases were developed based on an experimental research methodology that can be observed systematically.

Chapter 3 has also presented a detailed research data map in Figure 3.8 in order to show the logic connection and links among research questions, sub questions, research model, data requirements, and hypotheses. A fully description of the case scenario was presented thus data collection, data treatment, and data analysis from the case scenario were also discussed in detail to understand what will be done during the experimental test as well as what outcomes are expected after the test. Lastly, the research limitations has also been outlined and discussed. Therefore, the research findings should be objectively evaluated. The following Chapter will report those research findings from the experimental test.

Chapter 4

RESEARCH FINDINGS

4.0 INTRODUCTION

Chapter 3 has developed a research methodology for conducting a forensic investigation for audio steganographic. The research questions and hypotheses were established after reviewing current issues and problems from relevant studies in Chapter 2 the literature reviews and Section 3.2.1 the summary of similar studies. The data requirements for the experimental testing were then presented followed by the discussion of limitations of the proposed research methodology.

Chapter 4 is a report of the findings from the testing phases designed in Chapter 3. Variations between the designed methodology in Chapter 3 and the actual experimental testing will be discussed in Section 4.1. Section 4.2 will report the findings according to data collection including pre-test result, original data collection in case scenario. Section 4.3 will present findings from data treatment process which is generating the forensic images from collected raw data. Section 4.4 will report the analysis results after forensic investigation on audio steganography.

4.1 VARIATIONS IN DATA REQUIREMENTS

In this section, a few variations to the original proposed research methodology in data requirements (Section 3.3) have been made. It is important to identify those variations as they may affect the outcomes from the research testing phases. Any changes and differences between the proposed methodology and the actual methodology used during the testing phase are presented and discussed in following section.

4.1.1 Case Scenario

There are some changes have been made to implement the case scenario during the experimental test. Firstly, a change is involved with the software chosen to perform live network transmission. Originally, the software designed to transmit data packet between two suspects' PC is called Tencent QQ which is instant messenger software. Tencent QQ is user friendly designed that can transmit several kinds of files include audio files through network. During the set up in laboratory however, the software was blocked by administrative control. Therefore, backup software was chosen to conduct the data transmission between two suspects' PC named IPMsg (IP Messenger). IPMsg is based on TCP/IP protocol which means it's capable to transmit data through internet and local area network. During the test, IPMsg successfully transmit audio file as well as audio steganography file from suspect Braga's PC to suspect Toretto's PC.

A second problem discovered is that the Google map image used as steganography content in the case is PNG format. Therefore, the steganography embedding tool is required to be capable of hiding a PNG image rather than JPEG image as planed into audio file. PNG image is regularly used in the Internet the same as JPEG images. Most steganography tools are capable of using PNG images as well as JPEG images. Therefore, using a PNG image as steganography content file will not change the outcome and performance of the experimental test.

The third change in the case scenario is using FTK imager to create a forensic evidence image for analysis. As indicated in Section 3.3.1, the creation of a forensic image will be processed using a write blocker. During the actual experiment, FTK imager is used as the suspect's system was running at the time when forensic team seizing the hard drive. FTK imager saved an image of the hard disk in segments and calculated the MD5 hash value to confirm the integrity of evidence data.

The final variation is introducing StegAlyzerAS in the case processing in order to find out the steganography information. As WireShark and EnCase analysis could not indicate audio steganography information in the suspect’s hard drive. StegAlyzerAS was used to scan the hard drive searching for potential steganography tools.

4.1.2 Data Collection

As indicated in Section 3.3.2, the data collection part involved collecting pre-test results, collecting extracted data, collecting restored data, and a step by step process report. The few changes made during the actual processing of test phases will be outlined.

In the pre-test phase, three tools were designed to create audio steganography files. Originally, Mp3Stego, Openpuff, and S-Tools were chosen. But in practice, Mp3Stego is found very limited in file format. For example, only a mp3 file with compression ratio of 11 to 1 can be used as cover file and only a text file can be used to contain hidden information. Furthermore, Mp3Stego is a command line environment which is not considered user friendly for unprofessional users. Therefore, another steganography tool capable of hiding information inside audio file was chosen namely Mp3Stegz. Mp3Stegz has GUI interface and capable of hiding both text and image files into mp3 file.

The chosen cover file for pre-test is also been change because the file size is strictly required during embedding process especially when embedding image file into the audio file. Therefore, a 10MB mp3 audio file and a 44.2MB wav file (converted from 10MB mp3 audio file) were used as cover file instead of 3MB mp3 audio file and 12MB wav file in original design. The embedded file was also changed from JPEG image to PNG image as the download Google map is in PNG format and the Google map will be used as the route for case scenario.

The same change occurred to the original data collection for case scenario case. The cover audio files for the case scenario were mp3 file and wav file randomly chosen from various sample audio files that could cover the secret information with using the chosen steganography tool.

4.1.3 Data Treatment

Data treatment is the preparation stage among the data requirement categories. There was one change that occurred during the data treatment process. During the actual experiment, the forensic hardware write blocker was not used during the data collection. Hardware write blocker namely TABLEAU T8 was originally proposed to be used during evidence acquisition for maintaining data integrity purpose. However, as time consuming and lack of resources, the evidence acquisition was performed on live system using FTK imager to create image of the suspect's PC. The evidence image was DD raw image and the data integrity was maintained by FTK imager software. The created evidence image was then collected in a portable hard drive. The later connection between the portable hard drive and the PC to conduct forensic analysis was through software called SAFE Block by ForensicSoft to maintain data integrity.

4.1.4 Data Analysis

There were also some variations made during data analysis during the actual experiment. In Section 3.3.4, Steganography detection tool StegAlyzerSS was planned to be use during analyzing the forensic evidence. But in pre-test, StegAlyzerSS was found incapable of detecting audio steganography. Therefore, StegAlyzerSS was not used in data analysis of the extracted evidence data.

The second changing was adding the Recover Folders analysis and IM Parser analysis using EnCase during the actual test in order to look for the trace of any deleted folders and instant messenger programs in suspect's PC. Recover Folders analysis would exam deleted folders as well as the recycle bin within the suspect's hard drive. Any deleted folders were recovered to analyze the content and signature. Because suspects in the real world tend to delete sensitive and key evidence after conducting a crime activity, the Recover Folders analysis is important for any type of forensic analysis. IM Parser analysis was for analyzing the trace of using instant messenger programs. The purpose of adding IM Parser analysis in the experiment was trying to find out evidence on IPMsg software as it is the tool to send and receive audio steganography between the suspects.

4.2 DATA COLLECTION

The aim of data collection in the proposed research is generating and recording the fundamental results from the experimental test. The findings of data collection will be separate into different categories which are pre-test findings and case scenario findings. The following section will present those findings in detail.

4.2.1 Phase One: Pre-test Findings

The pre-test was acted as a pilot test for the whole proposed experiment. The pre-test was processing on Windows 7 professional OS environment. It was designed to collect the results of the three chosen audio steganography tools (Mp3Stegz, Openpuff, S-Tools), as well as the two steganography detection tools (StegAlyzerAS and StegAlyzerSS). Also the results of using IPMsg and data capture tool WireShark were collected during the pre-test.

4.2.1.1 Findings on Steganography Tools

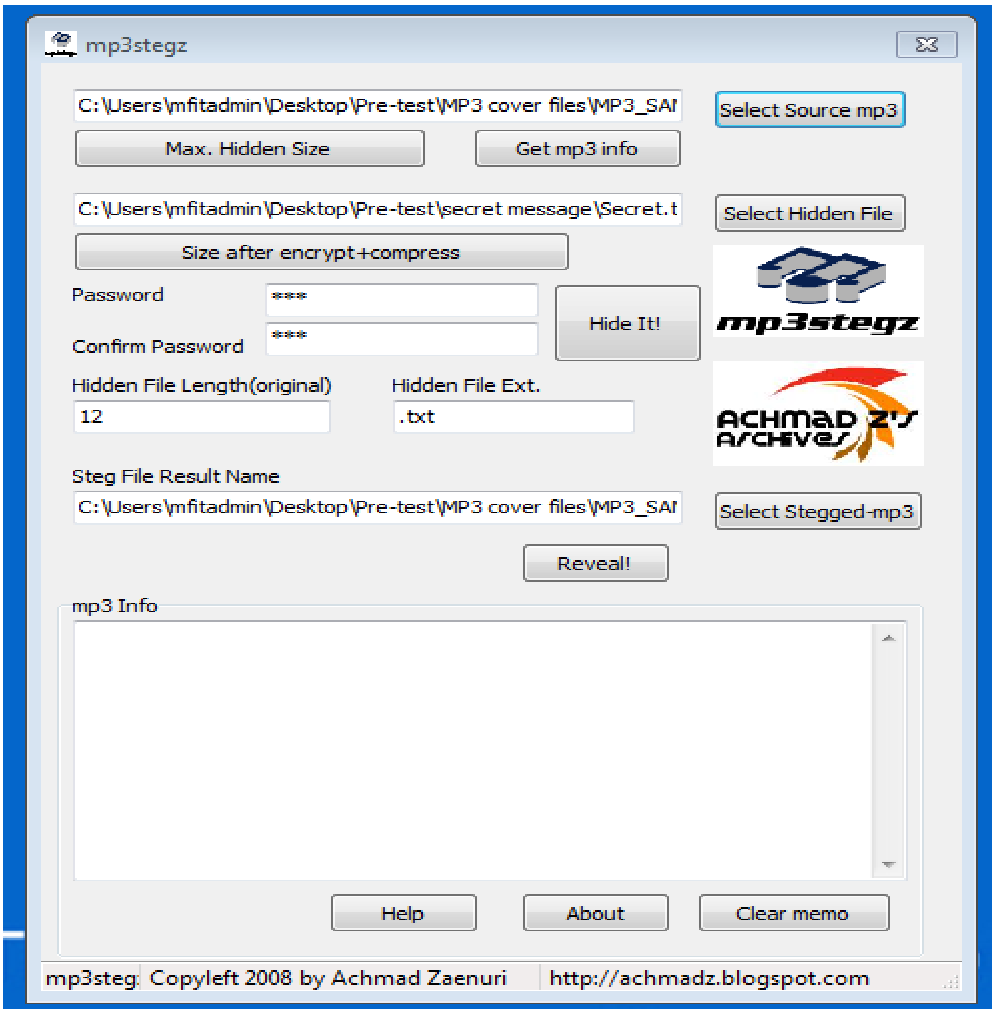
There were three audio steganography tools have been tested in pre-test phase. These tools which were Mp3Stegz, Openpuff, and S-Tools were mentioned in preceding section. The findings of using these tools are discussed in following section.

4.2.1.1.1 Mp3Stegz

The first tested steganography tool was called Mp3Stegz. Figure 4.1 shows the user interface of Mp3Stegz. It is a simple audio steganography tool that can hide various types of file into mp3 format audio file. Basically, to create an audio steganography file using Mp3Stegz is a three step process. First is selecting source mp3 file which is the cover or carrier file for audio steganography. Second step is selecting hidden file which is the steganography content or secret message to hide. The final step is clicking the hide-it button on the program to start the embedding process. Then the audio steganography file is made. Figure 4.2 shows successful using Mp3Stegz to create an audio steganography file named MP3_SAMPLE.MP3-steg.MP3 that carried a text file. Also, a secret text file and a secret image file were trying to embed into a sample mp3 file and a sample wav file using Mp3Stegz respectively.

Table 4.1: The hash value of cover files, secret files, and steganography files in Mp3Stegz test

Cover File	**Hash Value**
MP3_SAMPLE.MP3	3b5ee3763d5e5e5c6b88b5f03e7861dc
MP3_SAMPLE.MP3	3b5ee3763d5e5e5c6b88b5f03e7861dc
Stego File	**Hash Value**
MP3_SAMPLE.MP3-steg1.MP3	3a1068cd8ef5cf67386beefcac8b01e4
MP3_SAMPLE.MP3-steg2.MP3	9e16d64ab3231fc8eecc2b8704a1db9f



mp3stegz

C:\Users\mfitadmin\Desktop\Pre-test\MP3 cover files\MP3_SAI

C:\Users\mfitadmin\Desktop\Pre-test\secret message\Secret.t

Password ***

Confirm Password ***

Hidden File Length(original) 12

Hidden File Ext. .txt

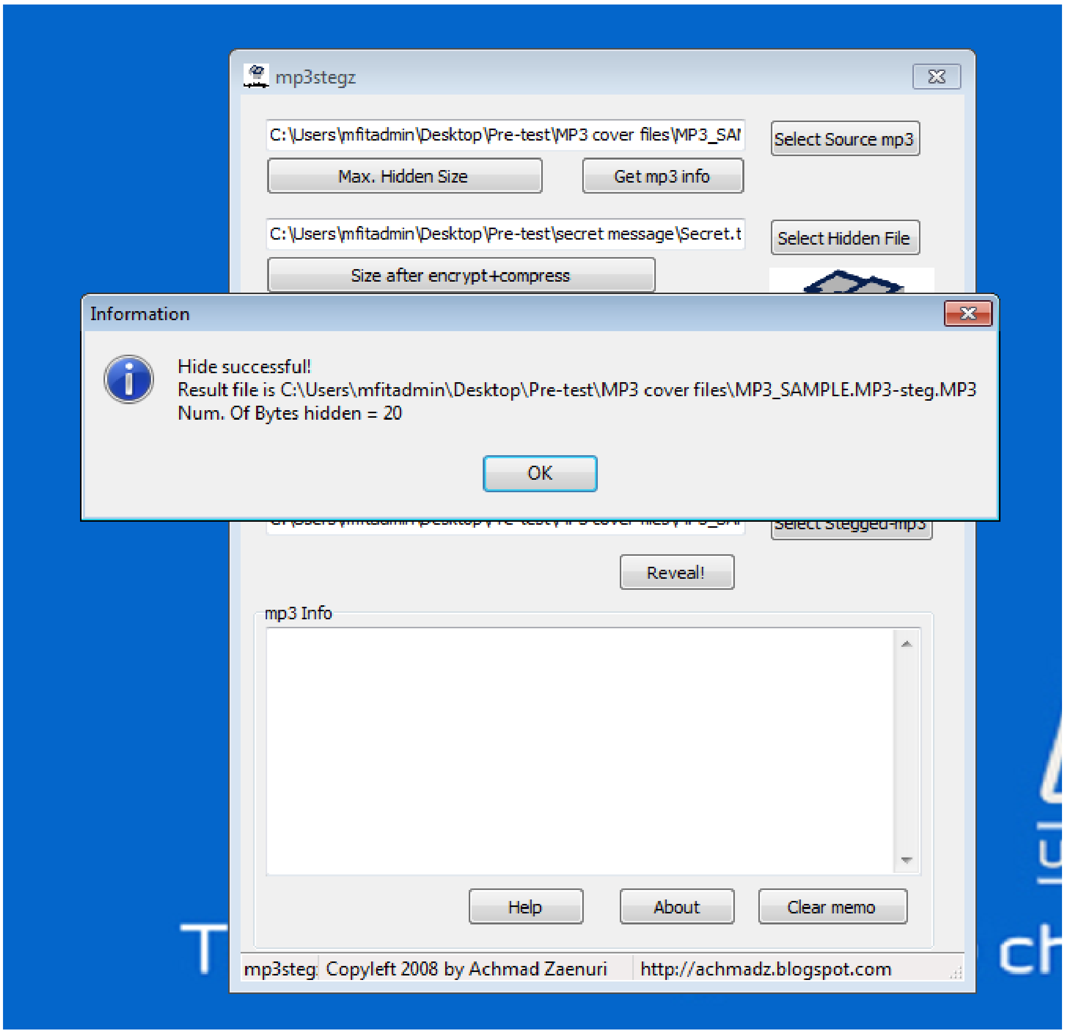
Steg File Result Name

C:\Users\mfitadmin\Desktop\Pre-test\MP3 cover files\MP3_SAI

mp3 Info

mp3steg: Copyleft 2008 by Achmad Zaenuri http://achmadz.blogspot.com

Figure 4.1: Interface of Mp3Stegz



C:\Users\mfitadmin\Desktop\Pre-test\MP3 cover files\MP3_SAI

C:\Users\mfitadmin\Desktop\Pre-test\secret message\Secret.t

Hide successful!
Result file is C:\Users\mfitadmin\Desktop\Pre-test\MP3 cover files\MP3_SAMPLE.MP3-steg.MP3
Num. Of Bytes hidden = 20

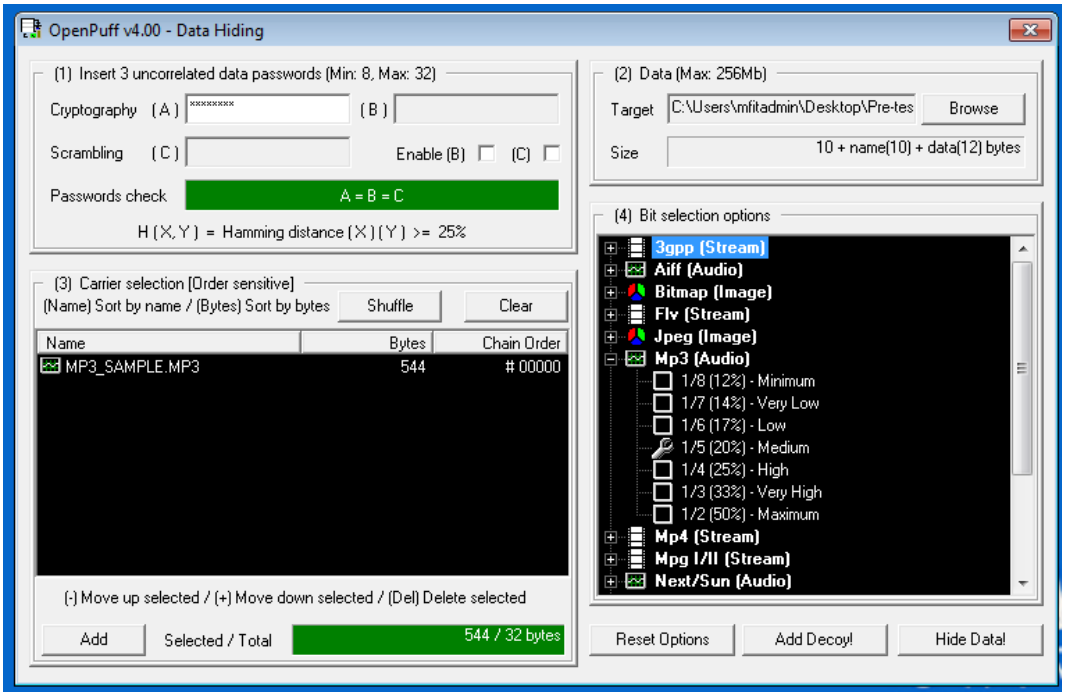
mp3steg: Copyleft 2008 by Achmad Zaenuri http://achmadz.blogspot.com

Figure 4.2: Successfully create audio steganography file using Mp3Stegz

The results showed that embedding secret into mp3 file was successful, and the steganography files were created in mp3 format. The created mp3 steganography files were named **MP3_SAMPLE.MP3-steg1.MP3** and **MP3_SAMPLE.MP3-steg2.MP3**. The hash values of the mp3 steganography which were calculated by WinMD5 were showed in Table 4.1. On the other hand, Mp3Stegz could not embed secrets into wav audio file because it does not recognize wav file.

4.2.1.1.2 Openpuff

The second audio steganography embedding tool that tested was Openpuff. Openpuff is a professional, portable, and stealth steganography tool that supports many types of carrier formats such as images (BMP, JPEG, PNG, TGA, PCX), audio (MP3, WAV, AIFF, NEXT/SUN), video (3GP, MP4, MPG, VOB), and flash-adobe (FLV, SWF, PDF). This advanced program has a more functional user interface than Mp3Stegz which is shown in Figure 4.3.



OpenPuff v4.00 - Data Hiding

(1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32)

Cryptography (A) ******** (B)

Scrambling (C) Enable (B) □ (C) □

Passwords check A = B = C

H (X, Y) = Hamming distance (X) (Y) >= 25%

(2) Data (Max: 256Mb)

Target C:\Users\mfitadmin\Desktop\Pre-tes

Size 10 + name(10) + data(12) bytes

(3) Carrier selection [Order sensitive]

(Name) Sort by name / (Bytes) Sort by bytes

Name	Bytes	Chain Order
MP3_SAMPLE.MP3	544	# 00000

(-) Move up selected / (+) Move down selected / (Del) Delete selected

Selected / Total 544 / 32 bytes

(4) Bit selection options

- 3gpp (Stream)
- Aiff (Audio)
- Bitmap (Image)
- Flv (Stream)
- Jpeg (Image)
- Mp3 (Audio)
  - □ 1/8 (12%) - Minimum
  - □ 1/7 (14%) - Very Low
  - □ 1/6 (17%) - Low
  - 1/5 (20%) - Medium
  - □ 1/4 (25%) - High
  - □ 1/3 (33%) - Very High
  - □ 1/2 (50%) - Maximum
- Mp4 (Stream)
- Mpg I/II (Stream)
- Next/Sun (Audio)

Figure 4.3: Interface of Openpuff

The process of using Openpuff to embed secret into a carrier file was very similar than using Mp3Stegz. The only difference was Openpuff could make user to choose bit selection level of the carrier file before the hide. As shown in Figure 4.3, there were seven bit selection levels that can be chosen for mp3 steganography. The bit selection levels were from 12% up to 50%, the bigger the bit percentage selected the easier it could be detected by steganalysis.

The embedding process was same as it was in Mp3Stegz test. The same secret text file and secret PNG file were embedded into same mp3 file and wav file respectively. The results from Openpuff testing indicated that creating audio steganography files were successful other than embedding PNG file into mp3 carrier because the size of mp3 carrier file used in test was too small to cover the PNG file. The created audio steganography file using Openpuff were named **MP3_SAMPLE.MP3-op1.MP3**, **WAV_SAMPLE.WAV-op1.WAV**, and

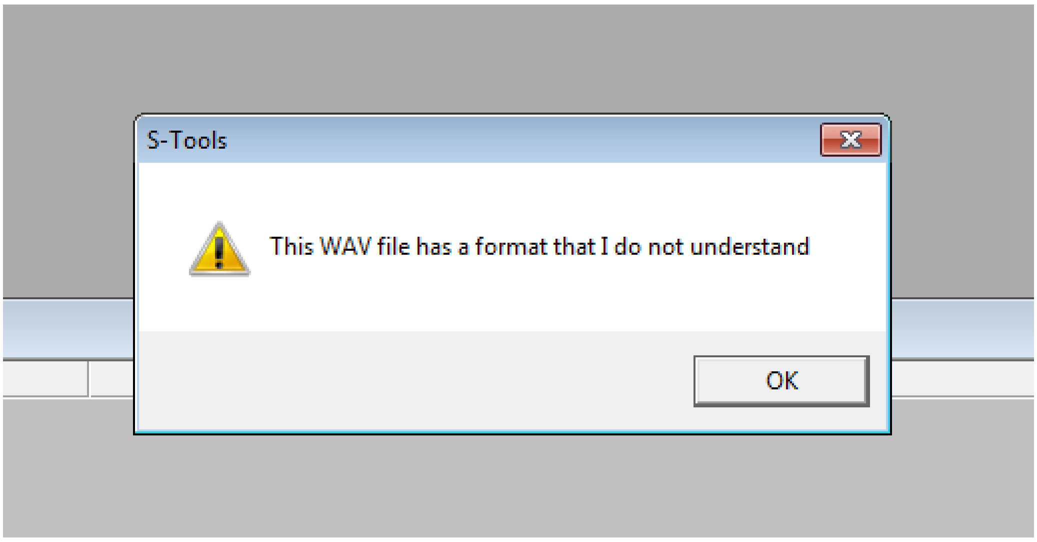
WAV_SAMPLE.WAV-op2.WAV. Hash value of each steganography files were also calculated by WinMD5. Table 4.2 shows the detail of the cover files, secret files, and audio steganography files in Openpuff test findings.

Table 4.2: Results from Openpuff test

Cover File	**Hash Value**
MP3_SAMPLE.MP3	3b5ee3763d5e5e5c6b88b5f03e7861dc
WAV_SAMPLE.WAV	5bf8accb3af895888141a716477a7200
WAV_SAMPLE.WAV	5bf8accb3af895888141a716477a7200
Secret File	**Hash Value**
Secret.txt	ed076287532e86365e841e92bfc50d8c
Secret.txt	ed076287532e86365e841e92bfc50d8c
Secret.PNG	0175bc3987450816d39e628e085892cc
Stego File	**Hash Value**
MP3_SAMPLE.MP3-op1.MP3	a86c76f7a94ee2175e51baf09470cc81
WAV_SAMPLE.WAV-op1.WAV	585cf58300d395b6135b03947319f97c
WAV_SAMPLE.WAV-op2.WAV	e46c7bc8246b8fa1725e5e1e57fe1b8c

4.2.1.1.3 S-Tools

S-Tool was the last steganography tool that was tested during pre-test phase. It is a drag and drop type of program which is user friendly designed. The designed carrier types are WAV audio file, BMP image file, and GIF image file. Other audio format files and image format files can be embedded into the designed carrier files. The embedding process was simply dragging the carrier file into the program window and then dragging the secret file onto the carrier file. The steganography file would then be created. However, during the pre-test, S-Tool was unable to identify both the mp3 carrier file as expected and wav carrier file as unexpected. Figure 4.4 shows the problem occurred during dropping wav carrier file into S-Tools. Therefore, S-Tools could not generate any audio steganography from prepared carrier files and secret files for collecting findings.


S-Tools

This WAV file has a format that I do not understand

Figure 4.4: Failure of using S-Tools to conduct audio steganography

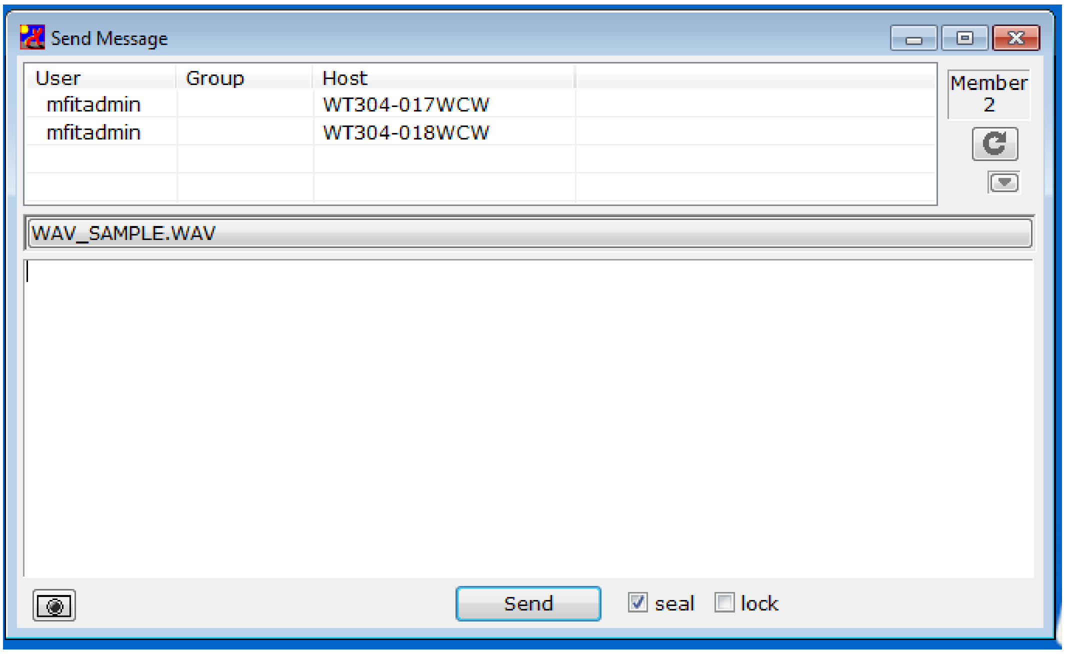
After testing on the three steganography embedding tools, the reliable findings could use to determine which tool is better for public to use in performing audio steganography. The comparative finding in Table 4.3 shows that Openpuff could both work with mp3 carrier and wav carrier while Mp3Stegz could only work with mp3 carrier. Furthermore, S-Tool was unable to embed secret file into mp3 and wav carrier during the test. Among the three audio steganography tools Openpuff has the highest success rate in creating audio steganography which is 75% while Mp3Stegz and S-Tools is 50% and 0%. Also, Openpuff uses lest time in embedding process which is 11.5 second in average while Mp3Stegz uses 28 second in average. According to the better performance among these three audio steganography tools, Openpuff was suggested to be involved in later experiment.

Table 4.3: Comparative results of steganography tools

Steganography Tools Embedding Results				
Audio Steganography Tools	**Steganography Embedding Tried**	**Audio Steganography Made**	**Success Rate**	**Average Time Taken**
Mp3Stegz	4	2	50%	28s
Openpuff	4	3	75%	11.5s
S-Tools	4	0	0%	0s

4.2.1.2 Findings on Audio Steganography Network Streams

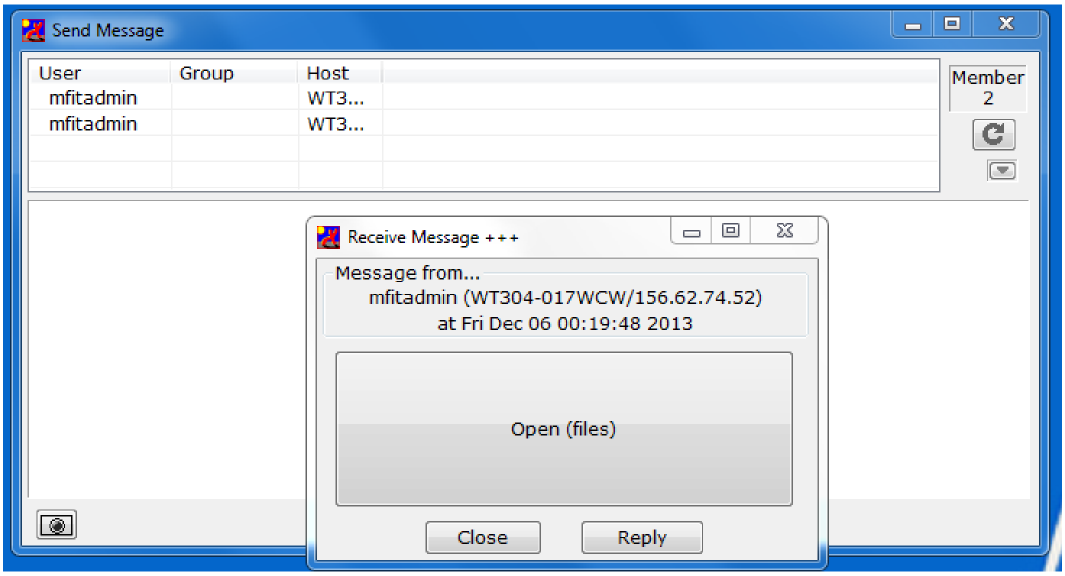
After testing the audio steganography embedding tools, three mp3 steganography and two wav steganography files had been stored in two folders separately.



User	Group	Host
mfitadmin		WT304-017WCW
mfitadmin		WT304-018WCW

WAV_SAMPLE.WAV

Figure 4.5: Send wav carrier file.



User	Group	Host
mfitadmin		WT3...
mfitadmin		WT3...

Message from...
mfitadmin (WT304-017WCW/156.62.74.52)
at Fri Dec 06 00:19:48 2013

Figure 4.6: Receive wav carrier file

Then the following test was sending one wav carrier file and one wav steganography file respectively from one PC to another PC using IPMsg. The data transmission test was conducted in MFIT laboratory in AUT. The sender's PC and the receiver's PC was both installed IPMsg program. By simply choosing the file and destination PC from the sender, the file was sent to the receiver in a stream of network packet automatically. Figure 4.5 and Figure 4.6 presents the sending and receiving file using IPMsg.

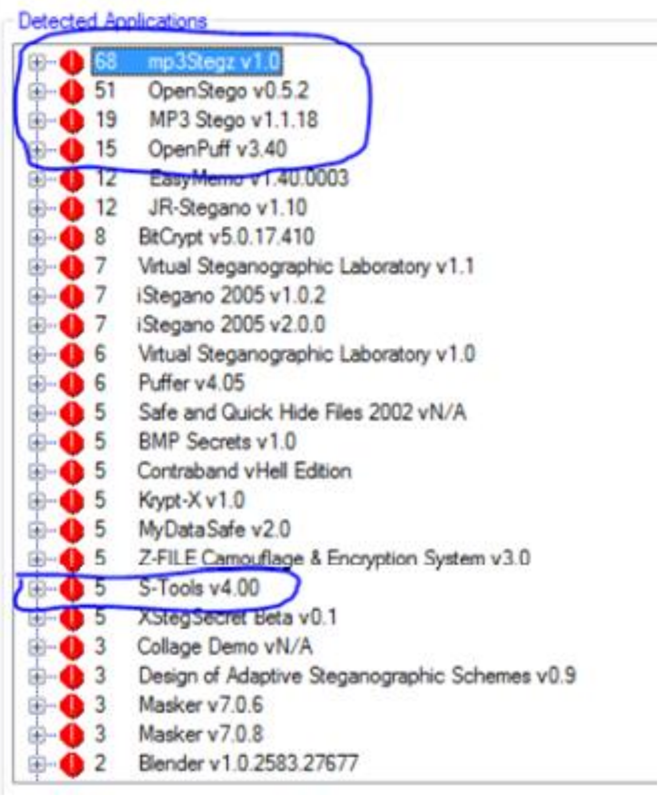
During the test, two data transmissions sent a wav carrier file and a wav steganography file one after another. Meanwhile, WireShark which installed in the sender's PC captured these two data streams. The two WireShark capture files were successfully collected were named **capture of cover file** and **capture of stego file**.

The analysis of the two WireShark capture files generated some critical findings which could affect the experiment's procedure. The analysis of findings will be discussed in Section 4.4 Data Analysis Findings in detail.

4.2.1.3 Findings on Audio Steganography Detection and Extraction

The detection of audio steganography is the most important part in the pre-test. Successfully detecting on audio steganography would achieve the first step in audio steganography forensic and audio steganalysis. During the test, two steganography detection tools were implemented and used. StegAlyzerAS (Steganography Analyzer Artifact Scanner) and StegAlyzerSS (Steganography Analyzer Signature Scanner) which are developed by the steganography analysis and research center with Backbone Security are the best steganography analysis tools around the world. StegAlyzerAS is developed to find digital steganography tools while StegAlyzerSS is developed to find digital steganography files. Those tools have GUI interface which are professional and user friendly.

Firstly, StegAlyzerAS was tested on the sender's PC intended to search and find installed steganography tools especially the three audio steganography tools (Mp3Stegz, Openpuff, and S-Tools). The result in Figure 4.7 showed that 13 steganography tools all found the three proposed audio steganography tools. The correct detection rate of StegAlyzerAS was 100% in the pre-test.

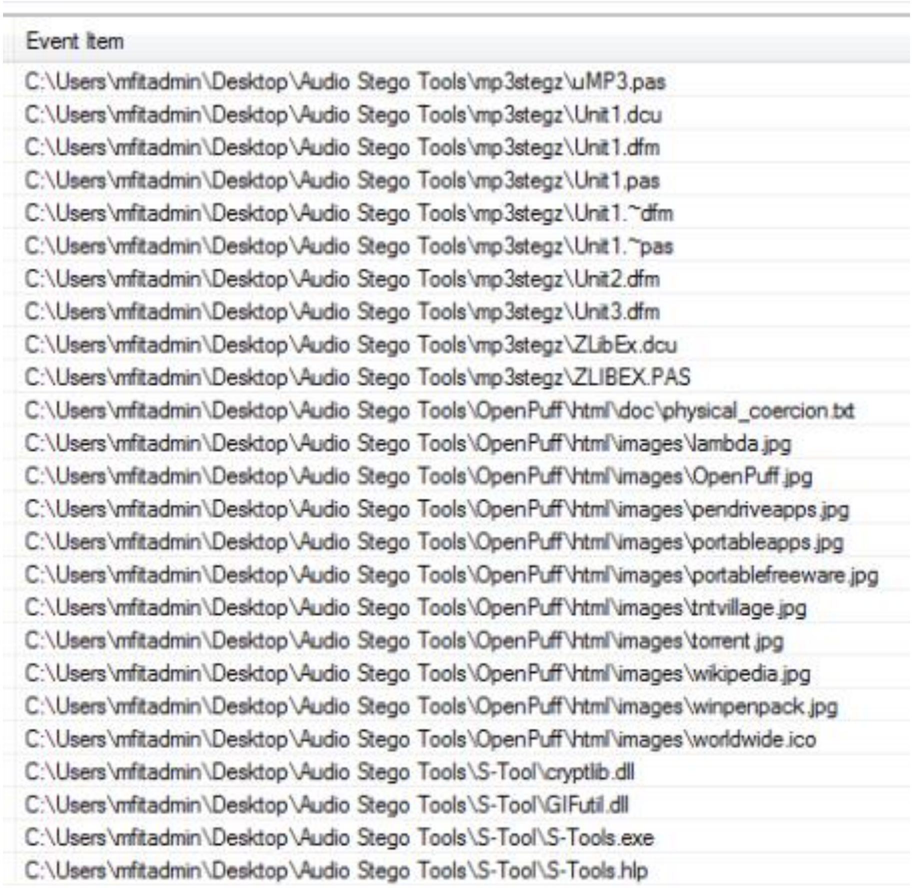


Detected Applications

68	mp3Stegz v1.0
51	OpenStego v0.5.2
19	MP3 Stego v1.1.18
15	OpenPuff v3.40
12	EasyMemo v1.40.0003
12	JR-Stegano v1.10
8	BitCrypt v5.0.17.410
7	Virtual Steganographic Laboratory v1.1
7	iStegano 2005 v1.0.2
7	iStegano 2005 v2.0.0
6	Virtual Steganographic Laboratory v1.0
6	Puffer v4.05
5	Safe and Quick Hide Files 2002 vN/A
5	BMP Secrets v1.0
5	Contraband vHell Edition
5	Krypt-X v1.0
5	MyDataSafe v2.0
5	Z-FILE Camouflage & Encryption System v3.0
5	S-Tools v4.00
5	XStegSecret Beta v0.1
3	Collage Demo vN/A
3	Design of Adaptive Steganographic Schemes v0.9
3	Masker v7.0.6
3	Masker v7.0.8
2	Blender v1.0.2583.27677

Figure 4.7: StegAlyzerAS detection result (1)

From Figure 4.8, not only steganography tools had been found, various types of files within the folder of detected steganography tools had also been found. However, from the case log of StegAlyzerAS we could tell that none of audio files were detected including audio steganography files stored in the sender’s PC. This result could determine that StegAlyzerAS is capable of detecting the steganography tools as well as audio steganography tools but it is unable to detect the audio steganography tools.

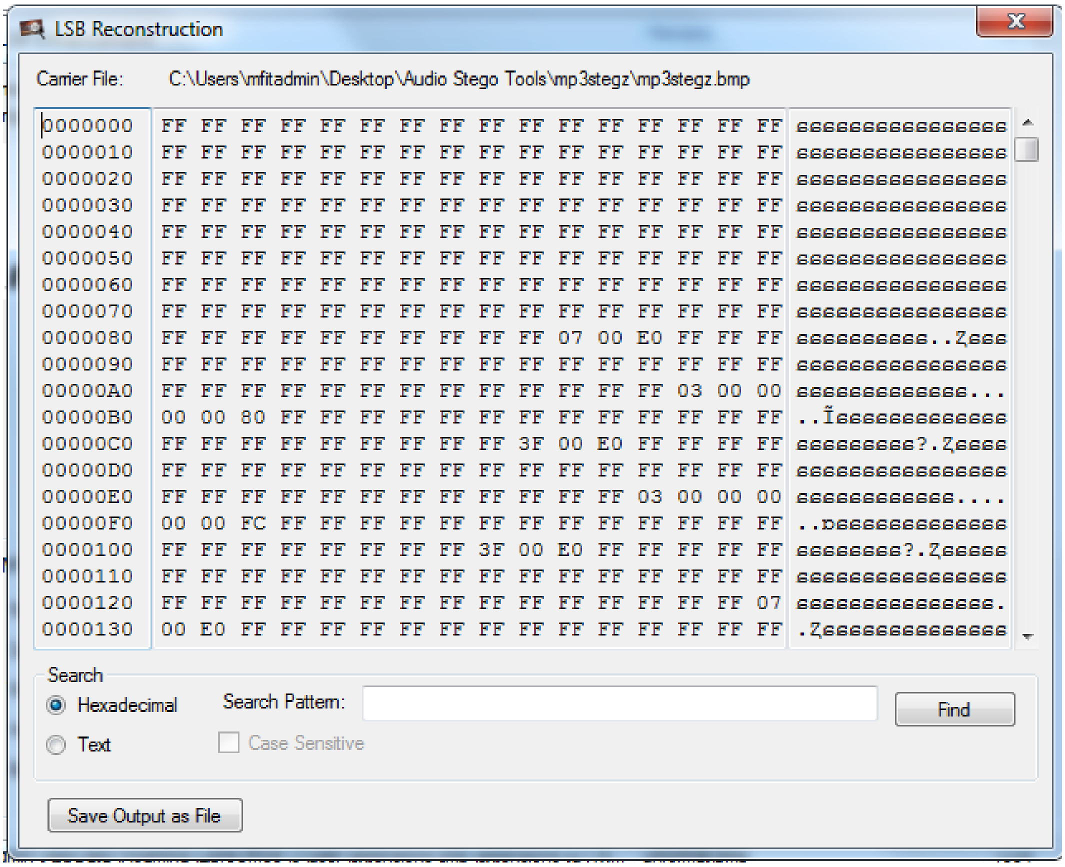


```
Event Item
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\uMP3.pas
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\Unit1.dcu
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\Unit1.dfm
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\Unit1.pas
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\Unit1.~dfm
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\Unit1.~pas
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\Unit2.dfm
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\Unit3.dfm
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\ZLibEx.dcu
C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\ZLIBEX.PAS
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\doc\physical_coercion.txt
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\lambda.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\OpenPuff.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\pendriveapps.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\portableapps.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\portablefreeware.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\tntvillage.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\torrent.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\wikipedia.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\winpenpack.jpg
C:\Users\mfitadmin\Desktop\Audio Stego Tools\OpenPuff\html\images\worldwide.ico
C:\Users\mfitadmin\Desktop\Audio Stego Tools\S-Tool\cryptlib.dll
C:\Users\mfitadmin\Desktop\Audio Stego Tools\S-Tool\GIFutil.dll
C:\Users\mfitadmin\Desktop\Audio Stego Tools\S-Tool\S-Tools.exe
C:\Users\mfitadmin\Desktop\Audio Stego Tools\S-Tool\S-Tools.hlp
```

Figure 4.8: StegAlyzerAS detection result (2)

The second steganography detection tool which has been tested was StegAlyzerSS. It was used to search the sender's PC during the test. StegAlyzerSS can apply three types of search which are signature search; append analysis; and LSB analysis by default function. During the pre-test, all three searches were applied in order to search for audio steganography files in the target PC. The results from StegAlyzerSS showed that all the image files in the target PC has been found and listed in the signature search; append analysis; and LSB analysis categories. Figure 4.9 displayed the steganography detection and steganalysis applied to an image file using StegAlyzerSS.

According to the StegAlyzerSS testing result, it can be found that this professional steganography detection tools can be only applied to find image steganography but not audio steganography. Therefore, StegAlyzerSS will not be used in the case experiment in next test phases.



LSB Reconstruction

Carrier File: C:\Users\mfitadmin\Desktop\Audio Stego Tools\mp3stegz\mp3stegz.bmp

```
0000000  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000010  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000020  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000030  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000040  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000050  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000060  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000070  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000080  FF FF FF FF FF FF FF FF FF FF 07 00 E0 FF FF FF  eeeeeeeeee..Ȥeee
0000090  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
00000A0  FF FF FF FF FF FF FF FF FF FF FF FF FF 03 00 00  eeeeeeeeeeeee...
00000B0  00 00 80 FF FF FF FF FF FF FF FF FF FF FF FF FF  ..Ĩeeeeeeeeeeeee
00000C0  FF FF FF FF FF FF FF FF FF 3F 00 E0 FF FF FF FF  eeeeeeeee?.Ȥeeee
00000D0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
00000E0  FF FF FF FF FF FF FF FF FF FF FF FF 03 00 00 00  eeeeeeeeeeee....
00000F0  00 00 FC FF FF FF FF FF FF FF FF FF FF FF FF FF  ..Deeeeeeeeeeeee
0000100  FF FF FF FF FF FF FF FF 3F 00 E0 FF FF FF FF FF  eeeeeeee?.Ȥeeeee
0000110  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  eeeeeeeeeeeeeeee
0000120  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 07  eeeeeeeeeeeeeee.
0000130  00 E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF  .Ȥeeeeeeeeeeeeee
```

Search

● Hexadecimal ○ Text

Search Pattern:

□ Case Sensitive

Figure 4.9: LSB analysis on image file using StegAlyzerSS

After testing on the two steganography detection tools, the findings showed that StegAlyzerAS is useful when searching for audio steganography tools on appointed hard drive area while StegAlyzerSS is not capable to search for audio steganography files. The following Table 4.4 presents the findings of the two steganography detection tools.

Table 4.4: Results from StegAlyzerAS and StegAlyzerSS

Forensic Tool	Detected Audio Steganography Artifacts	No. Applications Installed	No. Applications Detected
StegAlyzerAS	Unique File Artifacts	13	13
StegAlyzerSS – Signature	Signature Artifacts	5	0

Search			
StegAlyzerSS – Append Analysis	Appended Artifacts	5	0
StegAlyzerSS – LSB Analysis	LSB Artifacts	5	0

Since StegAlyzerAS can find all 13 steganography tools and StegAlyzerSS are not able to find any audio steganography files, but none of these two programs can extract steganography content from the audio carrier. Therefore, a proper way to extract audio steganography content was introduced and tested during the pre-test phase. During the first test on audio steganography embedding tools, it was found that Mp3Stegz, Openpuff, and S-Tools also have the extraction function. Thus, a performance test on the extraction function of those tools was conducted. During the embedding test, there were 5 audio steganography files created successfully. Those files were **MP3_SAMPLE.MP3-steg1.MP3** and **MP3_SAMPLE.MP3-steg2.MP3** created by Mp3Stegz; **MP3_SAMPLE.MP3-op1.MP3**, **WAV_SAMPLE.WAV-op1.WAV** and **WAV_SAMPLE.WAV-op2.WAV** created by Openpuff. The extraction test was using Mp3Stegz, Openpuff, and S-Tools to extract these 5 audio steganography files.

Table 4.5: Mp3Stegz extraction result

Extraction Tool	Steganography File	Extracted File	Extraction Succeed
Mp3Stegz	MP3_SAMPLE.MP3-steg1.MP3	Secret.txt	N
Mp3Stegz	MP3_SAMPLE.MP3-steg2.MP3	Secret.PNG	N
Mp3Stegz	MP3_SAMPLE.MP3-op1.MP3,	Not Applicable	
Mp3Stegz	WAV_SAMPLE.WAV-op1.WAV		
Mp3Stegz	WAV_SAMPLE.WAV-op2.WAV		

The extraction test using Mp3Stegz found that only mp3 format files could be recognized by the program. Also the extraction process with mp3 steganography files was turn out to be failed. The hash values of the secret files that have been embedded were **ed076287532e86365e841e92bfc50d8c** and **0175bc3987450816d39e628e085892cc**. But the hash values of the extracted files were changed to **d41d8cd98f00b204e9800998ecf8427e** and **969b37635d40e303396877957764b61f**. Furthermore, only steganography files that were created by Mp3Stegz can be extracted during the test. The result indicated that the secret files were modified during the embedding and extraction process using Mp3Stegz. Therefore, Mp3Stegz audio steganography extraction process was not successful as shown in Table 4.5.

The second extraction test was done by using Openpuff. The same as Mp3Stegz, Openpuff could only extract the audio steganography files that were created by itself. Among the 5 audio steganography files, there were 3 files created by Openpuff which were 1 mp3 file and 2 wav files. The findings from the test showed that the secret text file and secret image file can be extracted from the audio steganography files. The hash values were the same after extraction which determined the extraction process was succeeded. The tool Openpuff was reliable in both audio steganography embedding and extraction performance as shown in

Table 4.6: Openpuff extraction result

Extraction Tool	Steganography File	Extracted File	Extraction Succeed
Openpuff	MP3_SAMPLE.MP3-steg1.MP3	Not Applicable	
Openpuff	MP3_SAMPLE.MP3-steg2.MP3		
Openpuff	MP3_SAMPLE.MP3-op1.MP3,	Secret.txt	Y
Openpuff	WAV_SAMPLE.WAV-op1.WAV	Secret.PNG	Y
Openpuff	WAV_SAMPLE.WAV-op2.WAV	Secret.PNG	Y

The last extraction test was conducted on S-Tools. The result was as same as the embedding test with tool. S-Tools could not recognize the wav files using in the test. Also, is could not extract the steganography file which was not created by itself. Thus, S-Tools would not to be used in phase two the case scenario test.

4.2.2 Phase Two: Case Scenario Findings

The case scenario is about the message delivery between the drug lords Antonio Braga to his men Dominic Toretto using audio steganography. The objective of the case investigation is to capture, analyze, and extract the potential evidence message from suspect's computer.

4.2.2.1 Original Data Collection

The case was set up with two desktop PCs in the MFIT laboratory. One PC was assigned to Antonio Braga the drug lord and the other one was assigned to Dominic Toretto to receive data from Braga. The PC name for Braga was wt304-018wcw and the PC name for Toretto was wt304-017wcw. Both desktop PCs were Intel ® Core ™ i5-2400 CPU at 3.1GHz with 8GB RAM and 500GB hard drive. The Windows 7 Enterprise was installed as OS in both PCs. Network connections was using the laboratory network which was client-server architecture. The secret messages were three text files and one image file in PNG format which was a picture of map from google map. The secret messages then were embedded into two mp3 audio files and two wav audio files to create audio steganography files using Openpuff. These steganography files were **WAV1.WAV, MP5.MP3, Yellow Subrine.mp3, and .6EVERY BODY.wav**. Table 4.7 outlines the collected original stego data that generated in case scenario and Table 4.8 shows the MD5 hash values of every original data.

Table 4.7: Original stego data from case scenario

Steganography Tool	Secret Message	Steganography File	Embedding Succeed
Openpuff	route.PNG	WAV1.WAV	Y
Openpuff	address.txt	Yellow	Y

		Subrine.mp3	
Openpuff	address1.txt	MP5.MP3	Y
Openpuff	address2.txt	.6EVERY BODY.wav	Y

Table 4.8: Hash values of the original data

Secret Message	**MD5 Hash Value**
route.PNG	55f84b2302b8b7a7e042cdceabc9c1f6
address.txt	068443576b3744ffb887bafa1f43822c
address1.txt	f4d1a45e442ba7153f46ffb1352b21ad
address2.txt	3ba7ab1f8788e559f204ef973c4ebcc6
Audio Stego File	**MD5 Hash Value**
WAV1.WAV	d443363f78f3ff5ff472bd93b06ba609
Yellow Subrine.mp3	debea9ed45cd6f053102d52d25f22796
MP5.MP3	9b839ebf4448dd6d951870e8167e48d2
.6EVERY BODY.wav	4d8984c97c1bf395831811669663a766

After creating those audio steganography files that contain the evidence of the secret messages, they had been sent from suspect Braga's PC to suspect Toretto's PC using IPMsg. The transmission of the message was in the following order. First, the audio stego file **WAV1.WAV** and **Yellow Subrine.mp3** which contained the secret message file **route.PNG** and **address.txt** were sent to Toretto's PC. Second, the audio stego file **MP5.MP3** contained file **address1.txt** was sent. Finally, the audio stego file **.6EVERY BODY.wav** contained file **address2.txt** was sent.

Meanwhile, the three data transmissions between the two suspects were monitored and captured using WireShark by ICPO successfully. There were also three captured files called **1st capture.pcapng**, **2nd capture.pcapng**, and **3rd capture.pcapng** were collected and stored as original data for Phase two. The findings from analyzing the captured file will be present in Section 4.4 Data Analysis Findings.

4.2.3 Conclusion

In conclusion, Section 4.2 presents and introduces all the data that have been generated and need to be collected from pre-test and case scenario set up stage. The pre-test phase evaluated the embedding and extraction function of Mp3Stegz, Openpuff, and S-Tools for audio steganography. The findings indicate that Mp3Stegz could create audio steganography in mp3 format but it had a problem with extracting the embedded message from stego file. Furthermore, Openpuff was able to both create audio stego file in mp3 format and wav format and extract message from the stego file created by it successfully. On the other hand, S-Tools had problem to recognize the audio file using in the test which means it is very limited in the types of files to operate.

The pre-test also examined the transmission performance of IPMsg and network stream data capture performance of WireShark. The result shows that IPMsg could successfully transmit audio files between two desktop PCs while WireShark could successfully capture the transmissions. The capture files were collected and stored for further analysis.

In addition, StegAlyzerAS and StegAlyzerSS were used to detect any steganography tools and steganography files respectively. It turned out that StegAlyzerAS was able to detect all the audio steganography tools but StegAlyzerSS could not detect any audio stego files when there were five of them in the hard drive.

Finally, Openpuff was used to create audio steganography files in case scenario. Then the files have been transmitted from suspect’s PC to another suspect in three times. WireShark was used to capture all the activities. All the audio stego files and WireShark capture files were collected as the original data for Phase two the case scenario.

4.3 DATA TREATMENT FINDINGS

Data treatment was conducted during the third test phase in which the original data from the suspect's desktop PC was acquired and processed into forensic image file format for further analysis.

4.3.1 Phase Three: Data Acquisition and Processing Findings

As for the forensic analysis environment, the same desktop PC as in Section 4.2.2.1 was used in MFIT laboratory with same set up. The acquisition of the evidence from suspect's PC was performed with software write blocker named SAFE Block and FTK imager 2.9. The acquisition process was shown in Figure 4.10 and all the evidence images were verified with MD5 and SHA hash value and saved in EnCase evidence file format (.E01) into a portable 1TB hard drive.

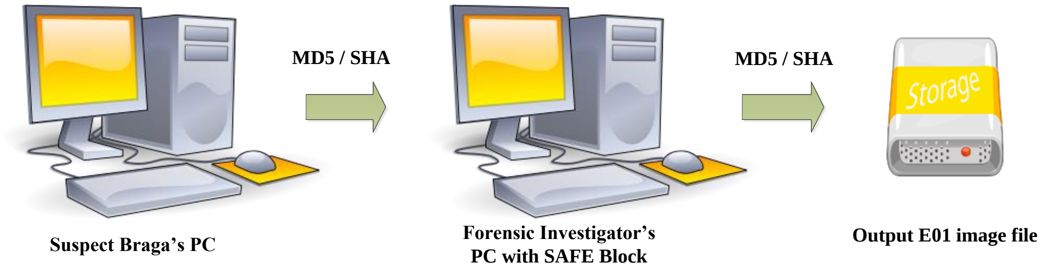


Figure 4.10: Data acquisition process

Acquisition is the critical process in digital forensic. Any mistake or improper action during the acquisition process will invalidate the evidence and may not be able to present to the court of law. Therefore, the acquisition process in this case scenario test were conducted strictly followed the general digital forensic procedure.

Firstly, the suspect's desktop PC was powered on when seized. The suspect's hard drive was taken off from the seized PC. The hard drive was Western Digital hard drive in 500GB with model WD5000AAKX-00ERMA0. The hard drive was connected to the investigator's PC using SAFE Block to ensure data integrity.

Then, FTK imager 2.9 was used to create the evidence image file. FTK imager imaged the suspect Braga's hard drive bit by bit and saved into a 1TB portable hard drive as Braga's harddrive.E01. The Braga's harddrive.E01 was the evidence file or image file that generated by data treatment in Phase three while it was the exact duplicate copy of suspect Braga's hard drive. Finally, the integrity of Braga's harddrive.E01 was verified with MD5 and SHA (Figure 4.14).

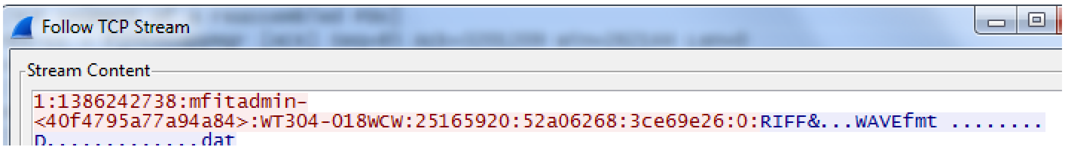
4.4 DATA ANALYSIS FINDINGS

Data analysis was the fourth phase of the designed research test. The purpose of data analysis was to find the audio steganography files that transmitted between suspects as well as any traces of audio steganography files from suspect's PC. Another purpose was to extract the steganography content from found audio steganography files. As designed in Section 3.3.4, there were four analysis procedures during the experiments. They were analysis on pre-test results which was the WireShark capture files, forensic analysis on evidence image file which was the Braga's harddrive.E01, comparative analysis on original data and extracted data, and analysis on observation reports.

4.4.1 WireShark Analysis Findings

During the pre-test and case scenario test, there were a total of five WireShark capture files collected. Those files were **capture of cover file.pcapng**, **capture of stego file.pcapng**, **1st capture.pcapng**, **2nd capture.pcapng**, and **3rd capture.pcapng**. The analysis of these capture files was using WireShark base on the guide from Chris Sanders's book *Practical Packet Analysis 2nd Edition – Using WireShark to Solve Real-World Network Problems* (Sanders, 2011). According to the book, the packet length in WireShark capture file can indicate a lot of information. For IP packet, there are usually 1460bytes for a packet to contain the transmittal data stream. Thus, only about 60% of packets from each capture file were containing the transmittal data and need to be analyzed.

In order to analyze those IP packets, another critical function of WireShark was involved. The WireShark gave investigators the opportunity to follow the TCP streams of every captured packet. Follow TCP Stream can make TCP streams easier to understand. It also sets the packets which have been transmitted between two machines into correct orders to read. The first WireShark Follow TCP Stream analysis was conducted to analyze two capture files from pre-test phase. For the first WireShark capture file **capture of cover file.pcapng**, the Follow TCP stream result showed that the file transmitted and captured in the stream was RIFF WAVE format. RIFF WAVE format file is wav audio file in Windows. Figure 4.11 shows the Follow TCP stream result of **capture of cover file.pcapng**.

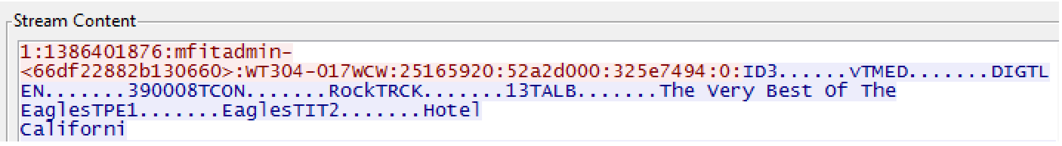


```
1:1386242738:mfitadmin-
<40f4795a77a94a84>:WT304-018WCW:25165920:52a06268:3ce69e26:0:RIFF&...WAVEfmt ........
D.............dat
```

Figure 4.11: Follow TCP Stream result for capture of cover file.pcapng

WireShark also found that the captured packets were sent to PC wt304-018wcw with IP address 156.62.74.34 and the source IP of the packets was 156.62.74.52. The key word search was applied to Follow TCP Stream result and it indicated that the file name which has been captured was unable to read. Then, the **capture of stego file.pcapng** was analyzed with Follow TCP Stream function. The findings showed that the transmitted file was also in wav format that sent to PC wt304-018wcw.

Base on the analysis experiment from pre-test capture files, the WireShark analysis of captured files from case scenario was conducted. The key function of the analysis was to follow the TCP Stream using WireShark. The result from **1st capture.pcapng** clearly presented that the destination PC was wt304-017wcw which was suspect Toretto's PC. Also, the result indicated that there was a wav format file had been transmitted. Furthermore there were few words that were readable rather than codes from the result. The words were “The Very Best of The Eagles Hotel California” as shown in figure 4.12.

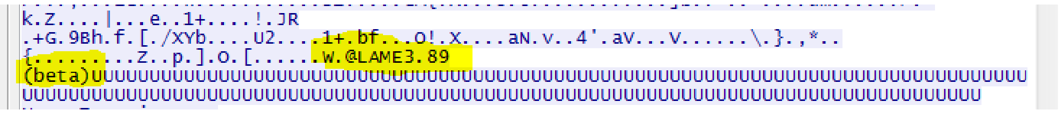


```
1:1386401876:mfitadmin-
<66df22882b130660>:WT304-017WCW:25165920:52a2d000:325e7494:0:ID3......VTMED.......DIGTL
EN.......390008TCON.......RockTRCK.......13TALB.......The Very Best Of The
EaglesTPE1.......EaglesTIT2.......Hotel
Californi
```

Figure 4.12: Follow TCP Stream result for 1st capture.pcapng

The "Hotel California" is a very famous song from bank called Eagles. This will also give a hint that the transmitted file could be an audio file and "Hotel California" can be used as keyword for search using EnCase later.

2nd capture.pcapng was also analyzed by Follow TCP Stream function. The findings indicated the packets were sent to PC wt304-017wcw as well. An important finding was that a keyword "LAME3.89 (beta)" was discovered. "LAME3.89 (beta)" is a kind of mp3 encoding method. Therefore, it determined that the captured packets contained an mp3 file. Figure 4.13 in below shows the finding from **2nd capture.pcapng**.



```
k.Z....|...e..1+....!.JR
.+G.9Bh.f.[./XYb....U2....1+.bf...O!.X....aN.v..4'.aV...V......\.}.,*..
{.........Z..p.].o.[......W.@LAME3.89
(beta)UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
```

Figure 4.13: Follow TCP Stream result for 2st capture.pcapng

Lastly, the **3rd capture.pcapng** was analyzed using Follow TCP Stream function. The result showed the packets were sent from PC wt304-018wcw to PC wt304-017wcw. The file transmitted was wav format file as same as the findings from pre-test WireShark analysis.

In conclusion, the WireShark analysis of the three capture results suggested that the transmitted files from Braga's PC to Toretto's PC were wav and mp3 format files. There were at least three audio files sent. One suspect audio file could be the song "Hotel California" from Eagles. Then, the knowledge level of steganography techniques will be tested.

4.4.2 StegAlyzerAS Analysis Findings

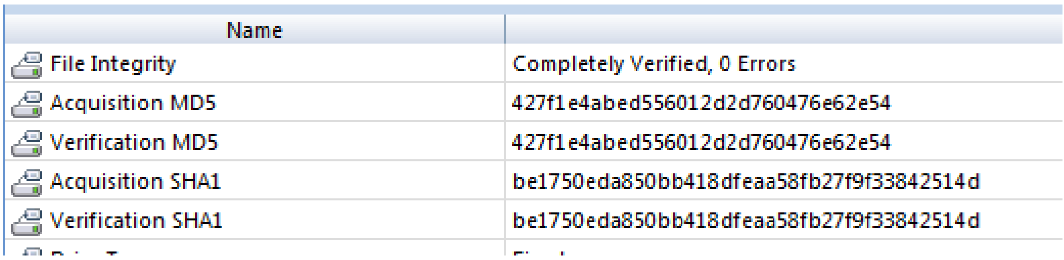
StegAlyzerAS was used after analyzing the WireShark capture files for evaluating the suspect’s knowledge level of steganography. Because StegAlyzerAS could find unique artifacts as well as steganography tools in target’s hard drive which had been proved in pre-test, it was also used to detect potential steganography tools from seized suspect Braga’s hard drive.

The process of StegAlyzerAS was the same as in pre-test phase. StegAlyzerAS was running to scan the suspect Braga’s hard drive when SAFE Block was used to guarantee data integrity. The scanning finished in 39 minutes with 171 applications detected and 688 unique file artifacts found (Appendix C). The finding of the StegAlyzerAS scanning showed audio steganography tool Openpuff was installed in suspect Braga’s hard drive with the path “D:\OpenPuff\OpenPuff.exe”. The finding could alert that audio steganography was highly possible involved in the case scenario.

Up to now, the WireShark findings and StegAlyzerAS findings suggested the potential evidence from the case scenario should be three or more audio files. Steganography might be involved because audio steganography tool Openpuff was found. Then, the major task of digital forensic analysis will be looking for the audio files using EnCase as well as extracting potential steganography content from the audio files.

4.4.3 EnCase Analysis Findings

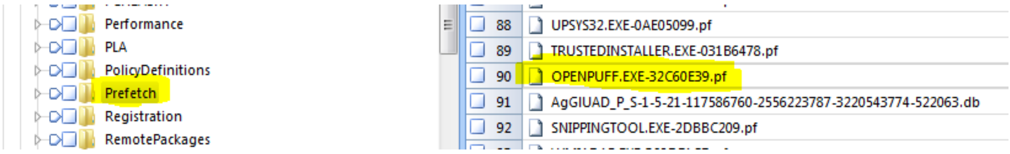
During EnCase analysis, the image files of suspect Braga’s hard drive was added into EnCase 7.0 on investigator’s PC. The hash value of every file in evidence images was verified. The verified MD5 and SHA1 hash values were same as the acquisition hash values thus the integrity of evidence image was guaranteed. Figure 4.14 shows the matched hash values from EnCase.



Name	
File Integrity	Completely Verified, 0 Errors
Acquisition MD5	427f1e4abed556012d2d760476e62e54
Verification MD5	427f1e4abed556012d2d760476e62e54
Acquisition SHA1	be1750eda850bb418dfeaa58fb27f9f33842514d
Verification SHA1	be1750eda850bb418dfeaa58fb27f9f33842514d

Figure 4.14: Hash value verified from EnCase

As Openpuff was found installed in suspect's PC, one necessary task in digital forensic investigation was to find out whether Openpuff had been executed or not. To achieve the task, the Windows prefetch file (*.pf) needed to be analyzed. Windows automatically keeps track of which programs were opened into small files that stored in prefetch folder. The benefit of this activity is to speed up the process in which the programs need to be executed next time. In another way, it helps digital forensic investigators to find out which programs were executed in suspect's PC. By using EnCase, the prefetch folder in Braga's PC was found and located under C drive. The path was "C:\Windows\Prefetch" for the folder. The prefetch file for Openpuff was found inside the folder named **OPENPUFF.EXE-32C60E39.pf**. Figure 4.15 shows the Openpuff prefetch file found by EnCase.

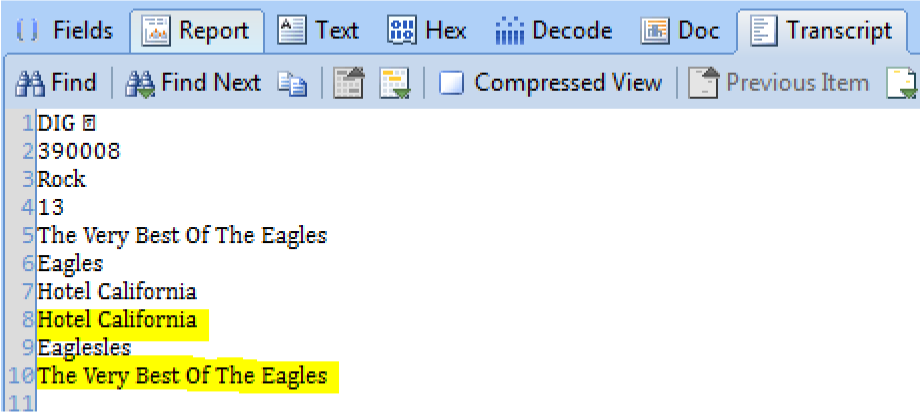


Performance
PLA
PolicyDefinitions
Prefetch
Registration
RemotePackages

88 UPSYS32.EXE-0AE05099.pf
89 TRUSTEDINSTALLER.EXE-031B6478.pf
90 OPENPUFF.EXE-32C60E39.pf
91 AgGIUAD_P_S-1-5-21-117586760-2556223787-3220543774-522063.db
92 SNIPPINGTOOL.EXE-2DBBC209.pf

Figure 4.15: Prefetch file of Openpuff found in EnCase

Program WinPrefetchView v1.5 was used to extract **OPENPUFF.EXE-32C60E39.pf** into readable format (Appendix C). The result showed that program Openpuff was originally installed in "D:\OpenPuff" folder in suspect's PC. Additionally, Openpuff has been executed for 11 times in suspect's PC. This finding further proved of steganography involvement in the case. The next task was trying to find out the steganography files in suspect Braga's hard drive.



```
DIG □
390008
Rock
13
The Very Best Of The Eagles
Eagles
Hotel California
Hotel California
Eaglesles
The Very Best Of The Eagles
```

Figure 4.16: Transcript of Yellow Subrine.mp3

As there was a hint message extracted from WireShark analysis which was "The Very Best of The Eagles Hotel California". A keyword search process by using EnCase was conducted. The finding pointed to a mp3 file named **Yellow Subrine.mp3** in folder "D:\OpenPuff\!!". Figure 4.16 shows the transcript of the file **Yellow Subrine.mp3** .

Moreover, there were three other audio files found in the "D:\OpenPuff\!!" folder named **WAV1.WAV**, **MP5.MP3**, and **.6EVERY BODY**.wav. As there was steganography suspicion, the audio file **Yellow Subrine.mp3** was tested using Openpuff in order to extract potential steganography content. However, the extraction was failed (Appendix C). From preceding WireShark analysis, it was known that Yellow Subrine.mp3 was sent from Braga to Toretto with another wav file. Therefore, the other two wav files which were in the same folder with **Yellow Subrine.mp3** file were tested with Openpuff in order to extract any steganography content. The result was positive. A PNG format image called **route.PNG** was extracted from **WAV1.WAV** audio file and a text file called **address2.txt** was extracted from **.6EVERY BODY.wav** audio file (Appendix C). According to the extraction result, the audio file **MP5.MP3** was extracted by Openpuff. A steganography file was found from **MP5.MP3** called **address1.txt**. Thus, the findings from WireShark analysis that two wav files, one mp3 files, and the hint world "Hotel California" were all found using EnCase. The audio steganography extraction process extracted three steganography contents from those audio files.

Additionally, a filter process using EnCase was conducted in order to filter off other wav and mp3 audio files in suspect Braga’s hard drive (Appendix C). The result showed totally 3072 wav and mp3 files were found. In order to find out any additional audio steganography content, those files were extracted using Openpuff manually because it was the only steganography tool had been discovered in suspect Braga’s hard drive. The finding indicated there were no additional audio steganography in the suspect’s hard drive.

The EnCase analysis was finished by this stage. Three audio steganography files were discovered and three stego contents were extracted successfully. Then, the comparative analysis between original data and extracted data is processed in the next section.

4.4.4 Comparative Analysis

The overall goal of this comparative analysis is to compare the original data in test phase 2 and extracted data from test phase 4 to determine the successful of audio steganography extraction. The original data created from test phase 2 is shown in Table 4.9 and Table 4.10. It contains four secret content files and four audio steganography files. The secret message contains a route map which was the route that suspect asked his man to went through for trading drugs. Each secret text file contains the date, time, and quantity of drugs to trade (Appendix B).

Table 4.9: Original secret files

Secret Message	MD5 Hash Value
rout.PNG	55f84b2302b8b7a7e042cdceabc9c1f6
address.txt	068443576b3744ffb887bafa1f43822c
address1.txt	f4d1a45e442ba7153f46ffb1352b21ad
address2.txt	3ba7ab1f8788e559f204ef973c4ebcc6

Table 4.10: Original audio steganography files

Stego Files	MD5 Hash Value
WAV1.WAV	d443363f78f3ff5ff472bd93b06ba609

Yellow Subrine.mp3	debea9ed45cd6f053102d52d25f22796
MP5.MP3	9b839ebf4448dd6d951870e8167e48d2
.6Every Body.wav	4d8984c97c1bf395831811669663a766

After a series of digital forensic analysis, all four audio steganography files were found from suspect's PC.

Table 4.11: Extracted secret files from stego carrier

Secret Message	MD5 Hash Value
rout.PNG	55f84b2302b8b7a7e042cdceabc9c1f6
address1.txt	f4d1a45e442ba7153f46ffb1352b21ad
address2.txt	3ba7ab1f8788e559f204ef973c4ebcc6

Among those stego files, three secret messages were extracted successfully while one secret message could not be extracted. The hash values of extracted secret messages were generated by WinMD5. The result proved that the extracted data were same as the original data (see Table 4.11). Thus, the evidence found from the digital forensic analysis could be able to be presented to a court of law.

In order to make a clear view of comparative analysis between the original data and extracted data, following table 4.12 is presented.

Table 4.12: Case scenario comparative analysis

Original Data	Extracted Data	
(Known Artifacts)	(Evidence)	(How)
Audio Steganography Tool - Openpuff	Found	Detected by StegAlyzerAS, execution found in Windows Prefetch file.
Audio Stego File - WAV1.WAV	Found	Captured by WireShark in streaming, discovered while browsing using EnCase.
Audio Stego File - Yellow Subrine.mp3	Suspected	Captured by WireShark in streaming, discovered while

		browsing using EnCase but fail to extract out stego content.
Audio Stego File - MP5.MP3	Found	Captured by WireShark in streaming, discovered while browsing using EnCase.
Audio Stego File - .6Every Body.wav	Found	Captured by WireShark in streaming, discovered while browsing using EnCase.
Secret Messages	Extracted	3 out of 4 secret messages have been extracted by Openpuff.

4.5 CONCLUSION

Chapter 4 has reported the findings discovered during the research testing phases. Variations to the original research design during actual testing were discussed in Section 4.1. A variety of audio steganography tools has been tested both in the embedding function and the extraction function. The results indicated that Openpuff was the most useful audio steganography tool both in embedding and extraction. Network traffic capture tool WireShark was also tested to capture audio stream packets when two PCs communication using IPMsg. The WireShark analysis function Follow TCP stream discovered the type of file and some hint messages from the captured packets.

The capability of the best automatic steganography detection and analysis tool on the current market was also tested. StegAlyzerAS was found to be capable of detecting all the famous steganography tools including audio steganography tools. On the other hand, StegAlyzerSS was able to detect image steganography files but unable to detect audio steganography files. Moreover, both StegAlyzerAS and StegAlyzerSS were incapable of extracting stego contents from steganography carrier files. The findings showed that the best steganography detection tools in the market nowadays were not helpful enough for a digital forensic investigation in audio steganography.

Additionally, the famous forensic tool EnCase was also tested. In has been found that only combined with other forensic tools such as WireShark, StegAlyzerAS, and WinPrefetchView can enCase located suspected audio steganography file. The steganography content could only be extracted using the same steganography tool when it has been embedded. Manual extraction processes were conducted on every audio file from the suspect’s hard drive in order to find out if any steganography content was present because there isn’t any auto scanning audio steganography detection tools currently available.

Chapter 5

RESEARCH DISCUSSION

5.0 INTRODUCTION

Chapter 4 reported the findings achieved from each phase of the experimental tests. The purpose of the proposed research methodology and several phases of testing were to investigate the current state of forensic tools related to an audio steganography investigation. Chapter 5 will now discuss those research findings from each phase of testing to answer the research questions and hypotheses proposed in Chapter 3. Furthermore, the findings will also be linked to the discussion to provide assurance on evaluating the research methodology, results, and conclusions.

To begin with, Section 5.1 will answer the research questions and test the asserted hypotheses respectively in tables. The argument made for and argument against each hypothesis will be generated from research findings. Section 5.2 will then discuss the research findings and explain the significant parts of the testing such as data treatment and comparative analysis in comparison with other research from literature and similar studies from Chapter 2 and Chapter 3. A brief recommendation will be made as well as a guideline in flow chart format for audio steganography investigation. Section 5.3 will conclude the whole chapter.

5.1 ANSWERING THE RESEARCH QUESTIONS

As designed in Chapter 3, the main research questions and the following sub questions were developed based on literature reviews from Chapter 2 and the similar studies from Section 3.2.1. The experimental findings from Section 4 will now be evaluated to answer the research questions in a table format in Section 5.1.1. Also the sub questions will be answered from the experimental findings in Section 5.1.2. Section 5.1.3 will then test the associated hypotheses of the research using both the theoretical knowledge from literatures and practical knowledge from tests. The tabular form will be presented both the arguments for and arguments against the hypotheses. Arguments for will be the findings that support, or prove the hypotheses while the arguments against will refute or disprove the associated hypotheses. A brief summary will be presented at the end of each argument table in order to accept, reject or be found as indeterminate based on the findings for each questions and hypotheses.

5.1.1 Answering Main Research Questions and Associated Hypotheses

There were two main research questions that have been developed based on similar studies for the proposed research. Two associated hypotheses were also presented according to the research question. Table 5.1 and Table 5.2 show the answers to the two main research questions and associated hypotheses respectively.

Table 5.1: main research question 1 and asserted hypothesis.

Main Question 1: What are the procedures and challenges when conducting digital forensic investigation for audio steganography?	
Asserted Main Hypothesis 1: The current state of digital forensic tools for steganography detection causes problems on digital forensic investigation for audio steganography.	
Argument For: There are four major embedding methods in audio steganography field (see Section 2.6.1). Many reliable and free audio steganography tools can be choose from internet such as Mp3Stegz, Openpuff, Mp3Stego, and so on. On the other hand, the digital forensic	**Argument Against:** Although the manually extraction of audio steganography was time consuming and frustrating, all the 4 audio steganography files were located by using combination of variety of digital forensic tools. Also 3 out of 4 stego contents were extracted successfully during the case scenario

tools have less focused on audio steganography. Automated steganography detection tool StegAlyzerAS and StegAlyzerSS could only find audio steganography artifacts but not stego content (Section 4.2.1.3 & Section 4.4.2). EnCase was also incapable to extract stego content from audio steganography (Section 4.4.3). The extraction of stego content was another key measurement for audio steganography investigation. There was a total of 19 extraction processes opportunities during the testing. 6 times the stego content was extracted successfully. Only the tool Openpuff was able to extract stego content successfully (Section 4.2.1.3 & Section 4.4.3).	testing (Section 4.4).
Summary: Audio steganography tools were successfully identified during pre-test phase and case scenario phase using StegAlyzerAS. Results from other common forensic tools such as WireShark, StegAlyzerSS, FTK, and EnCase were not as good as expected in audio steganography detection and analysis. The audio steganography contents were located and extracted manually during the case scenario testing. Another finding that only 6 out 19 audio stego contents were extracted successfully was also noticed. The augments made for and against prove that the main hypothesis 1 is to be accepted.	

Table 5.2: main research question 1 and asserted hypothesis.

<table>
<tr><td colspan="2"><b>Main Question 2:</b> How credible is the extracted content from audio steganography in relation to the forensic evidence purpose?</td></tr>
<tr><td colspan="2"><b>Asserted Main Hypothesis 2:</b> The detected audio steganography file can be extracted with the support from selected tools.</td></tr>
<tr><td><b>Argument For:</b>
Audio steganography tool Openpuff successfully extracted all three stego contents during pre-test phase (see Table 4.6). It also successfully extracted 3 out of 4 stego contents during case scenario (see Section 4.4.3).</td><td><b>Argument Against:</b>
During pre-test phase, audio steganography tool Mp3Stegz was incapable of extracting stego contents from audio steganography files created by it (see Table 4.5).
Furthermore, the result showed StegAlyzerAS and StegAlyzerSS were both incapable of extracting stego contents (see Section 4.2.1.3).
In addition, Openpuff extracted three stego contents during case scenario successfully, but there was still one stego content could not be extracted with uncertain reason (see Section 4.4.4 and Table 4.12).</td></tr>
<tr><td colspan="2"><b>Summary:</b>
There were 5 steganography extraction and detection tools tested during the experiment. Openpuff was found to be the most useful extraction tool among those selected tools. But the success rate of extraction was still 85%. Unfortunately the same feature for other selected tools was 0%. It means audio steganography extraction is a rough process and there could be less support</td></tr>
</table>

available for forensic investigator. The augments made for and against prove that the main hypothesis 1 is to be indeterminate.

5.1.2 Answering the Sub Research Questions

Sub research questions were designed linking to the main research question. The answer to the main research question was supported from the answers from sub questions. Following Table 5.3 to Table 5.6 was answer the four sub research questions in detail. (Note that because Mp3Stegz was used instead of Mp3Stego in actual testing, all Mp3stego mentioned in the sub questions is changed into Mp3Stegz.)

Table 5.3: Sub question 1 and answer

Sub Question 1: Does the network monitor and packet capture tool WireShark be able to capture potential audio steganography packets?
Answer: Yes
Summary: During pre-test, there were two audio streams transmitted while WireShark monitored and captured the streaming. The capture files were named **capture of cover file** and **capture of stego file** (see Section 4.2.1.2). One of the two captures was the audio steganography file in network streaming packet form. During the case scenario, all the three transmissions containing audio steganography files were successfully captured by WireShark as well (see Section 4.4.1).

Table 5.4: Sub question 2 and answer

Sub Question 2: Is it easy to embed various types of contents into audio streams with support from Mp3Stegz, Openpuff, and S-Tools?
Answer: Among these three tools, Openpuff was easy to embed text and image contents

into audio files. Mp3Stegz could only embed text content into mp3 format files. S-Tools failed to embed any contents into audio files.
Summary: The ability of embedding text content and image content was tested during pre-test phase. The three selected tools gave different outcomes. S-Tools could neither recognize mp3 file nor wav file during the testing. Mp3Stegz could only recognize mp3 file but not wav file. Also, Mp3Stegz embedded text content and image content into mp3 file successfully. The average time for Mp3Stegz to embed stego content into carrier was 28 seconds. Lastly, Openpuff was able to embed both text contents and image contents into wav file. But when it used mp3 file as carrier, only text content could be embedded. Image content was too big for mp3 carrier because the embedding algorithm of Openpuff is different from Mp3Stegz. The average for Openpuff to embed stego content was 11.5 second (see Section 4.2.1.1). The findings indicate it is lightly to embed stego content using Mp3Stez and Openpuff.

Table 5.5: Sub question 3 and answer

Sub Question 3: How efficient is StegAlyzerAS and StegAlyzerSS in order to identify audio steganography?
Answer: StegAlyzerAS was able to identify unique artifacts such as audio steganography tools on target hard drive while StegAlyzerSS was incapable to identify audio steganography files with signature, append, and LSB analysis.
Summary: In pre-test phase, the capability of StegAlyzerAS and StegAlyzerSS in related to audio steganography was tested respectively. StegAlyzerAS showed positive respond to audio steganography tools detection. There were 13 steganography tools installed and all of them were identified after StegAlyzerAS scanning. On the other hand, StegAlyzerSS could not identify any audio steganography files

and contents after signature analysis; append analysis; and LSB analysis. The result showed StegAlyzerSS did not process audio steganography (see Section 4.2.1.3).

Table 5.6: Sub question 4 and answer

Sub Question 4: Can StegAlyzerSS extract the steganography content that has embedded into audio streams? If it can't, can Mp3Stegz, Openpuff, and S-Tools extract the steganography content?
Answer: StegAlyzerSS can't extract stego content as well as Mp3Stegz and S-Tools. Openpuff is capable to extract stego content from audio steganography file created by it.
Summary: StegAlyzerSS was expected to extract stego content from audio steganography but it failed. S-Tool did not recognize the audio steganography files during the testing. Mp3Stegz extracted **Secret.txt** file from **MP3_SAMPLE.MP3-steg1.MP3** file but the content was changed after extraction. The same issue appeared in second extraction using Mp3Stegz when extracting **Secret.PNG** file from **MP3_SAMPLE.MP3-steg2.MP3**. Openpuff extracted achieved 85% success rate when extracting stego content from audio steganography during the testing. Then, Openpuff is the best tool to process stego content extraction from audio steganography (sees Section 4.2.1.3 and Section 4.4.4).

5.1.3 Testing Hypotheses

There were four associated hypotheses to be tested from Chapter 3. All of them were tested with the findings from Chapter 4. The form of hypotheses testing will be in tabular format with argument for and argument against same as Section 5.1.1.

Table 5.7 to 5.10 present the testing. (Note that because Mp3Stegz was used instead of Mp3Stego in actual testing, all Mp3stego mentioned in hypotheses is changed into Mp3Stegz.)

Table 5.7: Hypothesis 1 and testing

<table>
<tr><td colspan="2"><b>Hypothesis 1:</b> WireShark can capture network packets which have passed the network card; it cannot restore information of the packets for steganalysis purpose.</td></tr>
<tr><td colspan="2"><b>Tested Result:</b>
Accept</td></tr>
<tr><td><b>Argument For:</b>
WireShark was used in pre-test phase and case scenario phase. There were totally five capturing process during the testing. Each capturing process was successful.
After Follow TCP stream analysis on each captured file, the type of transmitted file was discovered. For example mp3 file or wav file (see Section 4.4.1). The source IP address and destination IP address were also discovered as well as the computer name.
But the captured streams were in</td><td><b>Argument Against:</b>
With the help from WireShark capture process and analysis, four audio steganography files were suspected to be present in the case scenario. Also to locate these audio steganography files will need the evdence found from WireShark analysis.</td></tr>
</table>

network packets form which means it is not readable. WireShark was also incapable to identify steganography content from captured packets.	
Summary: WireShark is the famous network capture and monitor tool for network administrative tasks. It fulfilled the primary task which was to capture the audio streaming during the proposed research testing. It also can perform several analyses on the captured file. One analysis called Follow TCP stream gave the hint of audio files were captured and a message “Hotel California” (see Section 4.4.1). Those hint helped to locate suspected audio steganography files in following testing. However, it has been found that WireShark could not directly identify or restore stego content. Therefore, the argument made for and argument against prove that the hypothesis is acceptable.	

Table 5.8: Hypothesis 2 and testing

Hypothesis 2: The chosen three steganography tools are quick and efficient for everyone to create their own audio steganography files.	
Tested Result: Indeterminate	
Argument For: The chosen three steganography tools were Mp3Stegz, Openpuff, and S-Tools. Mp3Stegz was able to embed text content and image content only into mp3 carrier in order to create audio steganography file. Average time for	**Argument Against:** The selected tool S-Tools was unable to create any audio steganography files because it could not recognize the wav and mp3 carrier file during the pre-test (see Section 4.2.1.1.3).

creating audio stego file using Mp3Stegz was 28 second (see Section 4.2.1.1.1 and Table 4.3). Findings of Openpuff testing showed it was capable to create audio steganography in mp3 and wav format quickly with average 11.5 second (see Section 4.2.1.1.2 and Table 4.3).	
Summary: Among the three selected steganography tools, two tools were found to be capable and efficient to create audio steganography files while the third tool S-Tools was unable to create audio steganography files during the testing. To consider this finding as well as the argument made for and argument against prove that the hypothesis is indeterminate.	

Table 5.9: Hypothesis 3 and testing

Hypothesis 3: Using StegAlyzerAS and StegAlyzerSS, over 90% of audio steganography tools and files can be detected and identified.	
Tested Result: Indeterminate	
Argument For: StegAlyzerAS was an advanced auto steganography detection tool. By scanning the target area, StegAlyzerAS was able to identify unique artifact such as steganography tools. During the testing, there were 13	**Argument Against:** StegAlyzerSS was reported to be effective when detecting image steganography files using signature analysis as well as LSB analysis algorithm. However, it was unable to identify any

<table>
<tr><td>different steganography tools installed in the hard drive. StegAlyzerAS was able to identify all the stego tools successfully (see Section 4.2.1.3). There was 100% detection on audio steganography tools using StegAlyzerAS.</td><td>audio steganography files during the testing (see Section 4.2.1.3) while there were certainly some audio steganography files stored in the hard drive. Even using LSB analysis, StegAlyzerSS could still not detect any audio steganography file. The detection rate was 0% on audio steganography files using StegAlyzerSS.</td></tr>
<tr><td colspan="2"><b>Summary:</b>
The performances of StegAlyzerAS and StegAlyzerSS were extreme. 100% of audio steganography tools were detected by StegAlyzerAS while 0% audio stego file detection for StegAlyzerSS. The argument for and argument against prove the hypothesis is indeterminate.</td></tr>
</table>

Table 5.10: Hypothesis 4 and testing

<table>
<tr><td colspan="2"><b>Hypothesis 4:</b> StegAlyzerSS is able to extract the steganography content from audio steganography files created by Mp3Stegz, Openpuff, and S-Tools.</td></tr>
<tr><td colspan="2"><b>Tested Result:</b>
Reject</td></tr>
<tr><td><b>Argument For:</b>
StegAlyzerSS was capable to process image steganography as reported.</td><td><b>Argument Against:</b>
Mp3Stegz and Openpuff created 9 audio stego files totally during the experimental test.

None of the audio stego files were found by StegAlyzerSS (see Table 4.4). Additionally, StegAlyzerSS has not got audio stego content extraction function</td></tr>
</table>

	found after testing.
Summary: StegAlyzerSS was programed in order to detect steganography files by signature scanning as well as LSB analysis. The proposed research testing on StegAlyzerSS expected to have audio steganography capability. However, it was found that StegAlyzerSS was neither capable of detecting audio stego files nor extracting audio stego contents. Therefore, the hypothesis is proved to be rejected.	

5.2 DISCUSSION

This section is focus on discussing and commenting on each testing phases of proposed research and the significant findings that have been found in these testing phases. Section 5.2.1 will briefly present the process of each testing phases as well as the reason for applying such testing phases to achieve the research goal. Section 5.2.2 will discuss on forensic acquisition process conducted with comparison with other available forensic acquisition process from literatures and in real world experience. Section 5.2.3 will then discuss the findings from analysis on audio steganography investigation. Lastly, recommendation will be made based on the testing procedures and results in relate to audio steganography investigation.

5.2.1 Discussion of Testing Phases

Research testing was separated into several linked phases each with specific goals. By together, they are designed to achieve the major goal of the proposed research. The discussion comprised of these testing phases will be conducted in order to explain and evaluate the significant findings from Chapter 4. Outcomes of interest discovered during the testing and research questions addressed by each testing phase will also be identified.

Phase one of the research testing was critical in testing the performance of all the selected tools in related to audio steganography. It was critical because the results came from phase one could be used to answer sub research question 2, 3 and 4. Three selected stego tools were tested for embedding secret content into audio files. It was discovered that Mp3Stegz and Openpuff were able to create audio steganography successfully while the S-Tool was unable to process audio steganography. Because efficiency and user friendly were considered during selecting audio stego tools, Mp3Stego (see Section 2.5.4) which is a powerful audio stego tool was replaced by Mp3Stegz. The reason was Mp3Stegz has GUI interface while Mp3Stego uses command line which considered for professional users. Among these three tools, Openpuff had best performance result thus it has been used in phase two to generate audio stego files as evidence for the case scenario. WireShark was used because it performed well in Leung & Chan's research (Leung & Chan, 2007) (see Section 3.1.5) in which WireShark was used to collect evidence from VoIP streams. Therefore, in order to collect evidence from audio streams in phase two WireShark was tested in the early stages. The same as Leung & Chan's result, WireShark had positive performance in capturing audio streams during the testing. Then automated stego detection tools StegAlyzerAS and StegAlyzerSS were tested. As many researchers were thinking to develop methods on finding steganography tools (see Section 2.9), the proposed research was challenging the idea by testing StegAlyzerAS because it is a tool to scan steganography artifacts (tools). The result indicated StegAlyzerAS was efficiently in identifying steganography tools includes audio steganography tools as well. It also proved that StegAlyzerAS should be used in analyzing audio steganography in phase four. StegAlyzerSS was tested as it could perform three different analyses in order to identify stego files. They are signature analysis; append analysis; and LSB analysis. Among these three analyses, LSB is the algorithm that has been wide used in audio steganography (see Section 2.6.1.1). Unfortunately, the result of StegAlyzerSS testing gave negative outcomes in which none of the three analyses could identify audio steganography files or contents. In addition, Openpuff was found to be the best tool for audio stego content extraction as it achieved 100% extraction rate in phase one testing.

Phase one tested and evaluated the performance of all the tools include audio steganography tools and forensic investigation tools. The phase two then used accepted tools to generate evidence for the case scenario as well as collecting the evidence. The evidence was 3 secret text contents embedded into 3 audio files and 1 secret image file embedded into 1 audio file by using Openpuff. Then they have been stored as original data which needed to be compared with extracted data in a later discussion. Those stego files have been transmitted from a suspect to his connections through a network traffic set up in the laboratory environment. It was similar to the way others approached research that "used WireShark to capture all incoming and outgoing packets" (Leung & Chan, 2007) (see Section 3.1.5), WireShark was used to capture audio streaming packets while evidence was transmitted.

Phase three started the digital forensic procedure in order to investigate suspect audio steganography activities processed in phase two. As announced by Erbacher et al. (Erbacher et al., 2009), "digital forensics is the science of collecting, discovering, and preserving digital data for use in court" (see Section 3.1.3). Phase three started the collecting process which was defined as the acquisition phase in the research. The suspect PC was seized with the power on and full data in the hard drive. FTK imager was used to image the hard drive. SAFE block was used to ensure the integrity of the acquired image. The image was stored in a blank portable hard drive and sent to be analyzed in a forensic investigator's PC.

After getting the evidence image, the discovering process began. This was defined as analysis and extraction and was performed in phase four. The objective was to identify suspected audio stego files and extract stego contents out of the found stego files. Forensic tools have been tested in the preceding phase one that was used combined with the digital forensic tool EnCase. This is the same concept with Abboud et al. (Abboud et al., 2010) "if the detection tools are used in conjunction with other tools, then it makes the lives of investigators much easier and gives them better chance of detecting suspicious data" (see Section 2.8). The combined use of WireShark, StegAlyzerAS, Openpuff, and EnCase has successfully located audio steganography evidence. The secret evidence contents were extracted in a success rate of 75%. Considering the probability of detection of LSB steganography in audio streams is 20% announced by Xu et al. (Xu et al., 2011) (see Section 3.1.2), 75% extraction rate shows audio steganography investigations are possible but are to be approached with preparation and caution.

Phase five gives recommendations to audio steganography investigation and to forensic investigators based on the procedure taken and outcomes from the testing phases. A flow chart diagram of a guideline is found in Section 5.2.4.

5.2.2 Discussion on Data Treatment

Data treatment included acquiring evidence data into a forensic image format in order for analysis using digital forensic tools. The key measurement of the acquisition process is to maintain the integrity of the evidence data. Because the literature reviewed has less information in forensic acquisition processes, the knowledge and technology used in the research for acquisition followed the instruction from postgraduate study notes on digital forensic procedures.

FTK imager tool was used to create a forensic image of the suspect’s PC. All data was imaged bit by bit and the integrity was ensured by using a software write blocker named SAFE block. Unlike normal forensic acquisition, the proposed research acquired evidence from a live computer which means the PC was powered on. Acquisition on a live computer could overwrite the timeline of some log files and system registration files. The Investigator needed to stay in the crime scene in order to make sure nobody touched the suspect PC and the acquired evidence hasn’t been changed. Because the evidence was acquired directly from live computer, the hardware write blocker such as tableau could not be applied. Therefore, software write block SAFE blocker was used. The diagram shows the acquisition process (see Figure 4.10). After acquisition, data integrity was checked by calculating the MD5 and SHA1 hash value. The results indicated that the acquisition was successful and the evidence image was reliable (see Figure 4.14).

5.2.3 Discussion on Comparative Analysis

Comparative Analysis was made in Chapter 4 which was the comparison between the original data and extracted data. The original data was the created audio steganography evidence included 3 secret text contents and 1 secret image file. The extracted data was 2 secret text contents and 1 secret image file that have been extracted from the discovered audio steganography files after digital forensic investigation. One secret text content failed to be extracted because a function error.

The reason for conducting the comparative analysis is looking for the possibility of successfully extracting stego content from audio steganography. The literature reviewed and similar studies reviewed were all focused on detecting the steganography. Geetha et al. (Geetha et al., 2006) presented their steganalyzer with detection rate from 80.7% to 85.4% of steganography for different algorithms (see Table 3.4). Erbacher et al. (Erbacher et al., 2009) announced 69.2% detection rate on OleDetection for steganography in Microsoft Office programs (see Section 3.1.4). Another author Munirajan et al. (Munirajan et al., 2004) used their stego-detection algorithm to achieve 85% positive stego detection rate. All the information indicated steganography can be detected. But only detecting is not enough for forensic investigation. In digital forensic, there has to be evidence to present in a court of law. And for steganography, the evidence must be the secret content which has been embedded within the steganography file. Therefore, the research was seeking for a positive extraction rate of using the selected tools. Additionally, the results of testing phase one showed that audio stego files created by Mp3Stegz was detected but failed to be extracted. This result also proved that the extracting of audio stego content is important for forensic investigation.

During the comparative analysis, there were 3 out of 4 secret contents extracted. The MD5 hash values between original secret contents and extracted secret contents were exactly the same. Thus, the extraction rate of the proposed forensic investigation is 75% and it is a helpful research finding for the audio steganography extraction area.

5.2.4 Recommendation on Audio Steganography Investigation

After investigating audio steganography during the experiment, the practical experience and theoretical studies are both supportable for making a guideline on audio steganography investigation for a forensic investigator. Therefore, a flow chart diagram outlines the best practice of audio stego investigation is presented in Figure 5.1.

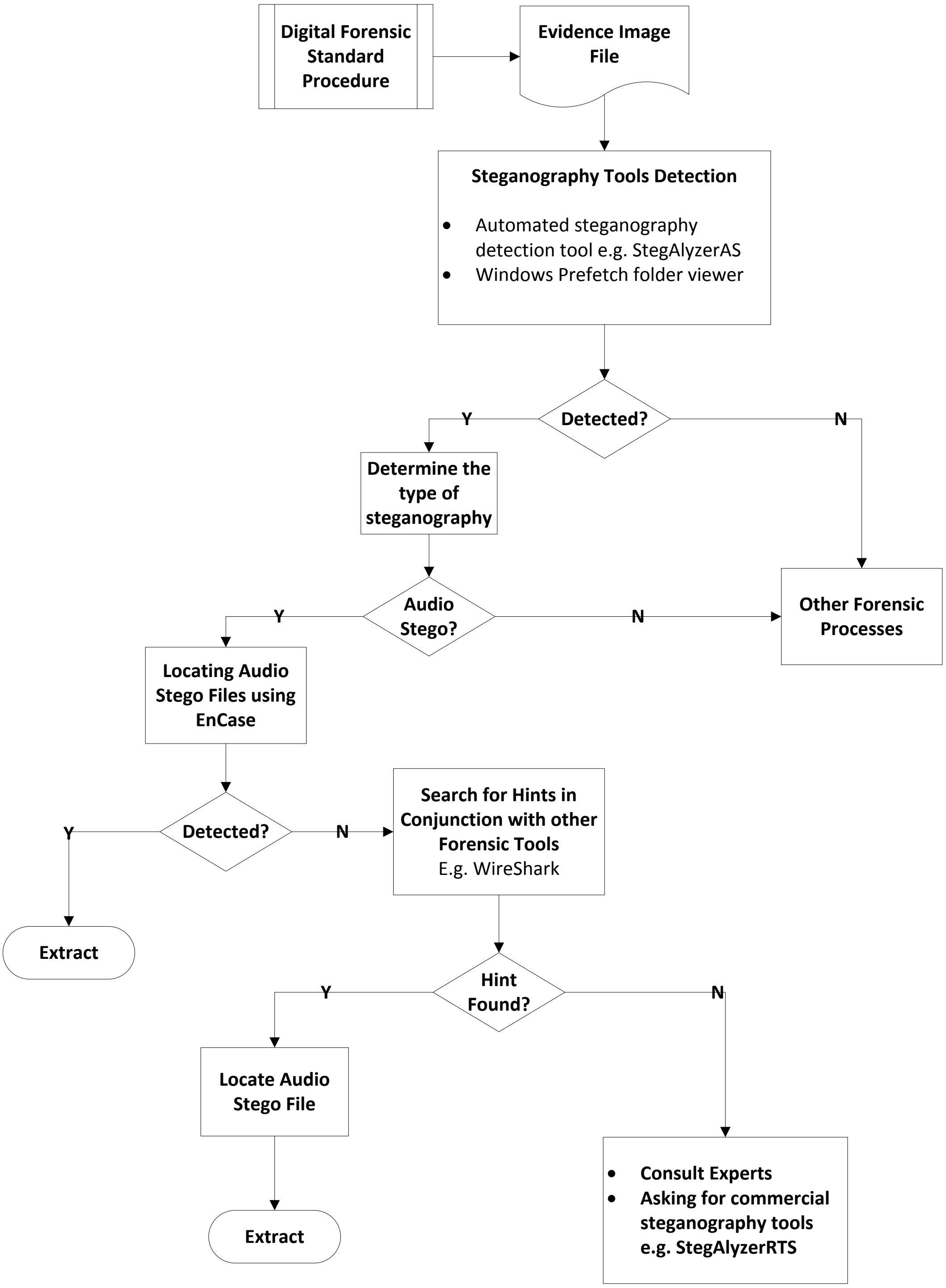

Digital Forensic Standard Procedure
Evidence Image File
Steganography Tools Detection
Automated steganography detection tool e.g. StegAlyzerAS
Windows Prefetch folder viewer
Detected?
Y
N
Determine the type of steganography
Other Forensic Processes
Audio Stego?
Y
N
Locating Audio Stego Files using EnCase
Detected?
Y
N
Search for Hints in Conjunction with other Forensic Tools E.g. WireShark
Extract
Hint Found?
Y
N
Locate Audio Stego File
Consult Experts
Asking for commercial steganography tools e.g. StegAlyzerRTS
Extract

Figure 5.1 and 1.2: Suggested audio steganography investigation flow chart diagram

Additionally, the current forensic tools cause problems to forensic investigators in audio steganography investigations. It has been found that none of the automated steganography detection tools worked accurately in detecting audio steganography files. The recommendation for forensic investigators is getting familiar with current audio steganography tools as well as audio stego algorithms. Also, the skill of using several steganography detection tools is required. Forensic investigators should also consult steganography experts in order to get advanced detection tools and technologies, because the selected steganalysis tools made significant difficulty during the proposed steganography investigation. Another recommendation would be working in conjunction with other forensic tools if the selected tool could not detect audio steganography. Other tools might not directly find out the audio stego information. But they could discover some important hints for investigation. The following of hints happened during the testing and made the investigation to be successful.

5.3 CONCLUSION

Chapter 5 has discussed the research findings discovered during testing phases that were reported in Chapter 4. The main research questions and asserted main hypotheses were answered and tested. Every sub research questions and asserted hypotheses were also answered and tested. Each testing phases were discussed with achievement, difficulties, and limitations.

The main objective of the research was to determine the challenges when conducting an audio steganography investigation. The research findings indicate that current forensic tools could cause problems in forensic investigations on audio steganography. On the other hand, more and more efficient audio steganography embedding tools available on the Internet make forensic investigation much harder. However, the research findings show positive signs to the investigator that using several forensic tools in conjunction can audio steganography content be extracted successfully. The extraction rate reaches 75% in the case scenario testing. Problems as well as a guideline were included in the discussion as reference for forensic investigators who are interested in a similar problem area.

The next Chapter 6 will make the conclusion to the entire thesis by reviewing the significant research findings. The limitations of the proposed research and recommendations for future research will also be report in Chapter 6.

Chapter 6

CONCLUSION

6.0 INTRODUCTION

Chapter 6 presents the final conclusion of the entire thesis as well as the research conducted. The experimental results from Chapter 4 will be reviewed and discussions from chapter 5 will be summarized. Then, the several limitations of the research will be outlined. Also, the potential future research within the chosen topic area will be discussed.

Audio steganography is the main concerned area of this research project. Steganography techniques have been used for hiding secret information for centuries. With the developing of computing and digital products, steganography was adopted and developed into different forms for the digital environment. Audio steganography is one of the forms that use digital audio files as carrier to hide the secret information. Steganography is a security enhancer when it was first developed. But the misuse of the hidden technology by criminals could raise the threats to security and system stability. This threat is realized by forensic investigators as secret evidence could be overlooked during forensic investigation. Moreover, the situation in audio steganography is even worse, because the current focusing is on image steganography and text steganography. Analysis on audio steganography will be interfered as digital audio files are already encoded and compressed. This is the reason that audio steganalysis tools could barely be found. Also, there is lack of investigation guidelines and procedure in audio steganography. Therefore, this research is not only seeking a method to detect audio steganography by using combination of forensic tools but also proposes a forensic guideline from the experience gained from the research testing in investigating audio steganography. In order to fulfill the objective, a research methodology was designed in Chapter 3 with theoretical knowledge from reviewed the literature in Chapter 2 and from five similar studies in Section 3.1.

The research was conducted using an experimental methodology that defined five testing phases to achieve the research goal. The testing phases were pre-test, case scenario, acquisition, analysis, and recommendation. Each testing phases had a specific goal. In testing phase 1, the designed goal was to evaluate the performance of different audio steganography embedding tools and detection tools as well as audio stream capture tool. Three audio steganography embedding tools include Mp3Stegz, Openpuff, and S-Tools were tested with audio steganography capability. The result indicated Mp3Stegz had 50% success rate in creating audio steganography using selected sources while Openpuff had 75% success rate and S-Tools had 0% success rate. Then, StegAlyzerAS and StegAlyzerSS were tested as they are the most advanced stego detection and analysis tools currently. The result of StegAlyzerAS was positive that 100% of 13 steganography tools were detected correctly while the result of StegAlyzerSS was negative with 0% detection rate of identifying audio stego files and contents. StegAlyzerSS used signature matching process; append analysis and LSB analysis which were techniques widely used in audio steganography. But the negative result of StegAlyzerSS showed there is big difference between image steganography and audio steganography in technical. WireShark was also tested to evaluate the capability of capturing audio stream packets from sender to receiver. The result showed a positive impact that WireShark accurately captured audio stream packets and stored in a capture file.

Testing Phase 2 was to generate a case scenario which simulated a crime activity using audio steganography. Basically, four audio files containing stego contents were sent from a suspect to his fellow criminals. WireShark was used to monitor the network traffic and capture audio streams into capture files. Once the simulation was completed, phase 3 and phase 4 were executed to acquire and analysis the evidence using tested digital forensic procedures. Not only does phase 3and phase 4 analyze the audio steganography activities in case scenario, but they also to discover the effective forensic procedures on audio steganography using the testing experience.

The experimental results from phase 3 and phase 4 showed that WireShark could analyze the captured audio streams by determining the format of the captured audio files. The results also showed that two wav files and one mp3 file was determined. Additionally, WireShark discovered a hint sentence which was “Hotel California” which implied a music file. StegAlyzerAS was also used to detect any potential steganography tools from suspect’s PC. Result showed that Openpuff was found. The windows prefetch fold then has been found using EnCase in analyzing the evidence image. A prefetch belongs to Openpuff was found and it showed that Openpuff was executed in suspect’s PC for 11 times. Therefore, audio steganography activates have been found and there were 4 to 11 audio steganography files needed to be discovered. A manually extraction on all wav and mp3 files from suspect’s PC were conducted and finally three secret contents were extracted. By conducting the comparison analysis between original data and extracted data, the extraction rate of 75% for audio steganography investigation during this research were reported.

Then, the main research questions were answered as well as each sub questions. It was found that the current digital forensic tools could cause problems in audio steganography investigation. Other than that, a discussion of the research findings was conducted. Recommendations included a flow chart diagram that suggested the forensic procedures for audio steganography investigation.

6.1 LIMITATIONS OF RESEARCH

The preceding Section 3.5 presented several limitations that came from the designed research. Also the actual research was conducted in a laboratory environment with limitations for transfer. These limitations will be discussed in this section.

From Section 3.5 discussions, the main limitation of the research is that the steganography tools and techniques used in implementing the testing were constructed in a laboratory. Actual testing experience showed the S-Tools were particularly sensitive. S-Tools reportedly to be a reliable steganography tools in image steganography and wav audio steganography. During the research, however it was not compatible with the wav carrier files selected. The problem could be the range of sample wav files was not big enough. There were types of wav files and S-Tools might only compatible with few types. If the sample size is big enough the selected audio steganography tools may perform differently.

Similarly, steganography detection tools StegAlyzerAS and StegAlyzerSS were both free trial version. The performance commercial version of StegAlyzerAS and StegAlyzerSS would be better. Not to mention StegAlyzerRTS (Steganography Analyzer Real-time Scanner) would have advantages on audio streaming steganalysis.

The steganography algorithm involved in this research was LSB algorithm. As mentioned in Chapter 2, the literatures reviewed indicated that LSB is one of the basic audio steganography algorithms. Therefore, the outcomes of this research could be different with outcomes from other audio steganography algorithms such as echo encoding algorithm and phase encoding algorithm.

Furthermore, the results of this research could also be limited by the way of conducting the forensic acquisition. During the testing phase 3, the suspect’s PC was acquired in live situation, the PC was not powered off and the hard drive was not taken. Also software write blocker was replaced hardware writer blocker which usually applied in real life case. Acquire evidence from live PC could have few changes on windows prefetch file. Additionally, the testing environment was set on Windows 7 Enterprise 64 bit operating system. The results were acceptably different if the environment changes to Windows 8, Linux or Mac OS.

Finally, the audio stream capturing using WireShark was conducted in wired network environment. Since wireless becomes more and more popular recently, the capture process in real life could affect by wireless performance. For example, wireless data could get more interference which causes errors on captured packets which can take effects on the research results. These matters all provide conditions for the transfer of results.

6.2 FUTURE RESEARCH

Audio steganography embedding, detection, and extraction were tested in this research with three steganography tools and two stego detection tools. For future research other steganography techniques and algorithms can be focused on. For example video steganography and visual steganography which relates to the frequency of a monitor screen can be tested. Also, future researchers can focus on testing other audio steganography algorithms such as phase encoding and echo encoding algorithms.

Secondly, the future research can set up another test-bed using Linux or Mac OS. The security elements such as firewall and intrusion detection system can be evaluated in testing with audio steganography or other steganography techniques. Also, the evidence image can be acquired from a pulled out hard drive from a suspect’s computer. Audio steganography on other platforms is suggested to be an area to research. For example, audio steganography can be applied on mobile phones, Xbox 360, PS3 and PS4.

Lastly, other automated steganography detection tools especially for audio steganography can be tested in future research. If conditions permit, the future research can be involved with some commercial steganography tools such as StegAlyzerRTS and StegoSuite from WetStone Technologies. Moreover, researchers with strong programming and mathematical skills could create a novel automated steganography detection tool in their research. The following logical flow chart of a hypothetical audio steganography detection tool is made in order to inspire future researchers.

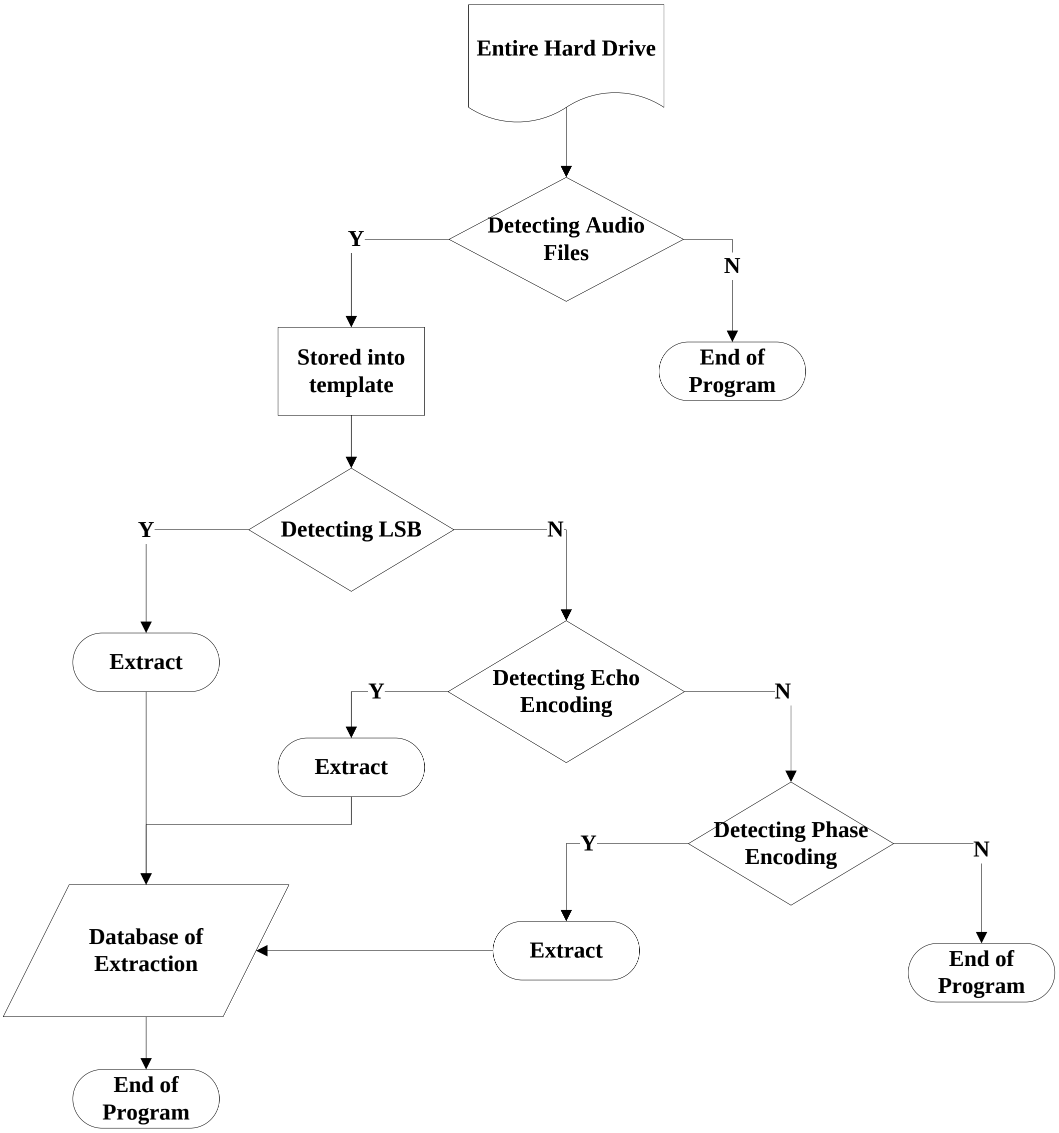


Figure 6.1: Logical diagram of potential audio stego detection tool

References

Abboud, G., Marean, J., & Yampolskiy, R. V. (2010). Steganography and Visual Cryptography in Computer Forensics. *Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE),* pp. 25-32.

Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal*, 313-336.

Britz, M. T. (2008). *Computer Forensics and Cyber Crime - An Introduction.* Pearson Higher Ed: USA.

Chen , M., Zhang , R., Niu , X., & Yang , Y. (2006). Analysis of Current Steganography Tools: Classifications & Features. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP '06.* pp. 384-387. Pasadena: IEEE.

Clarke, R. J. (2005). *Research Models and Methodologies*. University of Wollongong: New South Wales, Australia.

Dittmann, J., & Hesse, D. (2004). Network based intrusion detection to detect steganographic communication channels: on the example of audio data. *IEEE 6th Workshop on Multimedia Signal Processing,* pp. 343-346.

Erbacher, R. F., Daniels, J., & Mont, S. (2009). OleDetection Forensics and Anti-forensics of Steganography in OLE2-Formatted Documents. *Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE '09.* pp. 85-96. Berkeley: IEEE.

Erbacher, R. F., Daniels, J., & Montiero, S. (2009). OleDetection Forensics and Anti-forensics of Steganography in OLE2-Formatted Documents. *Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering. SADFE '09.* pp. 85-96.

Fridrich, J. (2010). *Steganography in Digital Media Principles, Algorithms, and Applications.* Cambridge University Press: UK.

Geetha, S., Sindhu, S. S., & Kannan, A. (2006). StegoBreaker: Audio Steganalysis using Ensemble Autonomous Multi-Agent and Genetic Algorithm. *India Conference, 2006 Annual IEEE* pp. 1-6. New Delhi: IEEE.

Geetha, S., Sindhu, S. S., Gobi, S., & Kannan, A. (2006). Evolving GA Classifiler for Audio Steganalysis based on Audio Quality Metrics. *Fourth International Conference on Intelligent Sensing and Information Processing. ICISIP 2006.* pp. 105-108. Bangalore: IEEE.

Ghazanfari, K., Ghaemmaghami, S., & Khosravi, S. R. (2011). LSB++: An improvement to LSB+ steganography. *TENCON 2011 - 2011 IEEE Region 10 Conference* pp. 364-368. Bali: IEEE.

Huang, Y., Tang, S., Bao, C., & Yip, Y. (2011). Steganalysis of compressed speech to detect covert voice over Internet protocol channels. *Information Security, IET 5(1)*, 26-32.

Jamil, T. (1999). Steganography the art of hiding information in plain sight. *IEEE Potentials*, 10-12.

Judge, J. C. (2001). *Steganography: Past, Present, Future.* SANS:USA.

Kipper, G. (2004). *Investigator's Guide to Steganography.* Florida: Auerbach.

Kumar, H., & Anuradha. (2012). Enhanced LSB technique for audio steganography. *Third International Conference on Computing Communication & Networking Technologies (ICCCNT),* pp. 1-4. Coimbatore: IEEE.

Leung , C.-M., & Chan, Y.-Y. (2007). Network Forensic on Encrypted Peer-to-Peer VoIP Traffics and the Detection, Blocking, and Prioritization of Skype Traffics. *16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. WETICE 2007.* pp. 401-408. Evry: IEEE.

Mitra , S., & Manoharan, S. (2009). Experiments with and Enhancements to Echo Hiding. *Fourth International Conference on Systems and Networks Communications. ICSNC '09.* pp. 119-124. Porto: IEEE.

Munirajan, V. K., Cole, E., & Ring, S. (2004). Transform Domain Steganography Detection using Fuzzy Inference Systems. *Proceedings. IEEE Sixth International Symposium on Multimedia Software Engineering.* pp. 286-291.

Niu, C., Sun, X., Qin, J., & Xia, Z. (2009, 08 08). Steganalysis of two least significant bits embedding based on least square method. *ISECS International Colloquium on Computing, Communication, Control, and Management. CCCM 2009. Volume:3* , pp. 124-127.

Nugraha, R. M. (2011). Implementation of Direct Sequence Spread Spectrum Steganography on Audio Data. *International Conference on Electrical Engineering and Informatics (ICEEI),* pp. 1-6. Bandung: IEEE.

Nutzinger, M., & Wurzer, J. (2011). A Novel Phase Coding Technique for Steganography in Auditive Media. *Sixth International Conference on Availability, Reliability and Security (ARES),* pp. 91-98. Vienna: IEEE.

Petitcolas , F. (2012, 21 10). *mp3stego*. Retrieved 08 16, 2013, from Fabien Petitcolas: http://www.petitcolas.net/fabien/steganography/mp3stego/index.html

Poisel, R., & Tjoa, S. (2011). Forensics Investigations of Multimedia Data: A Review of the State-of-the-Art. *Sixth International Conference on IT Security Incident Management and IT Forensics (IMF),* pp. 48-61. Stuttgart: IEEE.

Radhakrishnan, R., Kharrazi, M., & Memon, N. (2005). Data Masking A New Approach for Steganography. *Journal of VLSI Signal Processing*, pp. 293-303.

Sanders, C. (2011). *Practical Packet Analysis 2nd Edition - Using WireShark to Solve Real-World Network Problems.* San Francisco: William Pollock.

Sarreshtedari, S., Ghotbi, M., & Ghaemmaghami, S. (2009). One-third probability embedding: Less detectable LSB steganography. *IEEE International Conference on Multimedia and Expo, ICME* pp. 1002-1005. New York: IEEE.

Sathyal, V., Balasuhramaniyam, K., Murali, N., Rajakumaran, M., & Vigneswari. (2012). Data Hiding in Audio Signal, Video Signal Text and JPEG Images. *2012 International Conference on Advances in Engineering, Science and Management (ICAESM),* pp. 741-746. Nagapattinam, Tamil Nadu: IEEE.

Wei, Y., Guo, L., & Wang, Y. (2010, 10 18). Controlling Bitrate Steganography on AAC Audio. *3rd International Congress on Image and Signal Processing (CISP)*, pp. 4373-4375.

Westfeld, A. (2001). *F5—A Steganographic Algorithm High Capacity Despite Better Steganalysis.* Dresden: Technische Universit ̈at Dresden.

Xu, E., Liu, B., Xu, L., Wei, Z., Zhao, B., & Su, J. (2011). Adaptive VoIP Steganography for Information Hiding within Network Audio Streams. *14th International Conference on Network-Based Information Systems*

(NBiS), pp. 612-617. Tirana: IEEE.

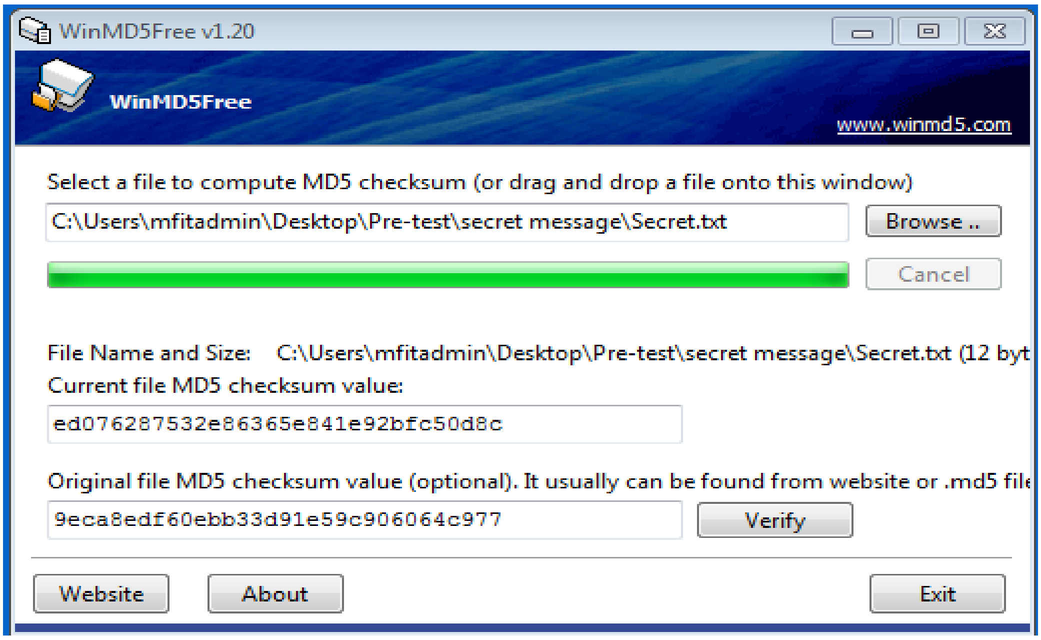
Zheng, Y., Liu, F., Luo, X., & Yang, C. (2012). A Method Based on Feature Matching to Identify steganography software. *2012 Fourth International Conference on Multimedia Information Networking and Security (MINES),* pp. 989-994. Nanjing: IEEE.

APPENDICES

APPENDIX A:

Phase One: Pre-testing

WinMD5 hashing process and result



WinMD5Free v1.20

WinMD5Free

www.winmd5.com

Select a file to compute MD5 checksum (or drag and drop a file onto this window)

C:\Users\mfitadmin\Desktop\Pre-test\secret message\Secret.txt

File Name and Size: C:\Users\mfitadmin\Desktop\Pre-test\secret message\Secret.txt (12 byt

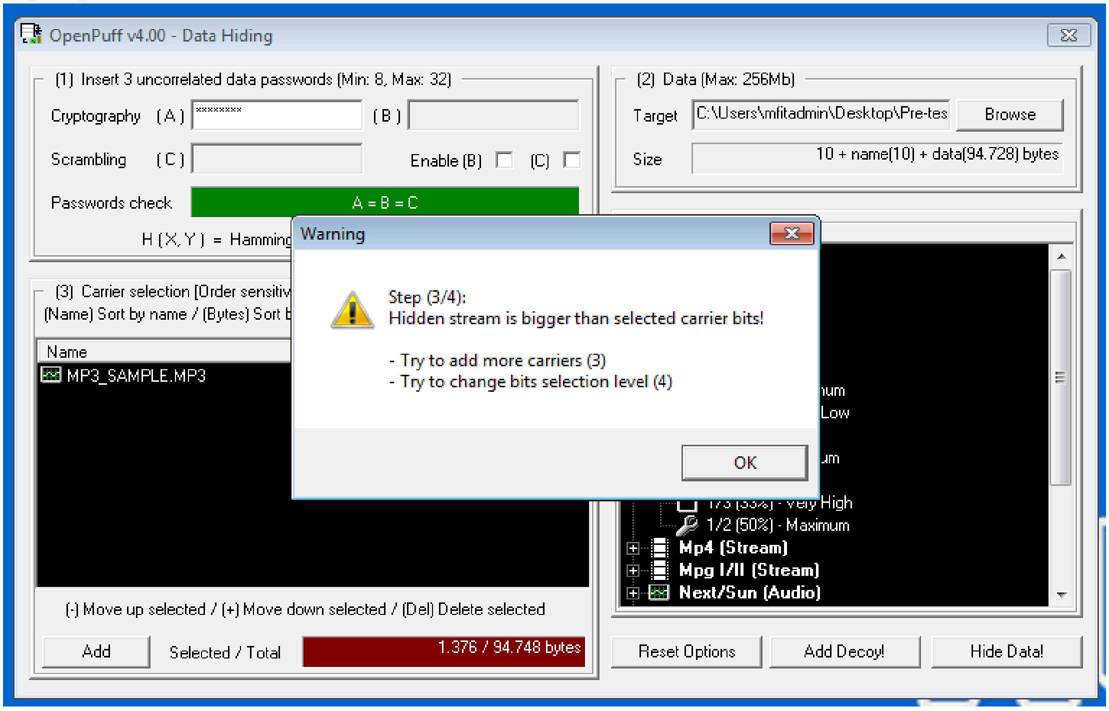
Current file MD5 checksum value:

ed076287532e86365e841e92bfc50d8c

Original file MD5 checksum value (optional). It usually can be found from website or .md5 file

9eca8edf60ebb33d91e59c906064c977

Openpuff failed to embed PNG file into MP3 file



OpenPuff v4.00 - Data Hiding

(1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32)

Cryptography (A) ******** (B)

Scrambling (C)

Enable (B) □ (C) □

Passwords check A = B = C

H (X, Y) = Hamming

(2) Data (Max: 256Mb)

Target C:\Users\mfitadmin\Desktop\Pre-tes

Size 10 + name(10) + data(94.728) bytes

(3) Carrier selection [Order sensitiv

(Name) Sort by name / (Bytes) Sort b

Name

MP3_SAMPLE.MP3

Warning

Step (3/4):
Hidden stream is bigger than selected carrier bits!

- Try to add more carriers (3)
- Try to change bits selection level (4)

ium

Low

um

1/3 (33%) - Very High

1/2 (50%) - Maximum

Mp4 (Stream)

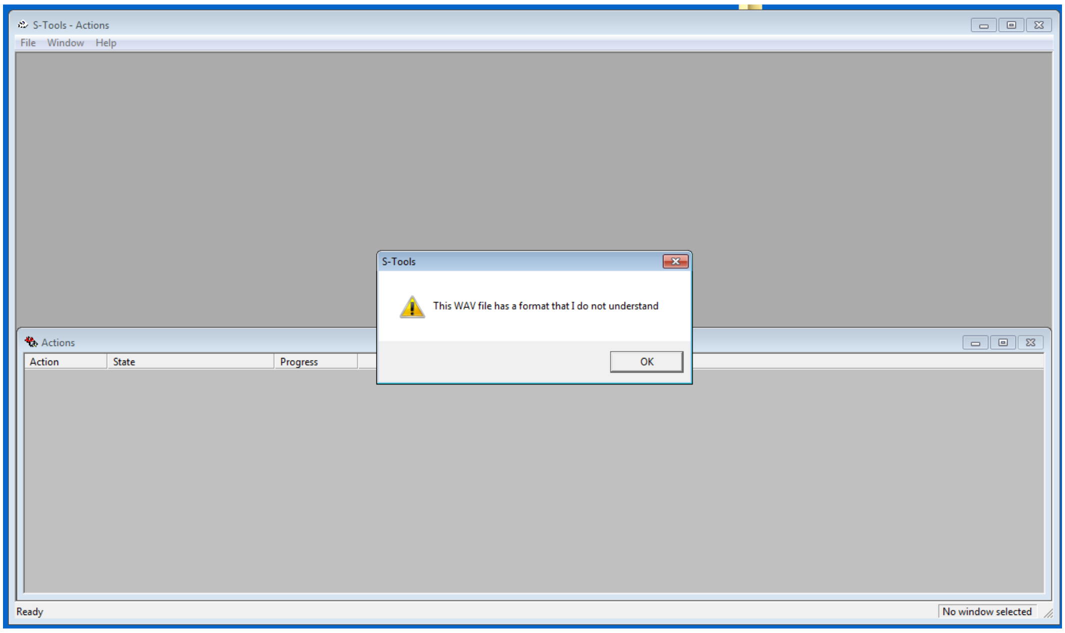
Mpg I/II (Stream)

Next/Sun (Audio)

(-) Move up selected / (+) Move down selected / (Del) Delete selected

Selected / Total 1.376 / 94.748 bytes

S-Tools failed to recognize WAV file

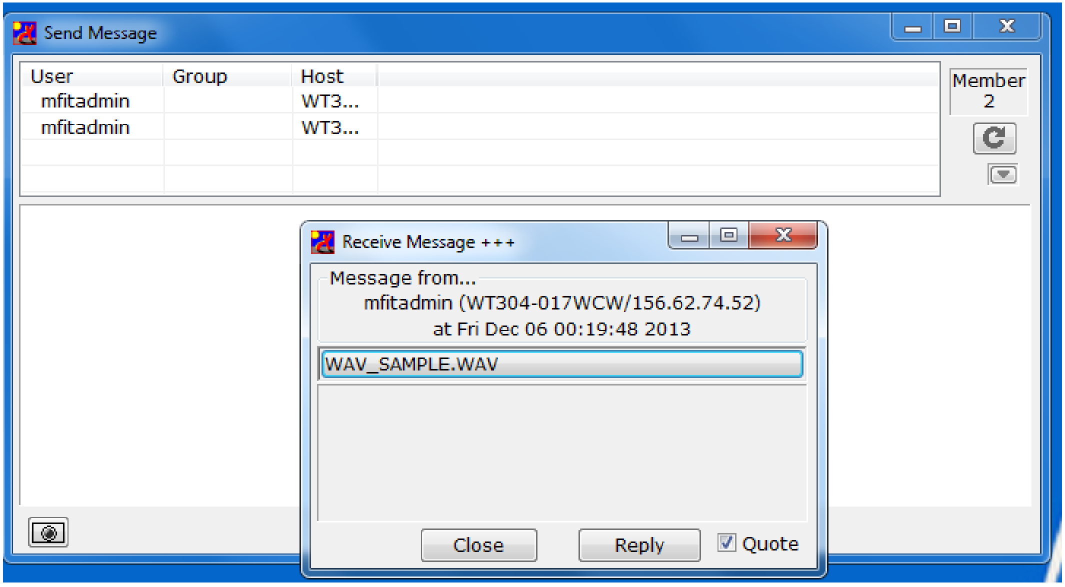


This WAV file has a format that I do not understand

Ready

No window selected

Information on Sending Audio Stream

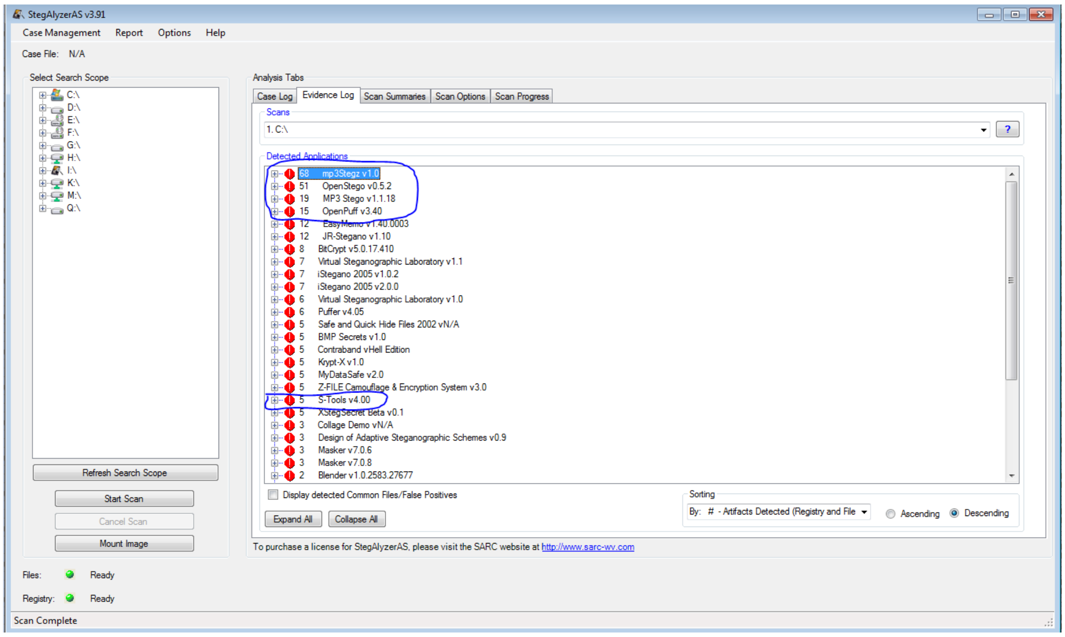


User	Group	Host
mfitadmin		WT3...
mfitadmin		WT3...

Message from...
mfitadmin (WT304-017WCW/156.62.74.52)
at Fri Dec 06 00:19:48 2013

WAV_SAMPLE.WAV

StegAlyzer found stego tools



Case File: N/A

1. C:\

```
68   mp3Stegz v1.0
51   OpenStego v0.5.2
19   MP3 Stego v1.1.18
15   OpenPuff v3.40
12   EasyMemo v1.40.0003
12   JR-Stegano v1.10
8    BitCrypt v5.0.17.410
7    Virtual Steganographic Laboratory v1.1
7    iStegano 2005 v1.0.2
7    iStegano 2005 v2.0.0
6    Virtual Steganographic Laboratory v1.0
6    Puffer v4.05
5    Safe and Quick Hide Files 2002 vN/A
5    BMP Secrets v1.0
5    Contraband vHell Edition
5    Krypt-X v1.0
5    MyDataSafe v2.0
5    Z-FILE Camouflage & Encryption System v3.0
5    S-Tools v4.00
5    XStegSecret Beta v0.1
3    Collage Demo vN/A
3    Design of Adaptive Steganographic Schemes v0.9
3    Masker v7.0.6
3    Masker v7.0.8
2    Blender v1.0.2583.27677
```

□ Display detected Common Files/False Positives

By: # - Artifacts Detected (Registry and File ○ Ascending ● Descending

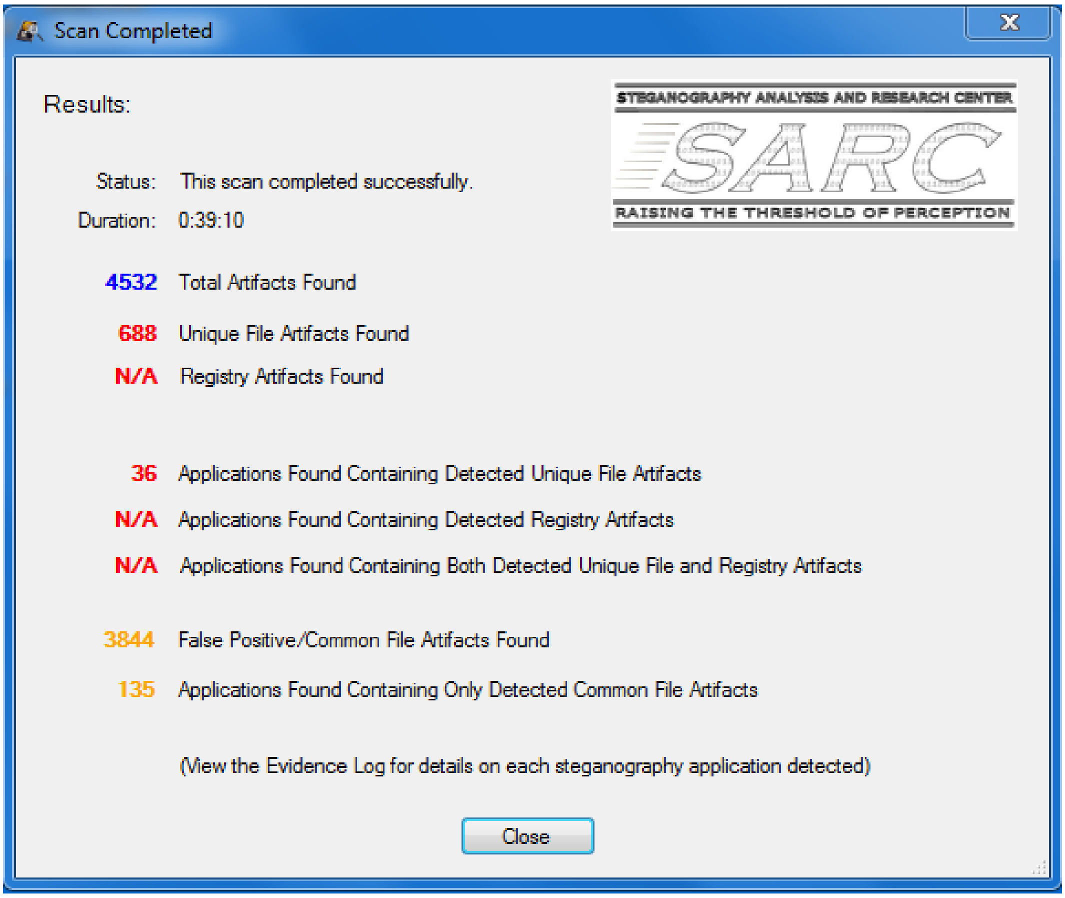
To purchase a license for StegAlyzerAS, please visit the SARC website at http://www.sarc-wv.com

Files: Ready

Registry: Ready

Scan Complete

StegAlyzerAS scanning information


Scan Completed
STEGANOGRAPHY ANALYSIS AND RESEARCH CENTER
SARC
RAISING THE THRESHOLD OF PERCEPTION
Close

Results:

Status: This scan completed successfully.

Duration: 0:39:10

4532 Total Artifacts Found

688 Unique File Artifacts Found

N/A Registry Artifacts Found

36 Applications Found Containing Detected Unique File Artifacts

N/A Applications Found Containing Detected Registry Artifacts

N/A Applications Found Containing Both Detected Unique File and Registry Artifacts

3844 False Positive/Common File Artifacts Found

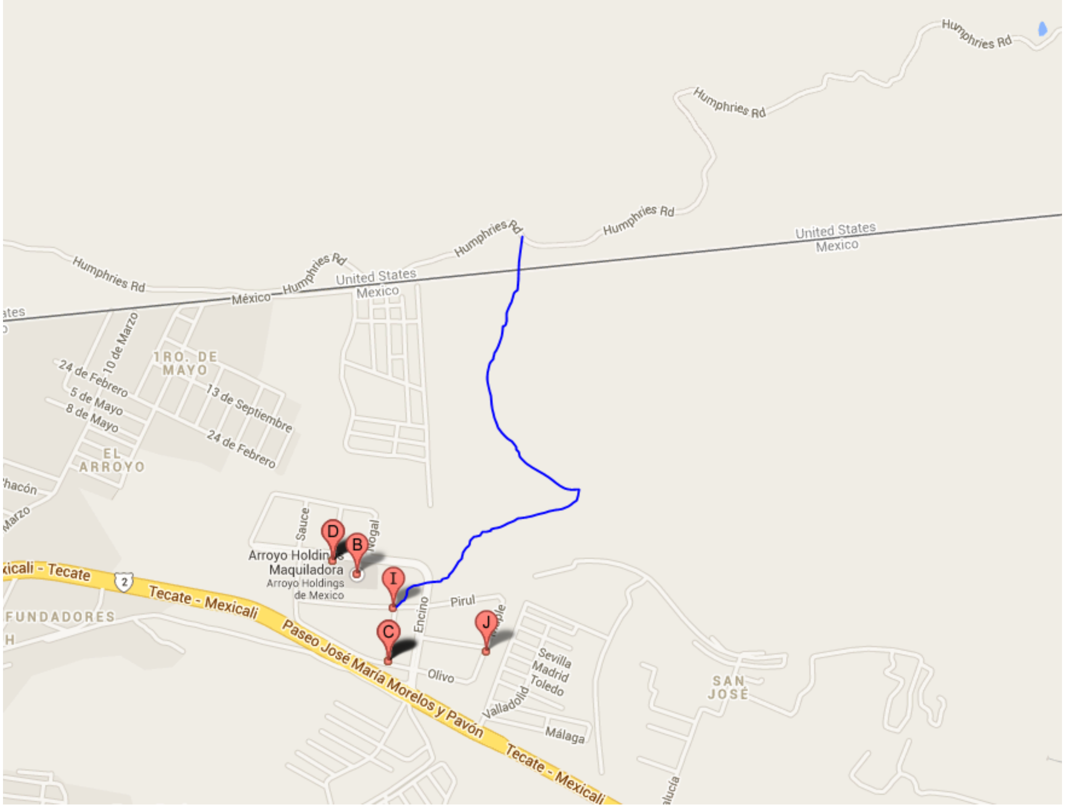
135 Applications Found Containing Only Detected Common File Artifacts

(View the Evidence Log for details on each steganography application detected)

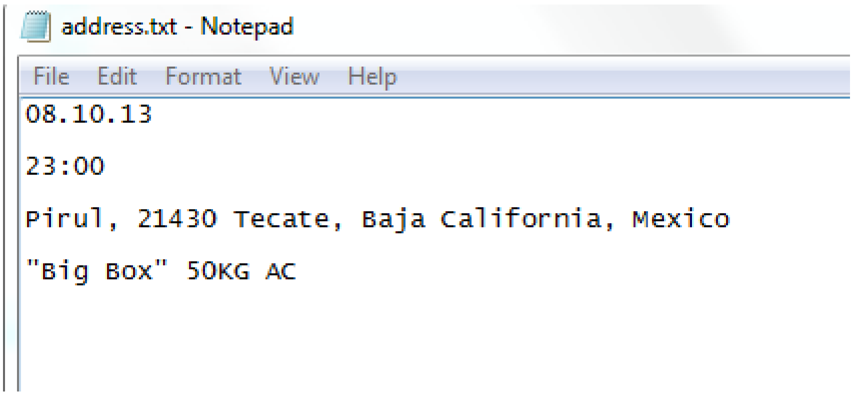
APPENDIX B:

Phase Two: Case Scenario

Secret PNG image



Secret text file (a)



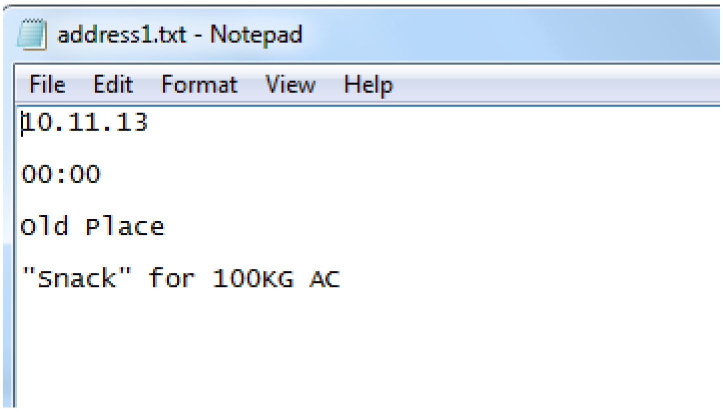
08.10.13

23:00

Pirul, 21430 Tecate, Baja California, Mexico

"Big Box" 50KG AC

Secret test file (b)



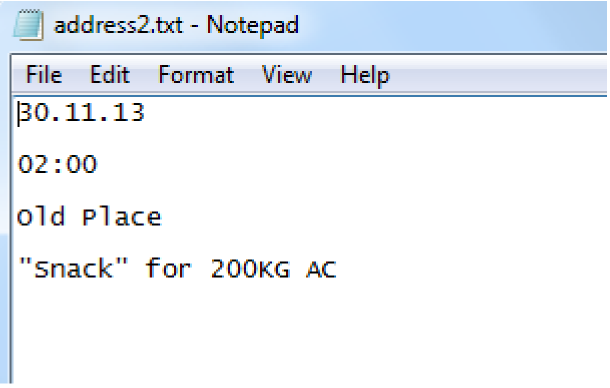
```
10.11.13

00:00

old Place

"Snack" for 100KG AC
```

Secret test file (c)



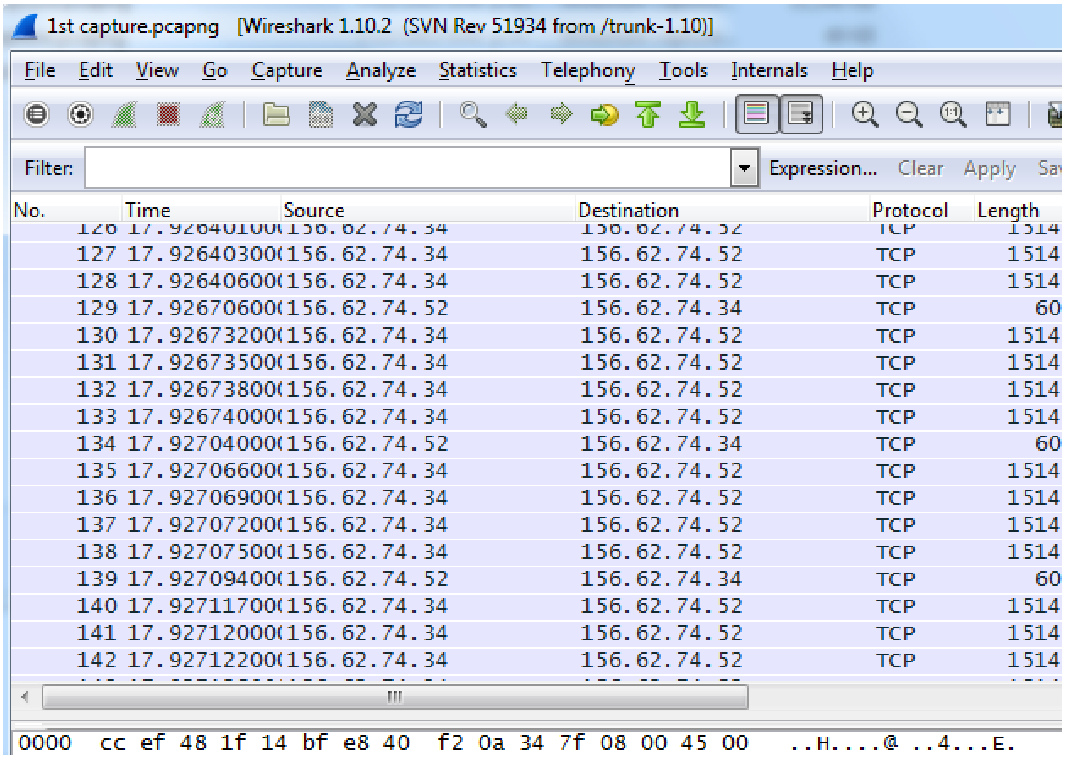
```
30.11.13

02:00

old Place

"Snack" for 200KG AC
```

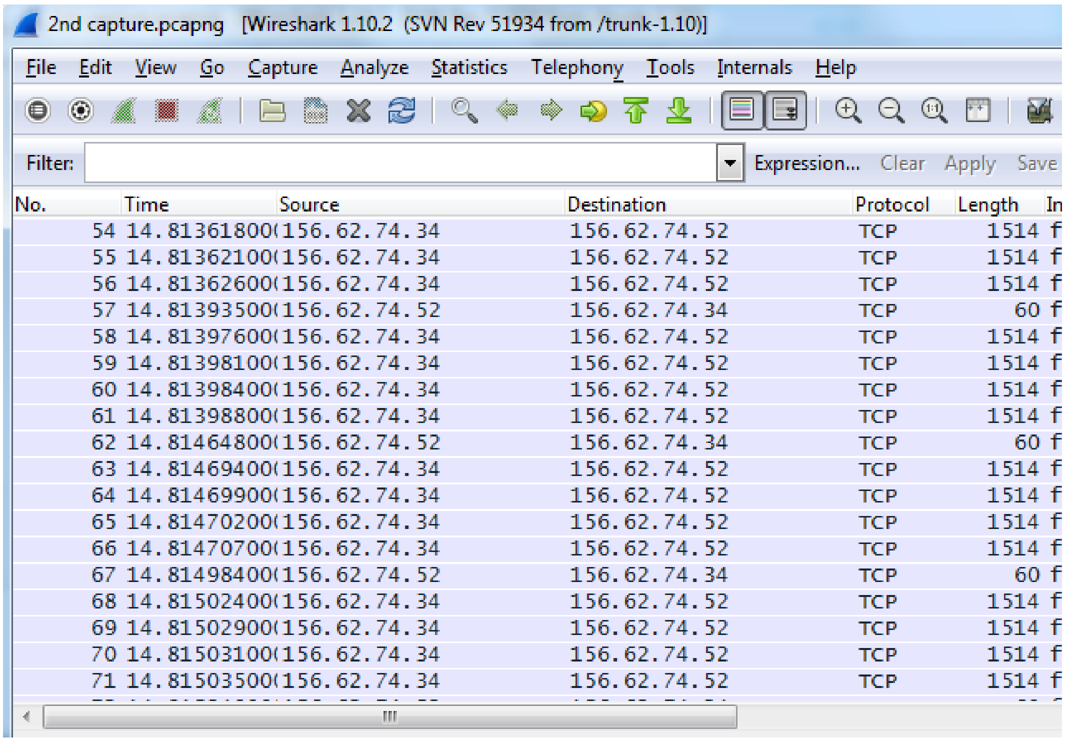
WireShark capture 1


1st capture.pcapng [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Sa

```
No.      Time          Source          Destination       Protocol  Length
     126 17.92640100(156.62.74.34   156.62.74.52     TCP       1514
     127 17.92640300(156.62.74.34   156.62.74.52     TCP       1514
     128 17.92640600(156.62.74.34   156.62.74.52     TCP       1514
     129 17.92670600(156.62.74.52   156.62.74.34     TCP         60
     130 17.92673200(156.62.74.34   156.62.74.52     TCP       1514
     131 17.92673500(156.62.74.34   156.62.74.52     TCP       1514
     132 17.92673800(156.62.74.34   156.62.74.52     TCP       1514
     133 17.92674000(156.62.74.34   156.62.74.52     TCP       1514
     134 17.92704000(156.62.74.52   156.62.74.34     TCP         60
     135 17.92706600(156.62.74.34   156.62.74.52     TCP       1514
     136 17.92706900(156.62.74.34   156.62.74.52     TCP       1514
     137 17.92707200(156.62.74.34   156.62.74.52     TCP       1514
     138 17.92707500(156.62.74.34   156.62.74.52     TCP       1514
     139 17.92709400(156.62.74.52   156.62.74.34     TCP         60
     140 17.92711700(156.62.74.34   156.62.74.52     TCP       1514
     141 17.92712000(156.62.74.34   156.62.74.52     TCP       1514
     142 17.92712200(156.62.74.34   156.62.74.52     TCP       1514

0000  cc ef 48 1f 14 bf e8 40  f2 0a 34 7f 08 00 45 00   ..H....@ ..4...E.
```

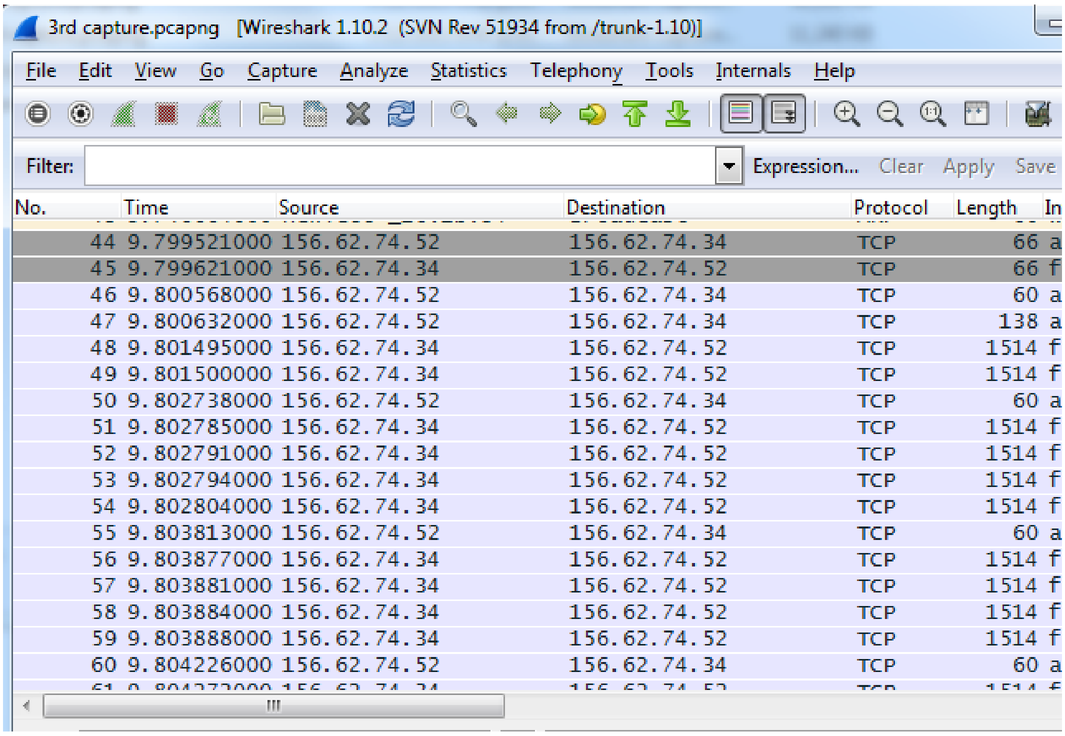
WireShark capture 2



2nd capture.pcapng [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.     Time             Source          Destination      Protocol  Length  In
     54 14.81361800(156.62.74.34    156.62.74.52     TCP        1514 f
     55 14.81362100(156.62.74.34    156.62.74.52     TCP        1514 f
     56 14.81362600(156.62.74.34    156.62.74.52     TCP        1514 f
     57 14.81393500(156.62.74.52    156.62.74.34     TCP          60 f
     58 14.81397600(156.62.74.34    156.62.74.52     TCP        1514 f
     59 14.81398100(156.62.74.34    156.62.74.52     TCP        1514 f
     60 14.81398400(156.62.74.34    156.62.74.52     TCP        1514 f
     61 14.81398800(156.62.74.34    156.62.74.52     TCP        1514 f
     62 14.81464800(156.62.74.52    156.62.74.34     TCP          60 f
     63 14.81469400(156.62.74.34    156.62.74.52     TCP        1514 f
     64 14.81469900(156.62.74.34    156.62.74.52     TCP        1514 f
     65 14.81470200(156.62.74.34    156.62.74.52     TCP        1514 f
     66 14.81470700(156.62.74.34    156.62.74.52     TCP        1514 f
     67 14.81498400(156.62.74.52    156.62.74.34     TCP          60 f
     68 14.81502400(156.62.74.34    156.62.74.52     TCP        1514 f
     69 14.81502900(156.62.74.34    156.62.74.52     TCP        1514 f
     70 14.81503100(156.62.74.34    156.62.74.52     TCP        1514 f
     71 14.81503500(156.62.74.34    156.62.74.52     TCP        1514 f
```

WireShark capture 3



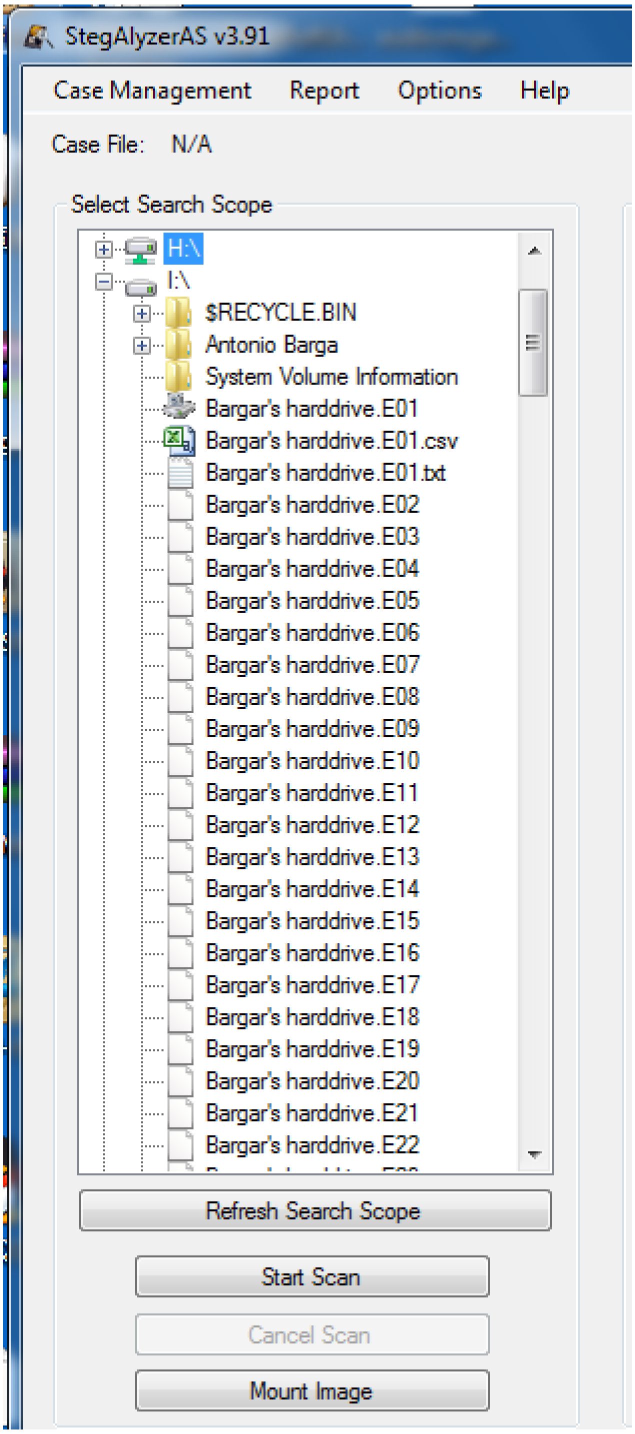
3rd capture.pcapng [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.     Time           Source          Destination      Protocol  Length  In
     44 9.799521000 156.62.74.52    156.62.74.34     TCP          66 a
     45 9.799621000 156.62.74.34    156.62.74.52     TCP          66 f
     46 9.800568000 156.62.74.52    156.62.74.34     TCP          60 a
     47 9.800632000 156.62.74.52    156.62.74.34     TCP         138 a
     48 9.801495000 156.62.74.34    156.62.74.52     TCP        1514 f
     49 9.801500000 156.62.74.34    156.62.74.52     TCP        1514 f
     50 9.802738000 156.62.74.52    156.62.74.34     TCP          60 a
     51 9.802785000 156.62.74.34    156.62.74.52     TCP        1514 f
     52 9.802791000 156.62.74.34    156.62.74.52     TCP        1514 f
     53 9.802794000 156.62.74.34    156.62.74.52     TCP        1514 f
     54 9.802804000 156.62.74.34    156.62.74.52     TCP        1514 f
     55 9.803813000 156.62.74.52    156.62.74.34     TCP          60 a
     56 9.803877000 156.62.74.34    156.62.74.52     TCP        1514 f
     57 9.803881000 156.62.74.34    156.62.74.52     TCP        1514 f
     58 9.803884000 156.62.74.34    156.62.74.52     TCP        1514 f
     59 9.803888000 156.62.74.34    156.62.74.52     TCP        1514 f
     60 9.804226000 156.62.74.52    156.62.74.34     TCP          60 a
```

APPENDIX C:

Phase Four: Analysis

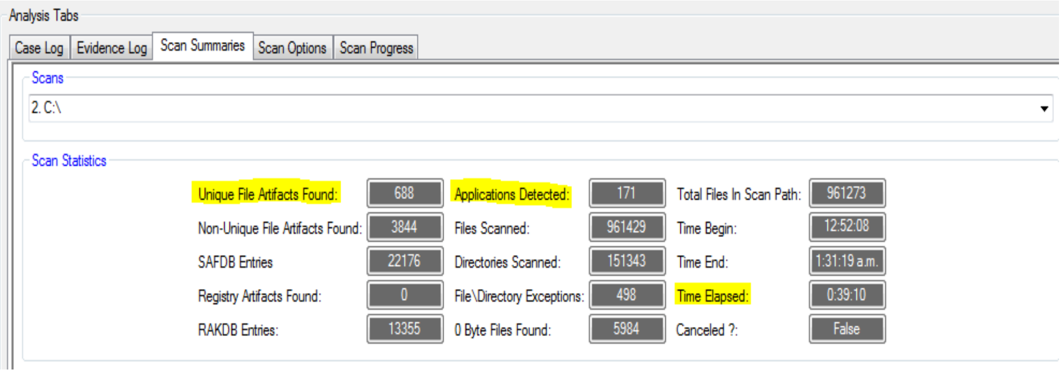
StegAlyzerAS scanning on evidence file



Case File: N/A

```
H:\
I:\
    $RECYCLE.BIN
    Antonio Barga
    System Volume Information
    Bargar's harddrive.E01
    Bargar's harddrive.E01.csv
    Bargar's harddrive.E01.txt
    Bargar's harddrive.E02
    Bargar's harddrive.E03
    Bargar's harddrive.E04
    Bargar's harddrive.E05
    Bargar's harddrive.E06
    Bargar's harddrive.E07
    Bargar's harddrive.E08
    Bargar's harddrive.E09
    Bargar's harddrive.E10
    Bargar's harddrive.E11
    Bargar's harddrive.E12
    Bargar's harddrive.E13
    Bargar's harddrive.E14
    Bargar's harddrive.E15
    Bargar's harddrive.E16
    Bargar's harddrive.E17
    Bargar's harddrive.E18
    Bargar's harddrive.E19
    Bargar's harddrive.E20
    Bargar's harddrive.E21
    Bargar's harddrive.E22
```

Result of scanning



Scans: 2. C:\

Scan Statistics

Unique File Artifacts Found:	688	Applications Detected:	171	Total Files In Scan Path:	961273
Non-Unique File Artifacts Found:	3844	Files Scanned:	961429	Time Begin:	12:52:08
SAFDB Entries	22176	Directories Scanned:	151343	Time End:	1:31:19 a.m.
Registry Artifacts Found:	0	File\Directory Exceptions:	498	Time Elapsed:	0:39:10
RAKDB Entries:	13355	0 Byte Files Found:	5984	Canceled ?:	False

Prefetch file of Openpuff from WinPrefetchView

File name	Full Path	Device Path	Index
ADVAPI32.DLL	C:\Windows\SysWOW64\advapi32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\ADVAPI32.DLL	14
APISETSCHEMA.DLL	C:\Windows\System32\APISETSCHEMA.DLL	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\APISETSCHEMA.DLL	8
COMCTL32.DLL	C:\Windows\winsxs\X86_MICROSOFT.WINDOWS.COMMON-CONTROLS_6595B64144CCF1DF_5.82.7601.17514_NONE_EC83DFFA859149AF\comctl32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\WINSXS\X86_MICROSOFT.WINDOWS.COMMON-CONTROLS_6595B64144CCF1DF_5.82.7601.17514_NONE_EC83DFFA859149AF\COMCTL32.DLL	13
COMDLG32.DLL	C:\Windows\SysWOW64\comdlg32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\COMDLG32.DLL	25
CR	C:\Windows\SysWOW64\CR	\DEVICE\HARDDISKVOLUME1\	19

YPTBASE.DLL	YPTBASE.DLL	WINDOWS\SYSWOW64\CRYPTBASE.DLL	
DWMAPI.DLL	C:\Windows\SysWOW64\dwmapi.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\DWMAPI.DLL	35
GDI32.DLL	C:\Windows\SysWOW64\gdi32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\GDI32.DLL	20
IMM32.DLL	C:\Windows\SysWOW64\imm32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\IMM32.DLL	29
KATRACK.DLL	C:\Windows\katrack.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\KATRACK.DLL	31
KERNEL32.DLL	C:\Windows\System32\kernel32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\KERNEL32.DLL	4
KERNEL32.DLL	C:\Windows\SysWOW64\kernel32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\KERNEL32.DLL	5
KERNELBASE.DL	C:\Windows\SysWOW64\KERNELBASE.DLL	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\KERNELBASE.DLL	9

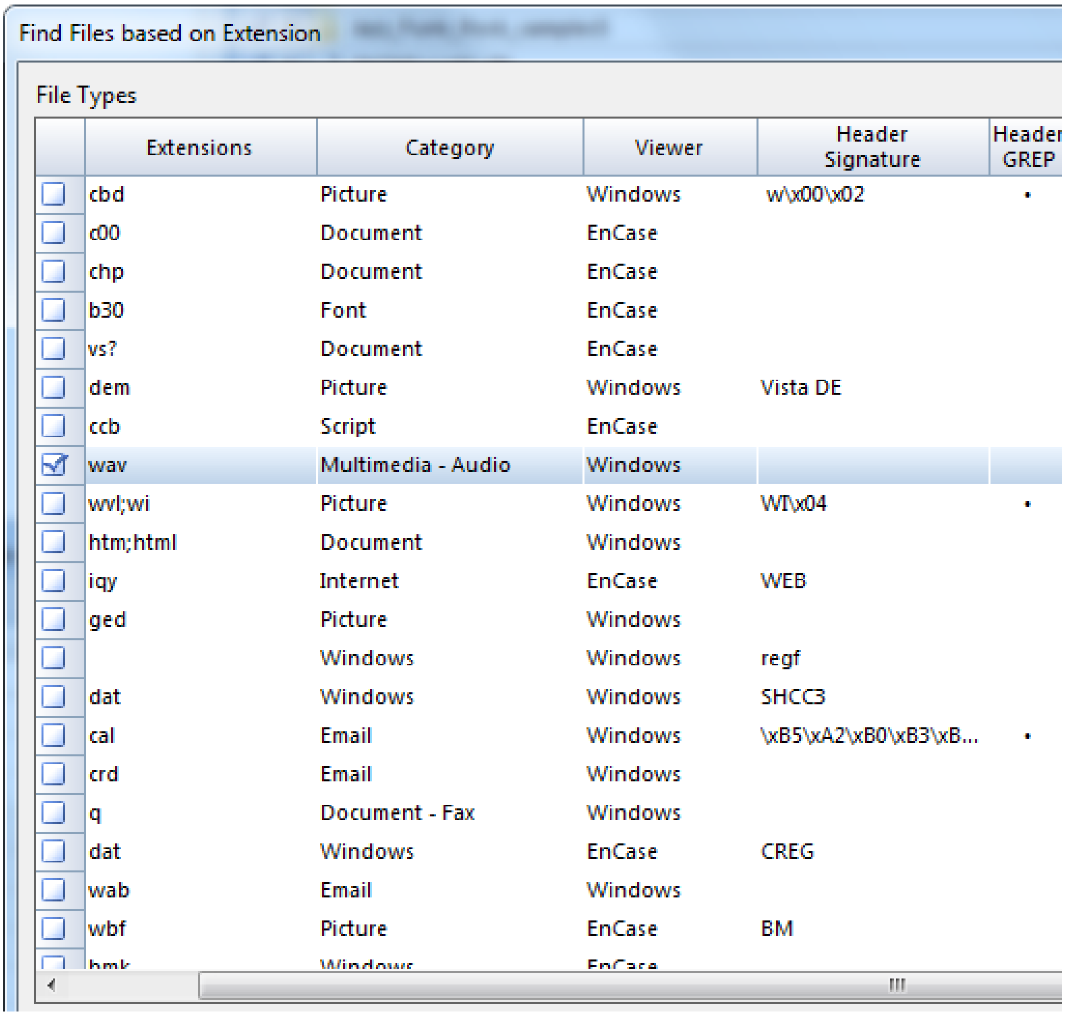
L			
LIBOBFUSCATE.DLL	D:\OPENPUFF\LIBOBFUSCATE.DLL	\DEVICE\HARDDISKVOLUME2\OPENPUFF\LIBOBFUSCATE.DLL	12
LOCALE.NLS	C:\Windows\System32\locale.nls	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\LOCALE.NLS	10
LPK.DLL	C:\Windows\SysWOW64\lpk.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\LPK.DLL	22
MSCTF.DLL	C:\Windows\SysWOW64\msctf.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\MSCTF.DLL	30
MSVCRT.DLL	C:\Windows\SysWOW64\msvcrt.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\MSVCRT.DLL	15
NTDLL.DLL	C:\Windows\SysWOW64\ntdll.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\NTDLL.DLL	7
NTDLL.DLL	C:\Windows\System32\ntdll.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\NTDLL.DLL	0
OLE32.DLL	C:\Windows\SysWOW64\ole32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\OLE32.DLL	28

OPENPUFF.EXE	D:\OPENPUFF\OPENPUFF.EXE	\DEVICE\HARDDISKVOLUME2\OPENPUFF\OPENPUFF.EXE	11
RPCRT4.DLL	C:\Windows\SysWOW64\rpcrt4.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\RPCRT4.DLL	17
SECHOST.DLL	C:\Windows\SysWOW64\sechost.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\SECHOST.DLL	16
SHELL32.DLL	C:\Windows\SysWOW64\shell32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\SHELL32.DLL	27
SHLWAPI.DLL	C:\Windows\SysWOW64\shlwapi.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\SHLWAPI.DLL	26
SORTDEFAULT.NLS	C:\Windows\GLOBALIZATION\Sorting\SORTDEFAULT.NLS	\DEVICE\HARDDISKVOLUME1\WINDOWS\GLOBALIZATION\SORTING\SORTDEFAULT.NLS	32
SSPICLI.DLL	C:\Windows\SysWOW64\sspicli.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\SSPICLI.DLL	18
STATIC	C:\Windows\Fonts\STATICCACHE.DAT	\DEVICE\HARDDISKVOLUME1\WINDOWS\FONTS\STATICCACH	36

CACHE.DAT		E.DAT	
USER32.DLL	C:\Windows\SysWOW64\user32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\USER32.DLL	21
USER32.DLL	C:\Windows\System32\user32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\USER32.DLL	6
USP10.DLL	C:\Windows\SysWOW64\usp10.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\USP10.DLL	23
UXTHEME.DLL	C:\Windows\SysWOW64\uxtheme.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\UXTHEME.DLL	33
WINSPOOL.DRV	C:\Windows\SysWOW64\winspool.drv	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\WINSPOOL.DRV	24
WOW64.DLL	C:\WINDOWS\SYSTEM32\WOW64.DLL	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\WOW64.DLL	1
WOW64CPU.DLL	C:\WINDOWS\SYSTEM32\WOW64CPU.DLL	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\WOW64CPU.DLL	3
WOW6	C:\WINDOWS\SYSTEM32\WOW64WIN.DLL	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\WOW64	2

4WIN.DLL		WIN.DLL	
WTSAPI32.DLL	C:\Windows\SysWOW64\wtsapi32.dll	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\WTSAPI32.DLL	34

Searching WAV file using EnCase

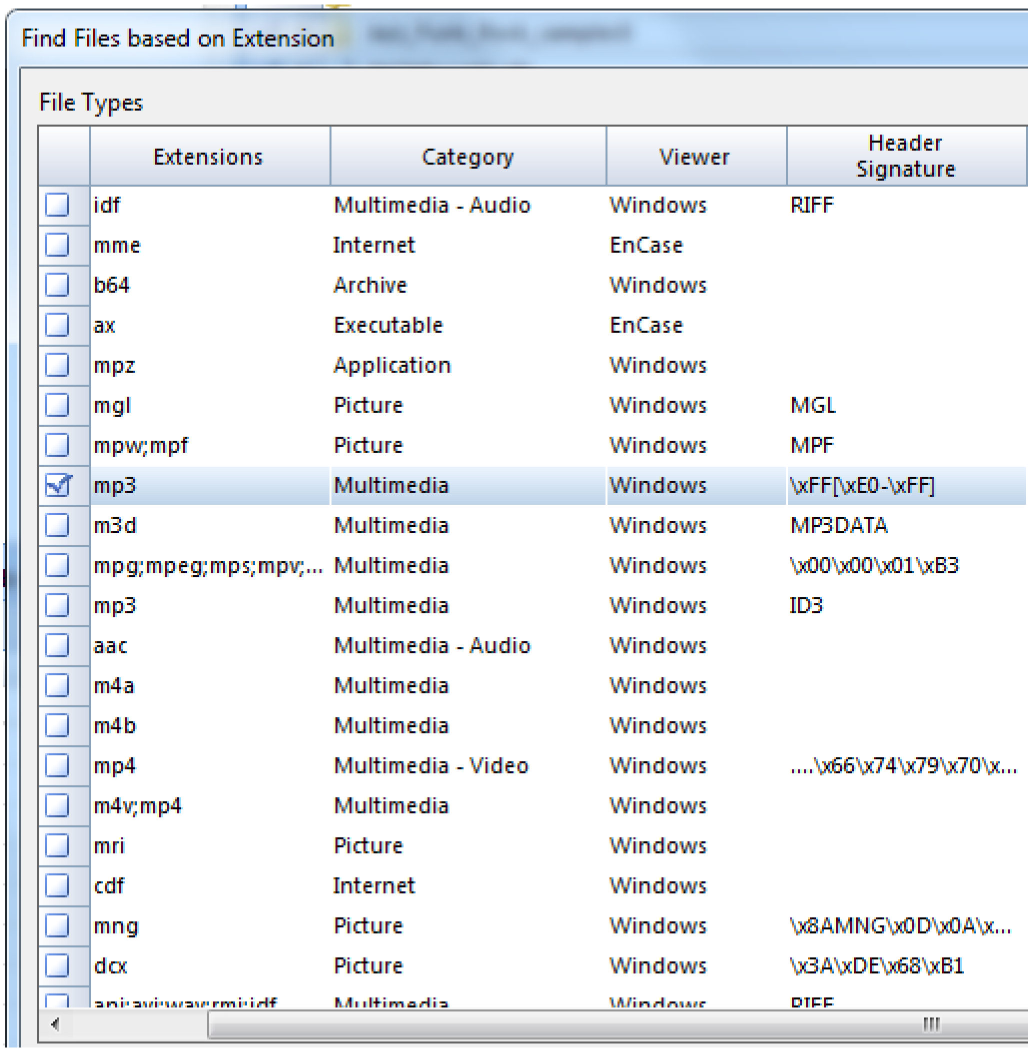


Find Files based on Extension

File Types

	Extensions	Category	Viewer	Header Signature	Header GREP
□	cbd	Picture	Windows	w\x00\x02	•
□	c00	Document	EnCase		
□	chp	Document	EnCase		
□	b30	Font	EnCase		
□	vs?	Document	EnCase		
□	dem	Picture	Windows	Vista DE	
□	ccb	Script	EnCase		
☑	wav	Multimedia - Audio	Windows		
□	wvl;wi	Picture	Windows	WI\x04	•
□	htm;html	Document	Windows		
□	iqy	Internet	EnCase	WEB	
□	ged	Picture	Windows		
□		Windows	Windows	regf	
□	dat	Windows	Windows	SHCC3	
□	cal	Email	Windows	\xB5\xA2\xB0\xB3\xB...	•
□	crd	Email	Windows		
□	q	Document - Fax	Windows		
□	dat	Windows	EnCase	CREG	
□	wab	Email	Windows		
□	wbf	Picture	EnCase	BM	
□	bmk	Windows	EnCase		

Searching MP3 file sing EnCase

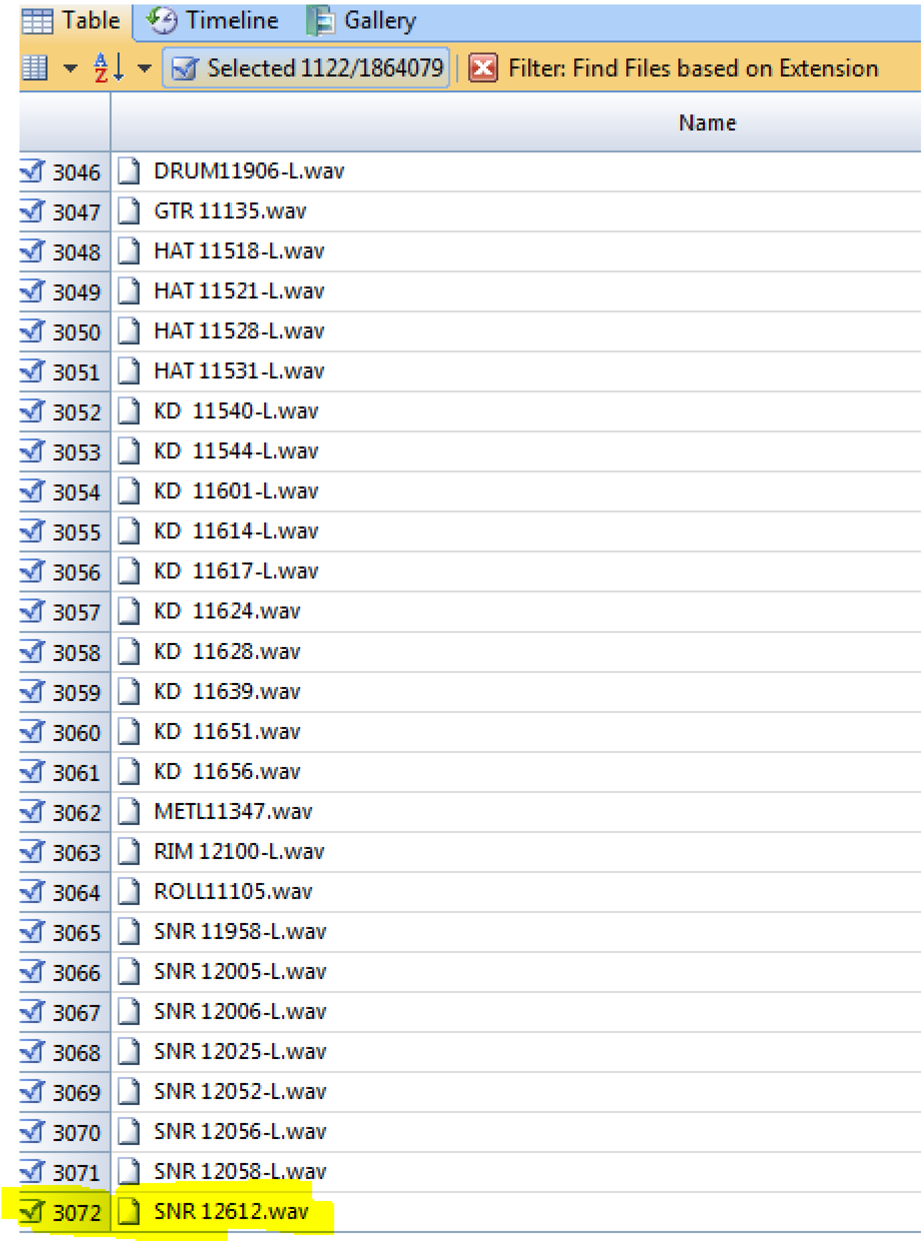


Find Files based on Extension

File Types

	Extensions	Category	Viewer	Header Signature
□	idf	Multimedia - Audio	Windows	RIFF
□	mme	Internet	EnCase	
□	b64	Archive	Windows	
□	ax	Executable	EnCase	
□	mpz	Application	Windows	
□	mgl	Picture	Windows	MGL
□	mpw;mpf	Picture	Windows	MPF
☑	mp3	Multimedia	Windows	\xFF[\xE0-\xFF]
□	m3d	Multimedia	Windows	MP3DATA
□	mpg;mpeg;mps;mpv;...	Multimedia	Windows	\x00\x00\x01\xB3
□	mp3	Multimedia	Windows	ID3
□	aac	Multimedia - Audio	Windows	
□	m4a	Multimedia	Windows	
□	m4b	Multimedia	Windows	
□	mp4	Multimedia - Video	Windows	....\x66\x74\x79\x70\x...
□	m4v;mp4	Multimedia	Windows	
□	mri	Picture	Windows	
□	cdf	Internet	Windows	
□	mng	Picture	Windows	\x8AMNG\x0D\x0A\x...
□	dcx	Picture	Windows	\x3A\xDE\x68\xB1
□	ani;avi;wav;rmi;idf	Multimedia	Windows	RIFF

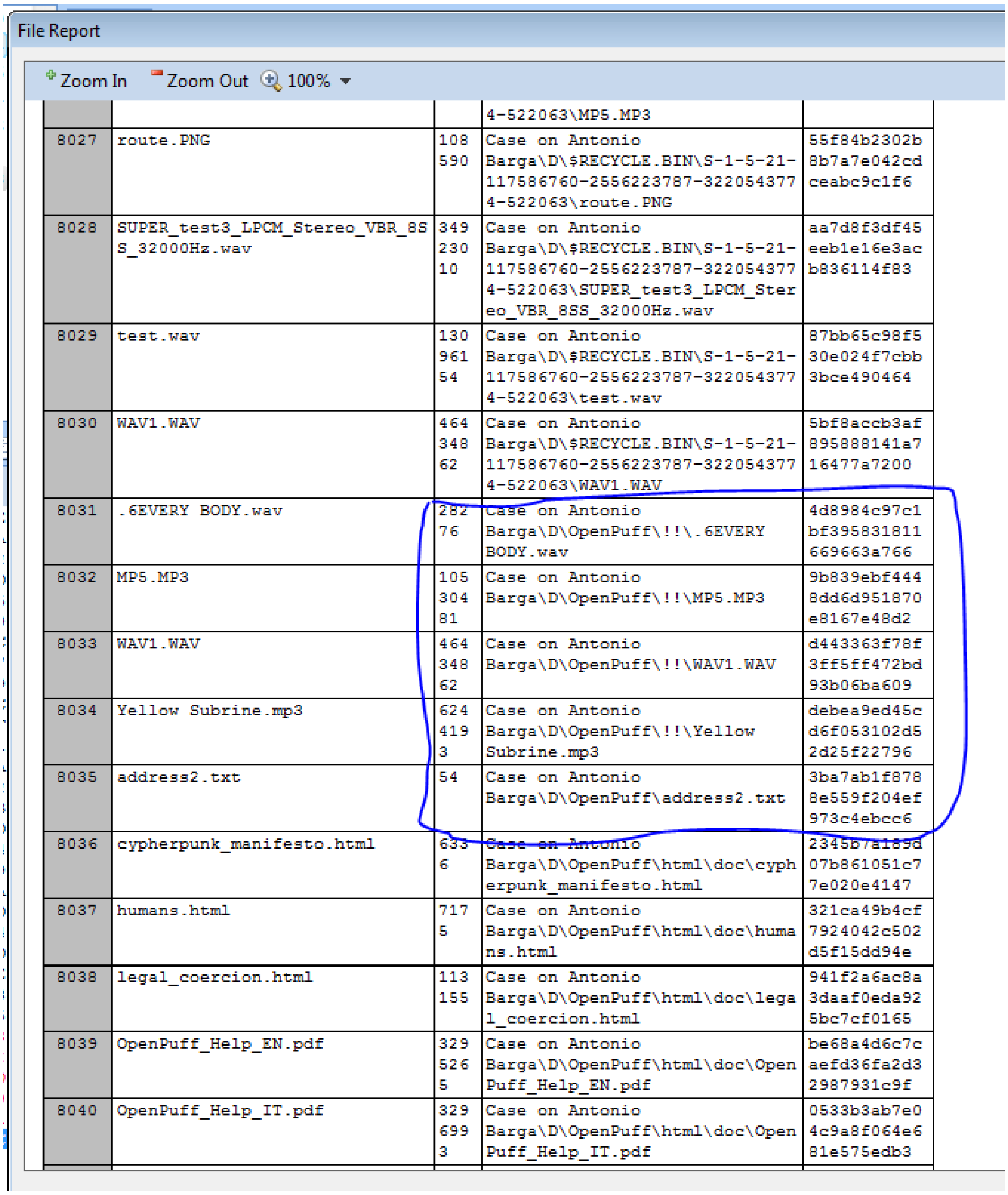
WAV and MP3 searching results



Selected 1122/1864079 ☒ Filter: Find Files based on Extension

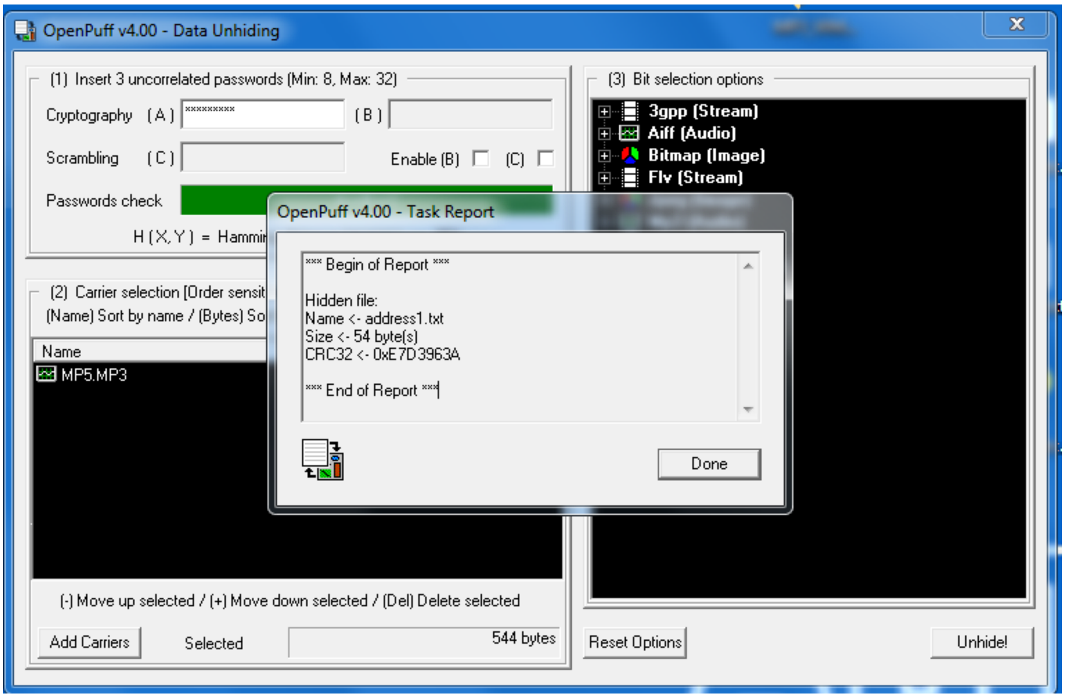
	Name
☑ 3046	DRUM11906-L.wav
☑ 3047	GTR 11135.wav
☑ 3048	HAT 11518-L.wav
☑ 3049	HAT 11521-L.wav
☑ 3050	HAT 11528-L.wav
☑ 3051	HAT 11531-L.wav
☑ 3052	KD 11540-L.wav
☑ 3053	KD 11544-L.wav
☑ 3054	KD 11601-L.wav
☑ 3055	KD 11614-L.wav
☑ 3056	KD 11617-L.wav
☑ 3057	KD 11624.wav
☑ 3058	KD 11628.wav
☑ 3059	KD 11639.wav
☑ 3060	KD 11651.wav
☑ 3061	KD 11656.wav
☑ 3062	METL11347.wav
☑ 3063	RIM 12100-L.wav
☑ 3064	ROLL11105.wav
☑ 3065	SNR 11958-L.wav
☑ 3066	SNR 12005-L.wav
☑ 3067	SNR 12006-L.wav
☑ 3068	SNR 12025-L.wav
☑ 3069	SNR 12052-L.wav
☑ 3070	SNR 12056-L.wav
☑ 3071	SNR 12058-L.wav
☑ 3072	SNR 12612.wav

Report from EnCase showing the location of audio steganorpahy



			4-522063\MP5.MP3	
8027	route.PNG	108590	Case on Antonio Barga\D\$RECYCLE.BIN\S-1-5-21-117586760-2556223787-3220543774-522063\route.PNG	55f84b2302b8b7a7e042cdceabc9c1f6
8028	SUPER_test3_LPCM_Stereo_VBR_8SS_32000Hz.wav	34923010	Case on Antonio Barga\D\$RECYCLE.BIN\S-1-5-21-117586760-2556223787-3220543774-522063\SUPER_test3_LPCM_Stereo_VBR_8SS_32000Hz.wav	aa7d8f3df45eeb1e16e3acb836114f83
8029	test.wav	13096154	Case on Antonio Barga\D\$RECYCLE.BIN\S-1-5-21-117586760-2556223787-3220543774-522063\test.wav	87bb65c98f530e024f7cbb3bce490464
8030	WAV1.WAV	46434862	Case on Antonio Barga\D\$RECYCLE.BIN\S-1-5-21-117586760-2556223787-3220543774-522063\WAV1.WAV	5bf8accb3af895888141a716477a7200
8031	.6EVERY BODY.wav	28276	Case on Antonio Barga\D\OpenPuff\!!\.6EVERY BODY.wav	4d8984c97c1bf395831811669663a766
8032	MP5.MP3	10530481	Case on Antonio Barga\D\OpenPuff\!!\MP5.MP3	9b839ebf4448dd6d951870e8167e48d2
8033	WAV1.WAV	46434862	Case on Antonio Barga\D\OpenPuff\!!\WAV1.WAV	d443363f78f3ff5ff472bd93b06ba609
8034	Yellow Subrine.mp3	6244193	Case on Antonio Barga\D\OpenPuff\!!\Yellow Subrine.mp3	debea9ed45cd6f053102d52d25f22796
8035	address2.txt	54	Case on Antonio Barga\D\OpenPuff\address2.txt	3ba7ab1f8788e559f204ef973c4ebcc6
8036	cypherpunk_manifesto.html	6336	Case on Antonio Barga\D\OpenPuff\html\doc\cypherpunk_manifesto.html	2345b7a189d07b861051c77e020e4147
8037	humans.html	7175	Case on Antonio Barga\D\OpenPuff\html\doc\humans.html	321ca49b4cf7924042c502d5f15dd94e
8038	legal_coercion.html	113155	Case on Antonio Barga\D\OpenPuff\html\doc\legal_coercion.html	941f2a6ac8a3daaf0eda925bc7cf0165
8039	OpenPuff_Help_EN.pdf	3295265	Case on Antonio Barga\D\OpenPuff\html\doc\OpenPuff_Help_EN.pdf	be68a4d6c7caefd36fa2d32987931c9f
8040	OpenPuff_Help_IT.pdf	3296993	Case on Antonio Barga\D\OpenPuff\html\doc\OpenPuff_Help_IT.pdf	0533b3ab7e04c9a8f064e681e575edb3

Successfully extracting address 1.txt file

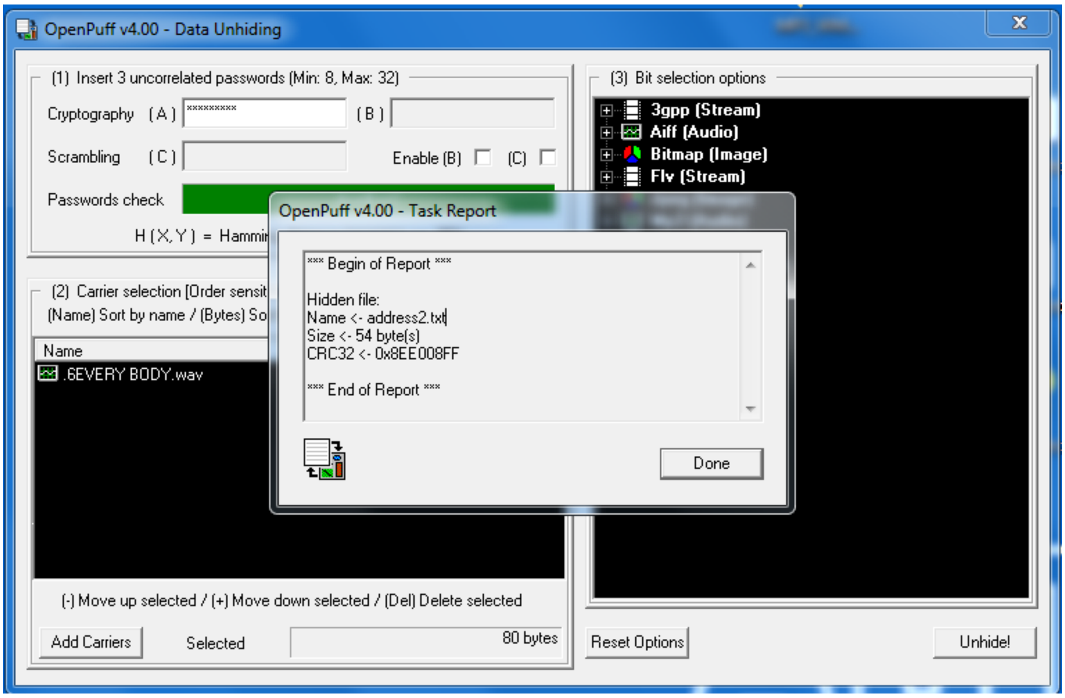

OpenPuff v4.00 - Data Unhiding
(1) Insert 3 uncorrelated passwords (Min: 8, Max: 32)
Cryptography (A) ********** (B)
Scrambling (C) Enable (B) □ (C) □
Passwords check
H (X, Y) = Hammin
(2) Carrier selection [Order sensit
(Name) Sort by name / (Bytes) So
Name
MP5.MP3
(-) Move up selected / (+) Move down selected / (Del) Delete selected
Add Carriers
Selected 544 bytes
(3) Bit selection options
3gpp (Stream)
Aiff (Audio)
Bitmap (Image)
Flv (Stream)
Reset Options
Unhide!
OpenPuff v4.00 - Task Report
Done

```
*** Begin of Report ***

Hidden file:
Name <- address1.txt
Size <- 54 byte(s)
CRC32 <- 0xE7D3963A

*** End of Report ***
```

Successfully extracting address 2.txt file

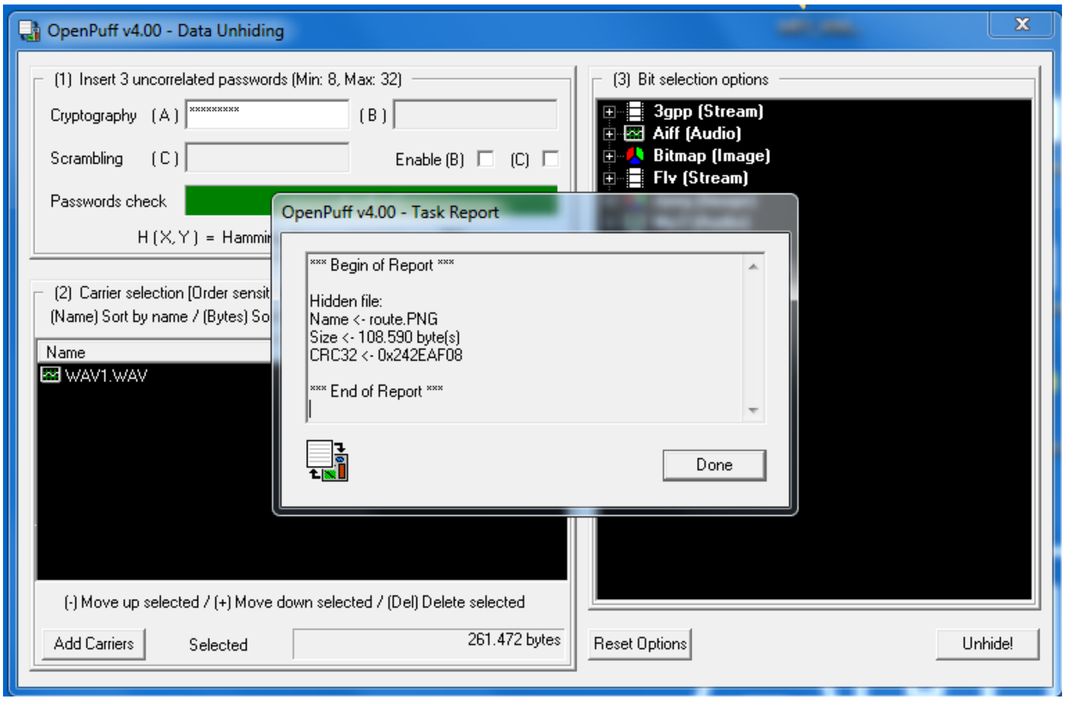

OpenPuff v4.00 - Data Unhiding
(1) Insert 3 uncorrelated passwords (Min: 8, Max: 32)
Cryptography (A) ********** (B)
Scrambling (C) Enable (B) □ (C) □
Passwords check
H (X, Y) = Hammin
(2) Carrier selection [Order sensit
(Name) Sort by name / (Bytes) So
Name
.6EVERY BODY.wav
(-) Move up selected / (+) Move down selected / (Del) Delete selected
Add Carriers
Selected 80 bytes
(3) Bit selection options
3gpp (Stream)
Aiff (Audio)
Bitmap (Image)
Flv (Stream)
Reset Options
Unhide!
OpenPuff v4.00 - Task Report
Done

```
*** Begin of Report ***

Hidden file:
Name <- address2.txt
Size <- 54 byte(s)
CRC32 <- 0x8EE008FF

*** End of Report ***
```

Successfully extracting route.PNG file

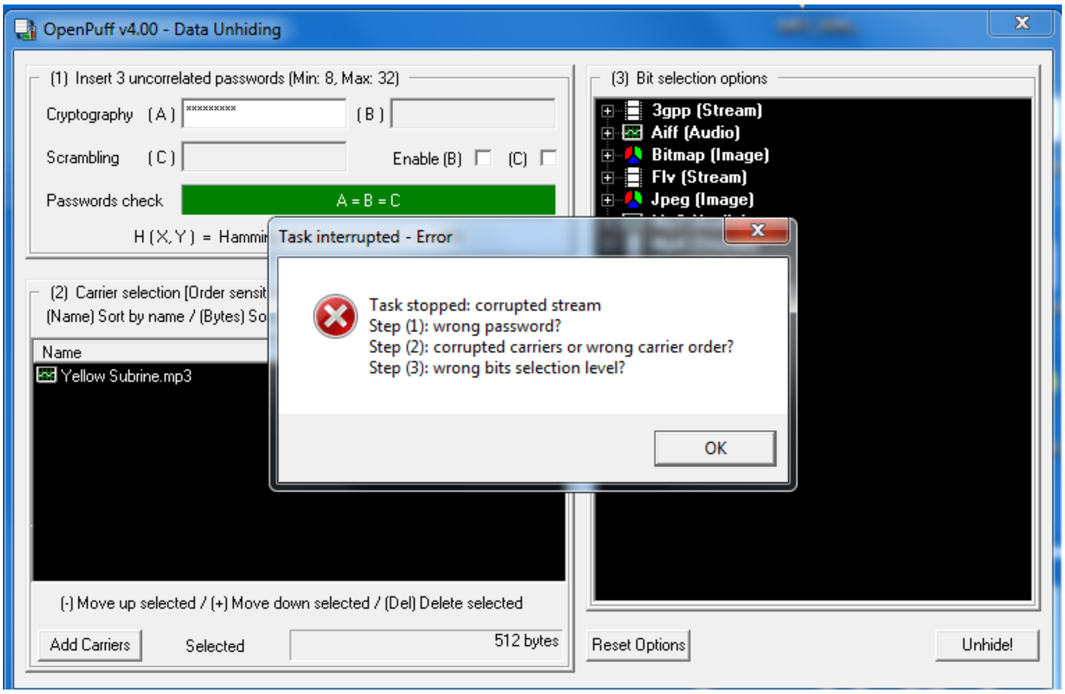

OpenPuff v4.00 - Data Unhiding
(1) Insert 3 uncorrelated passwords (Min: 8, Max: 32)
Cryptography (A) ********** (B)
Scrambling (C) Enable (B) □ (C) □
Passwords check
H (X, Y) = Hammi
(2) Carrier selection [Order sensit
(Name) Sort by name / (Bytes) So
Name
WAV1.WAV
(-) Move up selected / (+) Move down selected / (Del) Delete selected
Add Carriers
Selected 261.472 bytes
(3) Bit selection options
3gpp (Stream)
Aiff (Audio)
Bitmap (Image)
Flv (Stream)
Reset Options
Unhide!
OpenPuff v4.00 - Task Report
Done

```
*** Begin of Report ***

Hidden file:
Name <- route.PNG
Size <- 108.590 byte(s)
CRC32 <- 0x242EAF08

*** End of Report ***
```

Failed to extract address.txt file


OpenPuff v4.00 - Data Unhiding
(1) Insert 3 uncorrelated passwords (Min: 8, Max: 32)
Cryptography (A) ********** (B)
Scrambling (C) Enable (B) □ (C) □
Passwords check A = B = C
H (X, Y) = Hammi
(2) Carrier selection [Order sensit
(Name) Sort by name / (Bytes) So
Name
Yellow Subrine.mp3
(-) Move up selected / (+) Move down selected / (Del) Delete selected
Add Carriers
Selected 512 bytes
(3) Bit selection options
3gpp (Stream)
Aiff (Audio)
Bitmap (Image)
Flv (Stream)
Jpeg (Image)
Reset Options
Unhide!
Task interrupted - Error
OK

Task stopped: corrupted stream
Step (1): wrong password?
Step (2): corrupted carriers or wrong carrier order?
Step (3): wrong bits selection level?