EXTRACTION OF ELECTRONIC EVIDENCE FROM VoIP: IDENTIFICATION & ANALYSIS OF DIGITAL SPEECH

David Irwin
University of South Australia, Australia
david.irwin@unisa.edu.au

Arek Dadej
University of South Australia, Australia
arek.dadej@unisa.edu.au

Jill Slay
University of South Australia, Australia
jill.slay@unisa.edu.au

ABSTRACT

The Voice over Internet Protocol (VoIP) is increasing in popularity as a cost effective and efficient means of making telephone calls via the Internet. However, VoIP may also be an attractive method of communication to criminals as their true identity may be hidden and voice and video communications are encrypted as they are deployed across the Internet. This produces a new set of challenges for forensic analysts compared with traditional wire-tapping of the Public Switched Telephone Network (PSTN) infrastructure, which is not applicable to VoIP. Therefore, other methods of recovering electronic evidence from VoIP are required. This research investigates the analysis and recovery of digitised human voice, which persists in computer memory after a VoIP call.

This paper outlines the ongoing development of a software tool, the purpose of which, determines how remnants of digitised human speech from a VoIP call may be identified within a forensic memory capture based on how the human voice is detected via a microphone and encoded to a digital format using the sound card of a personal computer. This digital format is unencrypted whist stored in Random Access Memory (RAM) before it is passed to the VoIP application for encryption and transmission over the Internet. Similarly, an incoming encrypted VoIP call is decrypted by the VoIP application and passes through RAM unencrypted in order to be played via the speaker output.

A series of controlled tests were undertaken whereby RAM captures were analysed for remnants of digital audio after a VoIP audio call with known

conversation. The identification and analysis of digital audio from RAM attempts to construct an automatic process for the identification and subsequent reconstruction of the audio content of a VoIP call.

This research focuses on the analysis of RAM captures acquired using X-Ways Forensics software. This research topic, guided by a Law Enforcement Agency, uses X-Ways Forensics to simulate a RAM capture which is achieved covertly on a target machine without the user's knowledge, via the Internet, during or after a VoIP call has taken place. The authors assume no knowledge of the technique implemented to recover the covert RAM capture and are asked to base their analysis on a memory capture supplied in the format of a file with a '.txt' extension. The methods of analysis described herein are independent of the acquisition method applied to RAM capture.

The goal of this research is to develop automated software that may be applied to a RAM capture to identify fragments of audio persisting in RAM after a VoIP call has been terminated, using time domain and signal processing technique, frequency domain analysis. Once individual segments of audio have been identified, the feasibility of reproducing audio from a VoIP call may be determined.

Keywords: Computer forensics, digital evidence, electronic evidence, Voice over Internet Protocol, VoIP, Random Access Memory, RAM, Fast Fourier Transform, Frequency Domain analysis

1. INTRODUCTION

Voice over Internet Protocol technology, called VoIP, is an attractive alternative to the Public Switched Telephone Network (PSTN) which may be appealing to criminals, because of (1) VoIP being a global telephony service, in which it is difficult to verify the user's personal identification (2), the security of placing such calls, as many implementations use strong encryption to secure both the voice payload and control messages, and (3) monitoring or tracing such VoIP calls being difficult since conventional methods such as wire-tapping are not applicable to VoIP calls. Therefore, other methods of recovering evidence and information from voice over IP protocol are required. It is essential that forensic computing researchers devise methods to allow law enforcement agencies to overcome some of the aspects of this method of telephony that are advantageous to criminals.

This research aims to develop automated software that may be applied to a RAM capture to identify fragments of audio persisting in RAM after a VoIP call has been terminated. An algorithm searches the RAM capture to identify audio like samples displaying a symmetrical pattern similar to human voice. Digital signal processing techniques are then applied to the suspected audio fragments for analysis in the frequency domain looking at the power spectrum of each sample. An introduction to digital signal processing techniques for

forensic investigators is briefly discussed in section 3.

Voice over Internet Protocol Stack

This introduction provides an overview of the VoIP related protocols for the reader unfamiliar with this technology. VoIP is not a single protocol in itself but rather a collection of a number of co-existing protocols for the encapsulation and transport of voice packets over the Internet, referred to as the protocol stack.

The Internet Protocol (IP) (Postel, 1981) is responsible for providing the internet addresses in its internet header allowing packets to be routed from their source to a destination IP address. The IP header format is shown in Figure1.

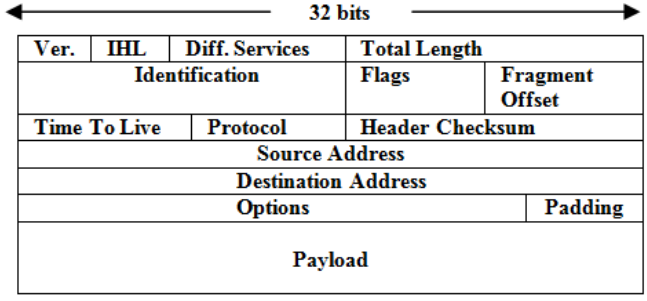

Figure 1 – IP packet header format

The User Datagram Protocol (UDP) (Postel, 1980) is an unreliable transport protocol because it does not guarantee delivery of packets. However, due to its simplicity and ability to transmit packets immediately after they have been created, UDP is well suited to the requirements of VoIP. A single packet may be measured in the order of 10s of milliseconds of audio and the human ear will not be able to detect the loss of packets until the threshold of human audibility is reached, an order of several 100 milliseconds. The UDP header format is shown in Figure 2.

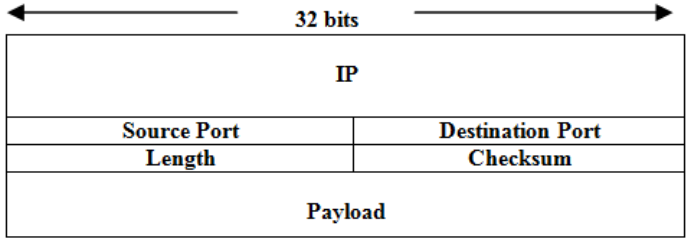

Figure 2 – IP/UDP stack showing the UDP packet header format

The research undertaken in this paper involves a series of experiments using the VoIP application Skype (2009) which makes use of the above mentioned protocols for Internet audio communications. To understand how human audio is converted to a packetised digital format contained within the payload of the IP packet, we briefly outline the common techniques employed in pulse code modulation (PCM). PCM is the technique whereby a digital value is assigned to the analogue value of a short sample of human audio.

1.2 Pulse Code Modulation

PCM is a technique used to digitally represent sampled analogue signals, to produce digital audio in computers and digital telephone systems. The frequency at which the analogue signal is sampled is termed the sampling rate, the number of times per second that a sample is taken. The quality of the sampled audio is determined by the sampling rate and the number of bits assigned to represent the digitised sample. The higher the number of bits the greater the accuracy of the digital representation of the analogue signal, often referred to as the bit-depth.

Figure 3 demonstrates the digital representation of an analogue signal, in which the magnitude of the analogue signal is sampled regularly at uniform intervals, with each sample being assigned to the nearest value within a range of digital steps, referred to as quantisation.

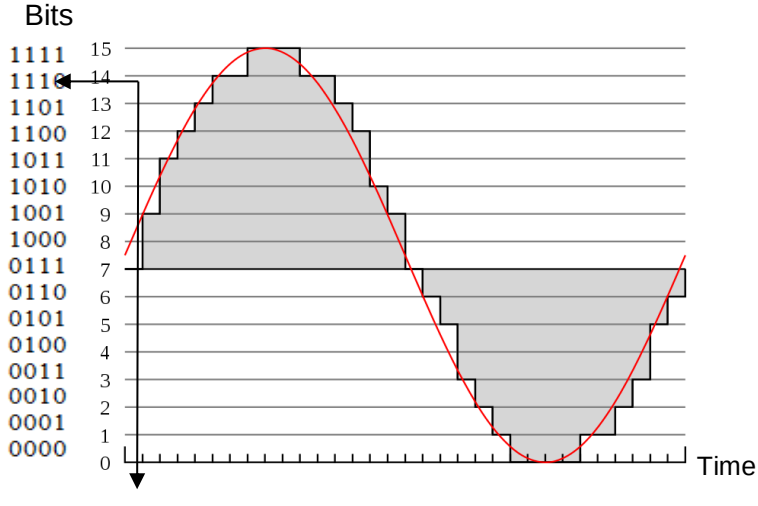

Figure 3 – 4-bit quantisation of an analogue signal

The original analogue signal, in this instance a sinusoid, depicted as the red curve, is sampled at regular intervals in time. The corresponding value for the

4-bit PCM quantisation is determined by using an imaginary vertical line on the time axis until it intersects the sinusoid and reading the digital value pointed at on the 'bits' axis with an imaginary horizontal line. The PCM value for the sample shown in the figure is '1101'.

A technique called companding, commonly deployed in digital telephony systems is the act of applying compression to the analogue input signal before passing through the analogue-to-digital converter. Figure 4a shows an analogue signal prior to compression whereas Figure 4b shows the compressed signal.

After compression, the analogue signal is digitised for transmission in a suitable format for VoIP applications. After the signal is transmitted across the Internet and received at the destination, it needs to be expanded to its original form.

The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) proposed recommendations on speech coding to standardise interoperability between telecommunications carriers resulted in the G.711 codec (ITU-T G7.11, 1972), which defines two main compression algorithms, A-law (Europe) and µ-law (U.S.A.). The choice of codec will be determined by the VoIP application if it is proprietary or may be chosen by the user from a list.

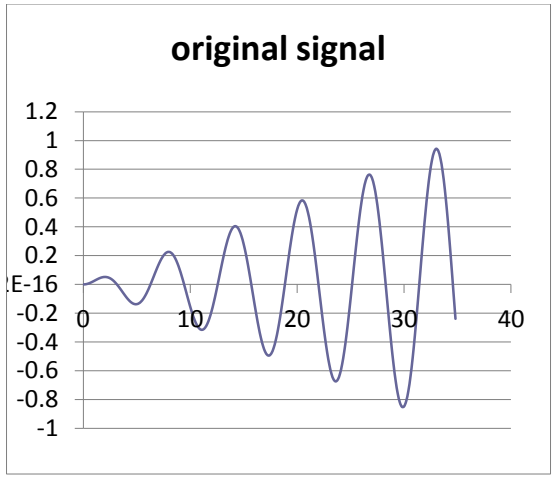

Figure 4a – Original signal

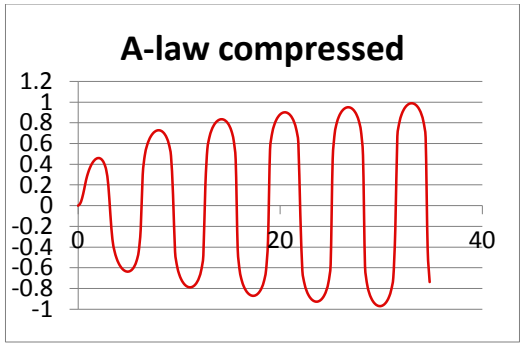

Figure 4b – After compressing

1.3 Defining Digital Forensics

Digital forensics as defined by Beebe, Clark, Deitrich, Ko and Ko (2011) is the extraction of data from digital devices (e.g. personal computers, mobile phones, digital cameras, networking devices, web/file/email servers etc.) to reconstruct events, confirm or refute allegations of criminal activities and/or obtain intelligence information. Additionally, Yasinsac and Manzano (2001) define the digital forensic domain stating that digital forensics involves the analysis of electronic devices for the purpose of discovery and retrieval of information regarding the criminal use of technology.

Within Australia, McKemmish (1999) defines digital forensic investigations with the use of a four-phase model to describe digital forensic investigations is widely cited and considered seminal research within the domain of digital forensics. The phases are defined as the Identification, Preservation, Analysis and Presentation (IPAP) of digital evidence.

The results of activities performed upon digital evidence in order to retrieve information must be legally acceptable f or court proceedings. The legal requirements of digital forensics defined by Civie and Civie (1998) states:

> *'The pursuit of knowledge by uncovering elemental evidence extracted from a computer in a manner suitable for court proceedings.'*

Therefore the combination of both legal and technical requirements is required to demonstrate in court proceedings the phases of the digital forensics investigation, analysis and results in a manner acceptable in a court of law. Carrier (2003) introduces a scientific approach to the definition of digital forensics methods stating:

> *'The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis,*

interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.'

To this end, the research and development of the software tool described herein for the analysis and recovery of fragmented audio from a VoIP call cannot be referred to as 'forensic' based on the definitions above at this time. However, this paper will outline the achievements and worthiness of this software tool so far and its suitability as a tool to assist forensic investigators in the analysis of captured memory.

1.4 Memory Acquisition

Traditional forensics memory capture takes place whilst the forensic investigator is on-site, and performs the physical memory capture from the target machine to the investigator's destination disk/image file system. The difference between 'dead' and 'live' acquisition is described below.

- Dead Acquisition
  - Occurs when the data from a suspect system is copied without the assistance of the suspect operating system.
  - Historically, the term 'dead' refers to the state of only the operating system, so a dead acquisition can use the hardware from the suspect system as long as it is booted from a trusted CD or floppy.
- Live Acquisition
  - Where the suspect operating system is still running and being used to copy data.
  - Acquisition tool needs to be able to access 'open' files (files in use)
  - Beneficial in circumstances where an encrypted data volume is mounted

– On a compromised system, there is the risk that the attacker has modified the operating system or other software to provide false data during acquisition.

Imaging a computer's hard disk can be a lengthy process. During the acquisition process, if the forensic investigator saves the data to a file, he/she will have the choice of what format the image will be e.g.

- A raw image contains only the data from the source device. Easy to compare the image with the source data.
- An embedded image contains data from the source device and additional descriptive data about the acquisition. User inputted data, hash values, dates & times
- Some tools will create a raw image and save the additional descriptive data to a separate file.

As most forensic tools support raw images, the raw image it is the most flexible format. A freely available utility for most operating systems called 'dd' (The Open Group, 2010) can make exact copies of memory that are suitable for forensic analysis without the need to own commercial forensic software packages.

1.5 Why RAM Acquisition?

Digital forensics tools play a vital role in reliably extracting information for analysis and presentation for industrial or legal purposes. These tools are typically used to investigate computer crimes, by identifying evidence that can be of probative value in a court of law. Digital forensics tools are rapidly becoming a substantial part of investigations all over the world, in both the law enforcement and private sector domains (Hibishi, Vidor, & Cranor, 2011).

Efficient examination of digital evidence would not be possible without the use of digital forensic tools. While an understanding of the scientifically derived processes and the volatility of digital evidence is required by analysis teams and technicians, it is not feasible to interpret the volumes of evidence required for investigation on a given case manually. Both expert witnesses and digital forensic practitioners are reliant on a set of tools for interpreting digital evidence and to help bridge the gap of understanding between the technical details of digital technologies and the evidence presented to a jury in court (Schatz, 2007). Due to the vast and complex variety of devices required for analysis by digital forensics teams, there exist many different tools suited to handling each. These tools perform different roles including acquisition, examination and analysis.

The use of RAM captures is more easily explained in terms of the increasing sources of digital evidence complexity and the technological advancements in the volume size of storage media. Digital evidence complexity, the vast array of different digital evidence sources, each with their own ways of storing data and retrieving data increases the difficulty of forensic investigation. Similarly the volume of digital evidence, i.e. the amount of digital evidence required for practitioners of digital forensics to preserve, analyse and present in a given case is increasing exponentially. It is both a difficult and time consuming process to search and comprehend large quantities of digital evidence. These issues are supported by key researchers within the domain as pertinent issues for study (Casey, Gordon, & Leeson, 2005; Mohay, 2005). This may impact court proceedings due to increased case backlogs and the inability for digital forensic investigation teams to complete cases in a reasonable period.

The current tools and techniques used to analyse digital evidence are not scaling and adapting to the increased data volume or complex array of devices now required for analysis with manual analysis remaining commonplace throughout the digital forensics industry.

The Windows Forensics Analysis Tool Kit (Carvey, 2007) discusses remote response methodology, whereby a series of commands may be executed against a system across a network using a Windows batch file comprising the name or IP address of the target system and the username/password logon credentials. The batch file contains executable code, which can be copied to and run on the target system with the corresponding output saved in a file on the target machine. The only limitation to perform analysis of the target machine is the ability to remotely login to the target system via the network.

This research introduces novel techniques and approaches with respect to the analysis of captured memory, required to address the key issues above resulting in Law Enforcement choosing 'live' RAM capture to minimise the complexity and volume of data to be analysed.

1.6 X-Ways Forensics

This focus of this research is on the analysis of the contents of RAM captures and as such it does not investigate memory acquisition techniques. No Law Enforcement Agency supplied RAM captures from a target machine, thus requiring the researchers to simulate a RAM capture, using X-Ways Forensics, computer forensics software, as shown in Figure 5.

The option exists to capture an individual running process and the RAM allocated to that process e.g. VoIP application Skype, expanded in Figure 5b. However, to maintain the same conditions for each capture, the entire physical memory is captured, as show in Figure 5a.

The X Ways Forensics RAM editor allows one to examine the physical

RAM/main memory and the logical memory of a process (i.e. a program that is being executed) where all memory pages committed to a process are presented in a continuous block. If one selects one of the listed processes, one may access either the so-called primary memory or the entire memory of this process, or one of the loaded modules.

The primary memory is used by programs for nearly all purposes. Usually it also contains the main module of a process (the EXE file), the stack; and the heap. The "entire memory" contains the whole logical memory of a process including the part of memory that is shared among all processes, except system modules.

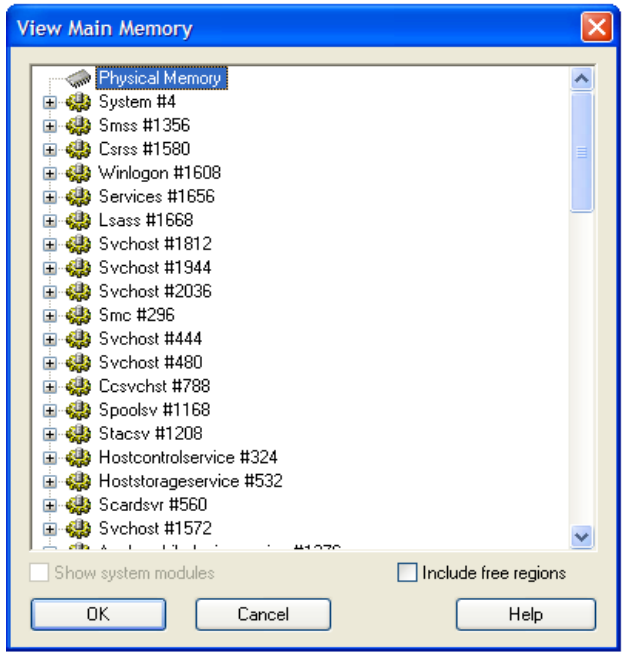

Figure 5a – Entire RAM physical memory.

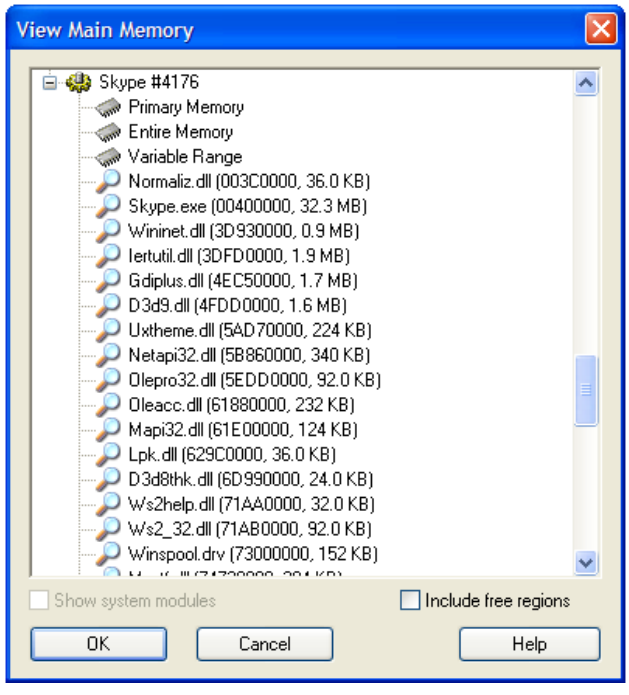

Figure 5b – Individual RAM processes e.g. Skype

When one opens the local physical RAM, processes will be listed in the directory browser, even hidden processes, with their timestamps and process IDs, and their own respective memory address spaces can be individually viewed with pages concatenated in correct logical order as seen by each process.

The purpose of this research is not to use the powerful capability of the X-Ways Forensics RAM editor to reverse engineer captures and identify running processes, shown in Figure 6.

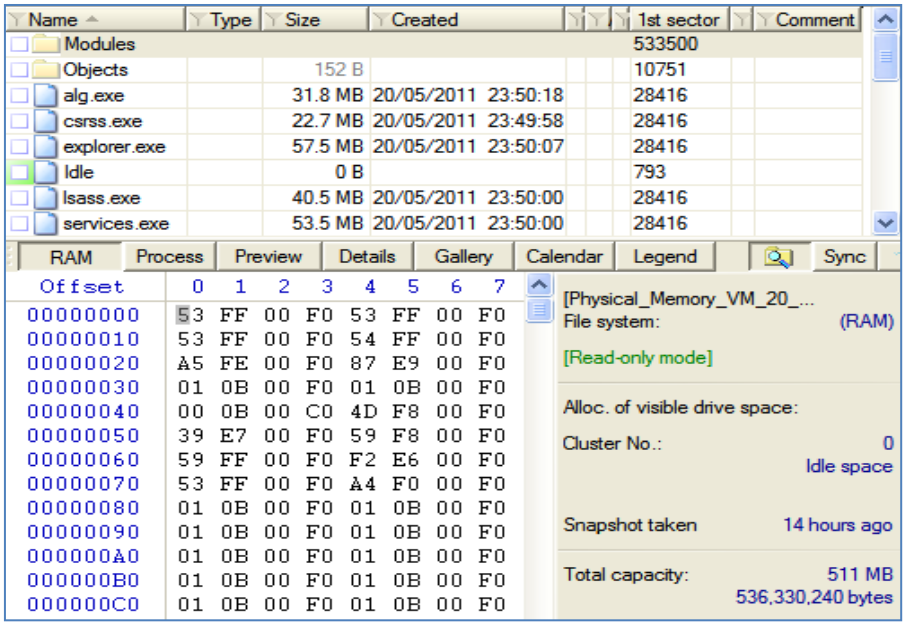

Figure 6– Byte offsets within a RAM capture for modules and objects.

2 METHODOLOGY & EXPERIMENTS

This research approach draws on the strengths of both quantitative and qualitative research approaches. This research focuses on outcomes that are of practical use, the creation of knowledge that advances digital forensics based on tangible and measurable results. This research strives for objectivity and measurability via controlled experiments using algorithms developed to pattern recognise human speech.

2.1 Baseline Experiments

X-Ways Forensics software was installed on the target Windows XP virtual machine initiating the VoIP call, and the capture taken after the VoIP call was terminated and the VoIP application closed down. The amount of RAM captured is 512MB. However, to reduce the possibility of false positives, identifying suspect audio fragments, which in fact are not audio and false negatives, failing to identify audio which is indeed fragments of suspect audio, the experiment shown in Table 1 was implemented.

Table 1 – Initial results from RAM capture analysis.

RAM Capture	Detection Method	Expected outcome	Actual outcome	Byte size	Segments
1	Visual inspection	No audio	Sinusoids	4096	24
2	Visual inspection	No audio	Sinusoids	4096	24
3	Visual inspection	No audio	Sinusoids	4096	24

The initial analysis technique involved displaying the byte values of RAM graphically, and visually inspecting the RAM contents. This is a time consuming process but produced some unexpected results. No audio was introduced to the system, however, a sinusoidal pattern with an amplitude offset was detected, in all three RAM captures tested and was repeatable. The authors believe this to be the Windows XP sound that is played at system start up shown in Figure 7.

Figure 7 – Windows XP signature audio at start up.

The audio segments found resembled those shown in Figure 8b whereas Figure 8a displays the similar sinusoid with no amplitude offset. This initial technique of creating baseline knowledge of the RAM contents before introducing known audio fragments and a VoIP call is essential.

Direct current voltage (dc) is an electronics term to identify an offsetting of a signal from zero. This offset may be implemented in hardware such as a sound card.

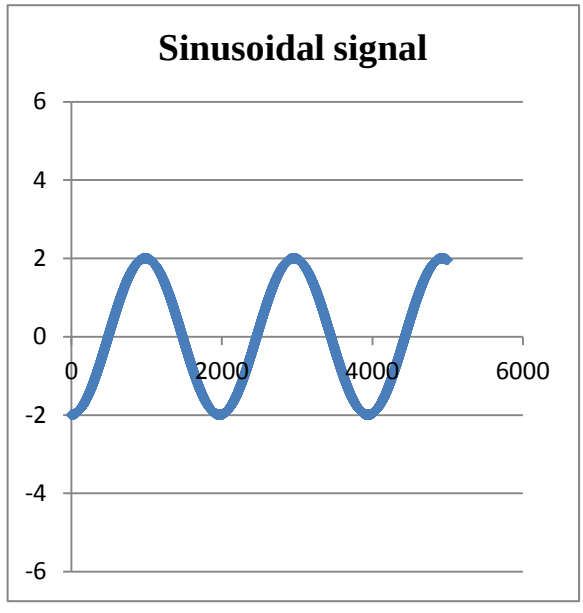

Figure 8a – A sinusoidal signal.

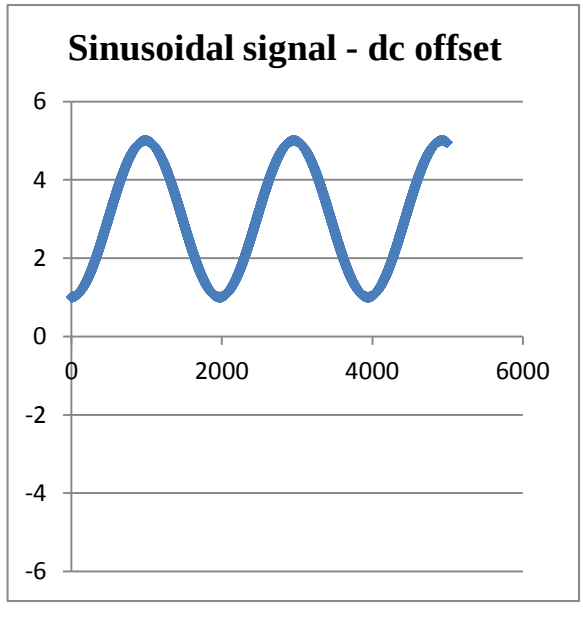

Figure 8b – A sinusoidal signal with amplitude offset.

2.1 Introduce Known Audio

The next round of experiments involved introducing a known audio pattern into the RAM contents, still without the introduction of a VoIP call at this time. The audio signal has been selected from the TIMIT Corpus (Garofolo,

Lamel, Fisher, Fiscus, Pallett, Dahlgren, & Zue, 1993) which provides speech data for acoustic phonetic studies. These are 16-bit, 16 KHz time aligned speech waveforms i.e. the byte locations within the waveform have been identified for each uttered word and the phonemes that constitute that word. The chosen phonetic sentence known as 'LDC93S1W' is single channel PCM. Five samples of audio extracted from the phonetic sentence, each 3000 bytes in size, were randomly inserted into a RAM capture using X-Ways hex editor. An example of one segment of the known audio is shown in Figure 9.

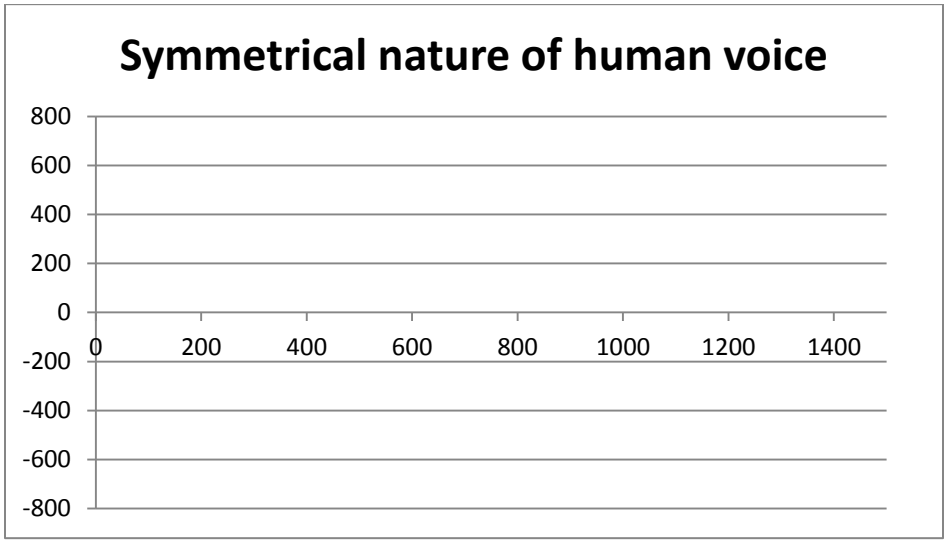

Figure 9 – An audio extract from the known phonetic sentence inserted into RAM.

This experiment was repeated three times, each with a different ordering of the known audio and its location within the RAM capture. The results are shown in Table 2.

Table 2 – The insertion of known audio pattern into RAM captures.

RAM Capture	Detection Method	Known audio sequence	Expected outcome	Actual outcome	Byte size	Segments
1	Automatic algorithm	A B C D E	audio	A B C D E	3000	5
2	Automatic algorithm	C A E B D	audio	C A E B D	3000	5
3	Automatic algorithm	D C E A B	audio	D C E A B	3000	5

This set of experiments involved the detection of known audio segments

implanted into the RAM capture to test and develop an algorithm that detects the features of human audio. The inserted audio segments were all detected in the order in which they were inserted into the RAM capture.

2.2 Introduce VoIP Call

The following experiments consisted of using VoIP application Skype to make a VoIP call to the Skype sound test service. The call was then terminated and a RAM capture performed on the computer initiating the VoIP call using X-Ways Forensics (2009). This was then repeated after the lapse of a 24-hour period whereby the laptop on which the RAM capture was performed, was powered down to allow the RAM contents to dissipate.

Audio analysis tool, ESection (2010) was also used as the basis for the starting point in audio signal identification whilst performing VoIP calls. The VoIP calls were initiated and the RAM captured from inside a virtual machine (VM) using VMware (2009). The amount of RAM that is subsequently captured and analysed in the VM may be reduced The ESection software is operated externally to the virtual machine whilst running on the host machine supporting the virtual machine. This prevents the identification of audio within the RAM capture inside the virtual machine from being confused with the audio input at the microphone captured using ESection. Therefore all ESection audio captures saved on the host do not appear in the VM RAM capture

The ESection capture allows us to further develop an algorithm based on the properties of the captured audio, magnitude and symmetry to identify the type of signal that should be searched for within the VM RAM capture as shown below in Figure 10.

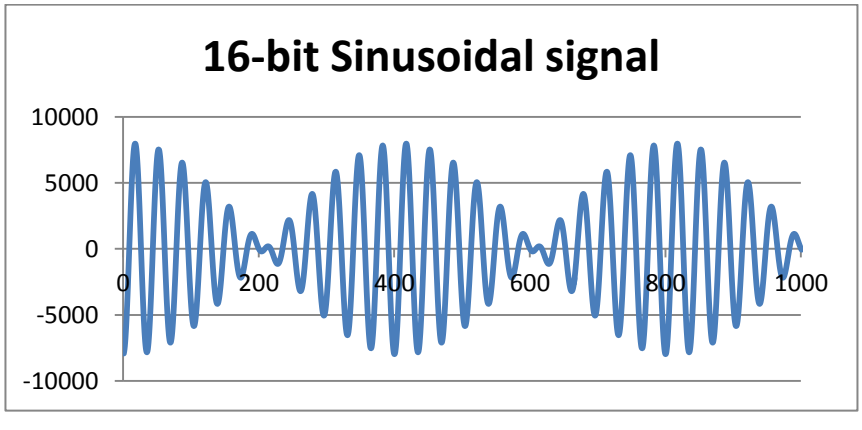

Figure 10 – Initial signed sinusoidal signal (x-axis: No. of samples, y-axis: Amplitude)

Several of the sinusoidal-like signals observed are believed to form part of the VoIP application dialling tones and not human speech. This is later confirmed by frequency domain analysis in Section 3.

The ESection capture allows us to develop an algorithm based on the properties of the captured audio, magnitude and symmetry to identify the type of signal that should be searched for within the VM RAM capture as shown below in Figure 11.

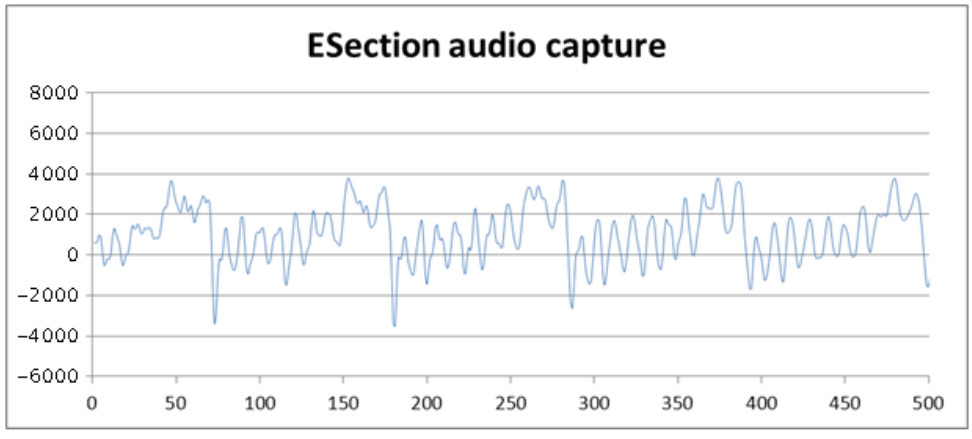

Figure 11 – ESection audio capture (x-axis: No. of samples, y-axis: Amplitude)

This allows one to focus on the specific attributes of human speech within the ESection captured signal in order to implement an algorithm which can automatically search a block of computer memory. By close inspection of the properties of captured human speech such as changes in amplitude and symmetry, one can construct an algorithm that will exclude signals which do not show the typical attributes of digitised human speech such as symmetry and repetition of the waveform.

The use of virtual machines allows a much smaller amount of virtual RAM to be captured e.g. 512MB as opposed to the order of Giga bytes. This may decrease the amount of human speech captured from the VoIP call but the purpose of this research is to demonstrate that suspected audio fragments may be human speech identified from analysis of a memory capture. The individual components of identified audio tend to be typically 4096 bytes in length and as such will require a sequence of these audio fragments to be reconstructed to form one continuous piece of human speech to form playable audio.

Suspected audio samples of digitised human speech are fragmented throughout the physical memory due to virtual address translation by the operating system shown below in Figure 12. The virtual address pages are linked to a page table entry (PTE) highlighted by the dashed line. The PTE contains the mapping from the virtual address to the physical address. This diagram highlights how three consecutive virtual pages of digitised human speech are mapped to three non-consecutive pages in the physical memory (Solomon & Russinovich, 2005).

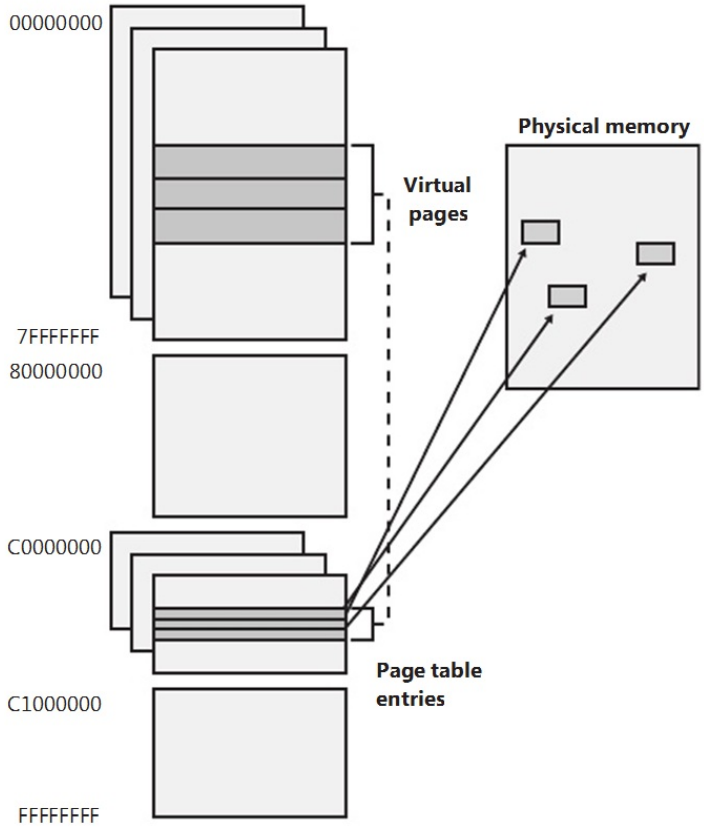

Figure 12 – x86 Virtual address translation

2.3 X-Ways Analysis

The virtual machine consisted of a Windows XP operating system with the Skype VoIP application downloaded within it. A VoIP call to the Skype sound Test Service was made then the call was terminated. X-Ways forensic software, installed within the virtual machine, was used to capture the 512MB of RAM. Whilst the call was being made, ESection audio capture software

was also started on the host to record the audio input.

This experimental setup allowed for the search of RAM as outlined above, in addition to this, the RAM was opened in the X-Ways hexadecimal editor and specific keywords were searched for e.g. Sound Test Service. Figure 13a shows an extract from the hexadecimal editor for search string 'sound test service', providing 237 hits.

266396096	43	48	41	54	20	23	64	61	76	69	64	5F	74	5F	69	72	CHAT #david_t_ir
266396112	77	69	6E	2F	24	65	63	68	6F	31	32	33	3B	62	34	66	win/$echo123;b4f
266396128	32	30	38	62	64	34	63	32	63	37	33	37	63	20	46	52	208bd4c2c737c FR
266396144	49	45	4E	44	4C	59	4E	41	4D	45	20	45	63	68	6F	20	IENDLYNAME Echo
266396160	2F	20	53	6F	75	6E	64	20	54	65	73	74	20	53	65	72	/ Sound Test Ser
266396176	76	69	63	65	BE	F8	48	11	08	F7	B2	08	E9	D2	54	4B	vice¾øH ÷² éÒTK
266396192	AE	21	39	E4	1F	0F	A6	D2	76	B2	40	E4	54	37	C6	78	®!9ä ¦Òv²@äT7Æx
266396208	E0	A3	81	A7	C2	65	29	38	9D	AD	76	C7	80	7D	43	5E	à£ §Âe)8 -vÇ€}C^
266396224	D7	30	8C	4B	51	98	3A	B4	82	A2	15	26	0D	85	2C	F6	×0ŒKQ˜:´‚¢ & …,ö
266396240	8F	B5	72	FE	DE	87	C9	73	A5	75	55	30	06	C7	06	A4	µrþÞ‡És¥uU0 Ç ¤
266396256	EF	A9	4C	76	94	91	C4	F3	24	02	E5	34	90	9B	3D	A8	ï©Lv”‘Äó$ å4 ›=¨
266396272	F8	A8	C5	82	58	B3	CF	58	30	3F	5B	77	E0	6C	9F	CB	ø¨Å‚X³ÏX0?[wàlŸ

Figure 13a – X-Ways search hit for 'sound test service'

The bytes immediately following the 'sound test service' string were plotted and produced an audio signal extract as shown in Figure 13b.

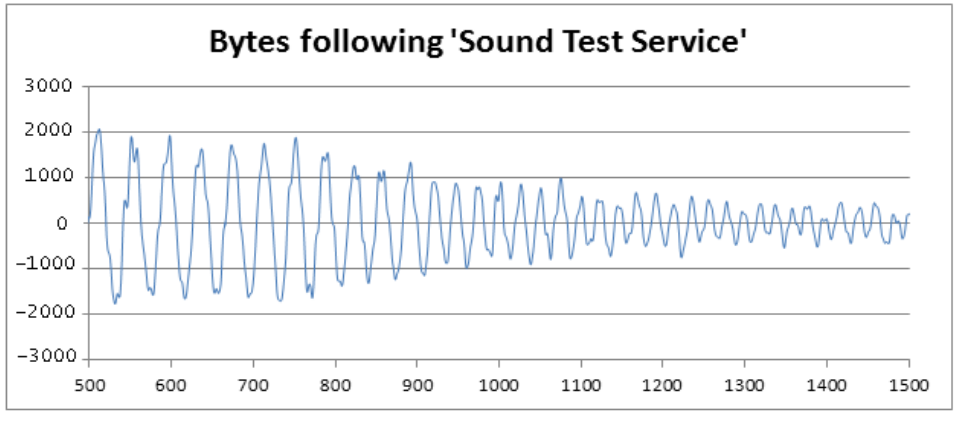

Figure 13b – 4096 bytes immediately following search string (x-axis: No. of samples, y-axis: Amplitude)

Using known Skype caller id as a search string also allows the call information attributed to the VoIP call to be extracted as shown below in Figure 13c, such as the caller identities in raw xml format, call initiator and timestamps. This is using X-Ways hex editor to view the captured RAM and perform string searches. Not only does the software tool search for audio fragments, it can retrieve information related to the VoIP call.

```
"RT INTO Messages

(id,is_permanent,convo_id,chatname,author,from_dispname,gu
id,dialog_partner,timestamp,type,sending_status,body_xml,i
dentities,reason,participant_count,chatmsg_type,chatmsg_st
atus,body_is_rawxml,pk_id,call_guid)

VALUES

(164,0,30,'#david_t_irwin/$echo123;b4f208bd4c2c737c',
'david_t_irwin','davidirwin',x'61b6a47d0ab0894bca8bdb65
8307051d9bb9f2e42e126198a1aaeca2f68658fa','echo123',130590
1540,30,2,

'<partlist alt="">

<part
identity="david_t_irwin"><name>davidirwin</name></part>

<part identity="echo123"><name>Echo / Sound Test
Service</name></part>

</partlist>',

'echo123','',2,18,2,1,1160776592,'f1676bd45b6963ef2522d976d59b361
9');"
```

Figure 13c – X-Ways extract of Skype VoIP call setup to Sound Test Service

The use of a programmed search algorithm is more efficient than a visible search. A number of possible segments of human speech have been identified based on amplitude and symmetry and displayed on a single graph for the user to visually inspect for the difference between a pure or amplitude-modulated sinusoidal trace and that typical of human speech. This reduces the search space of a RAM capture to a single point of investigative analysis but none the less still requires human intervention in the form of visual inspection. Table 3 indicates the number of suspect audio fragments detected from each VoIP call.

Table 3 – VoIP call to Skype Test Call Centre made from inside virtual machine.

RAM Capture	Detection Method	VoIP Call	Expected outcome	Actual outcome	Byte size	Segments
1	Automatic algorithm	Skype	dialing tone, automated voice & caller's voice	sinusoids & suspect human voice	4096	48
2	Automatic algorithm	Skype			4096	46
3	Automatic algorithm	Skype			4096	53

It would not be an unreasonable question to ask 'why use the capture of memory for the purpose of obtaining audio, why not just capture the microphone input or speaker output directly'?

The capture of computer memory allows information to be retrieved including specific information relating to the use of VoIP applications, such as call identifiers, user names, date and timestamps, the captured information may subsequently be used to testify to the authenticity of such a call having been made.

The test of a known audio dissected into five segments of equal audio length and inserted into the RAM capture as outlined in 2.1 yielded all five segments being detected. Five segments are easily re-assembled into its original form visually. However, the amount of audio segments recovered from the VoIP call are significant and potentially three distinct sources, the VoIP application dialling tone, the automated answering of the call to the Skype test sound service and the caller.

Dialling tones are easily identified using a signal processing technique called frequency domain analysis. A brief introduction to digital signal processing is discussed in section three however the removal of dialling tone still requires two separate call stream to be identified. This requires an additional algorithm to interrogate the start and ending bytes of each segment retrieved and attempt to find another once with which matches to form two separate streams of continuous audio.

Similarly, one may ask the question "why focus on a RAM capture, without extending the search to a hard disk(s) as the contents of RAM are continually being swapped out from virtual memory to physical memory stored on the hard disk(s). The answer is simple; it wasn't within the remit of the Law Enforcement Agency to require analysis of anything other than a perceived RAM capture represented as a file with a '.txt' extension. However, the analysis techniques described in this research are easily extended to include analysis of the hard disk(s) and information and files related to the transfer of virtual memory pages to a physical location.

3. INTRODUCTION TO DIGITAL SIGNAL PROCESSING

Although information stored in RAM is paged in Windows operating systems, the information within each page e.g. a fragment of human speech is ordered sequentially in time. Therefore all research until now has taken place in the time domain with graphical plots of signal samples on the y-axis appear as how they are digitised in time in memory. The main research theme is to demonstrate the ongoing development of an automatic audio search functionality to identify the fragments of human speech.

Having identified a series of signal components from the RAM capture which exhibit a symmetrical pattern (Figure 13b) based on simple characteristics of

human speech displayed in Figures 9 and 11, one can further remove composite sinusoidal signals unrelated to human speech by processing the sampling values through a Fast Fourier Transform (FFT) and viewing the result in the frequency domain.

A FFT is itself an algorithm for calculating the Discrete Fourier Transform (DFT) which decomposes a sequence of values, in this case, amplitudes of suspected audio fragments into components of different frequencies.

For the purpose of drawing comparison, how do other files such as word documents or excel worksheets appear when graphically displayed in the time domain. The technique employed displaying the file on 50 graphs where each graph displays 4096 byes. Each graph has a different starting point within the file, e.g. graph 1 (top row, 1st column) starts at 0 bytes and displays the first 4096 bytes and graph 2 (top row, 2nd column) starts at 1/50th of the file length and displays the next 4096 bytes. This process is repeated for the fifty graphs, and with one click of a button, all graphs advance 4096 bytes. The word document shown in Figure 14 was visually inspected and contained no similarities to composite sinusoids or audio fragments.

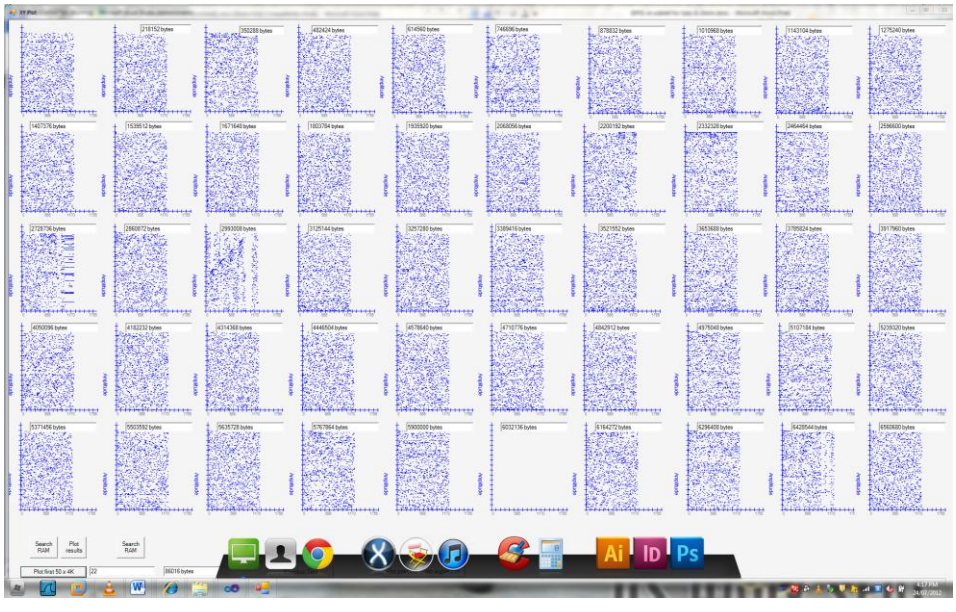

Figure 14 – Visual display of byte values for a word document.

Based on the above visual inspection, no audio-like byte segments are detected and subsequently would not be passed to the frequency domain for analysis. Note that the document tested and one you are reading are the same document.

Similarly for an excel document, the document tested was the one containing

the power spectrum plots and sinusoids and suspected audio fragments. Once again a visual inspection of the file contents displayed as a graphical plot of the bytes making up the file revealed no audio like fragments.

The visual inspection of graphical memory displayed in Figure 14 was laborious and the first step in searching for audio like fragments to determine their properties for automatic algorithm development.

3.1 Power Spectrum Analysis in the Frequency Domain

For the purpose of this research, the programming code for the FFT has been extracted (and manipulated to suit) from Audacity (2011), a free cross-platform audio editor developed by a team of software developers, translators, documentation writers. The Audacity application contains the function "Plot Spectrum", which analyses a section of audio and converts it to a graph of frequencies against amplitudes using the FFT algorithm to provide a value for each narrow band of frequencies that represents how much of those frequencies are present. This research is based upon the code contained in the Audacity "Plot Spectrum" function to analyse the portions of RAM capture, which are suspected of being fragments of audio.

The term, aliasing, is used to describe the effect of different signals becoming indistinguishable from each other. To counter the effect of aliasing, the FFT is used in conjunction with a Hann Window function, to process the suspected audio fragments in the time domain. The code for the Hann Window function is also extracted from Audacity, which allows a smaller subset of the suspected audio to be analysed, just as the name infers, applying an overlapping window, which traverses the original FFT input to produce the corresponding frequency domain output.

The composite sinusoidal signal shown in Figure 10 produces a frequency domain plot as shown in Figure 15, highlighting its composition from more than one frequency. A periodic sinusoid would display as a single frequency component in the frequency domain.

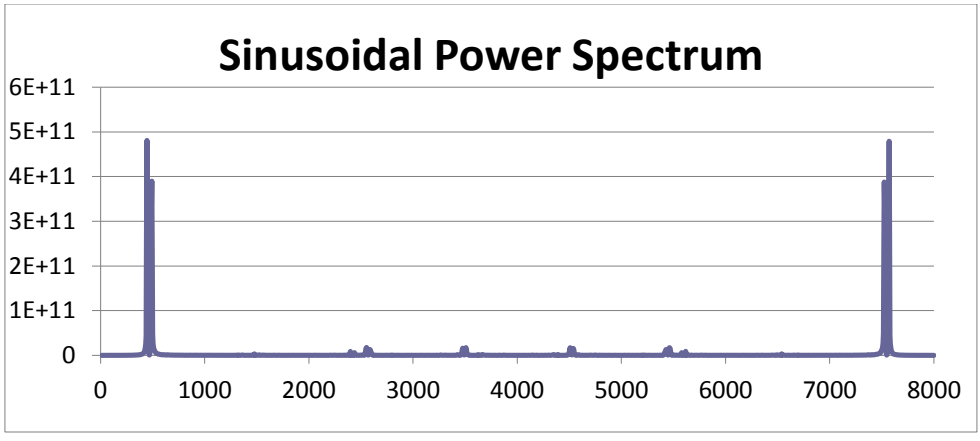

Figure 15 - Frequency domain analysis (x-axis: Frequency (Hz), y-axis: Raw amplitude)

The FFT transforms the time domain signal into a frequency domain representation of that signal. It generates a description of the distribution of the energy in the signal as a function of frequency. The vocal range of human speech varies from approximately 70 Hz to 7 KHz. However, most of the information conveyed in human speech does not exceed 4 KHz.

The modelling of human speech and the pronunciation of vowels, shown below in Figure 16 indicates that the majority of the energy is concentrated below 4 KHz.

Three different vocal tract shapes are shown corresponding, from top to bottom, to the vowels "ah" (/a/), "ee" (/i/), and "oo" (/u/). Plotted in the same graph for each tract shape is the spectrum. Note all three vowels have differing spectra due to the different vocal tract shapes. A variety of methods are being used to explore this mapping (Kawato, 1989; Saltzman, Munhall, 1989; Jordan, 1990).

Nyquist's theorem states that the sampling frequency must be at least twice as high as the highest input frequency (4 KHz) thus a sampling frequency of 8 KHz will allow the digitised voice to correctly represent the original signal. A pure sinusoid in the frequency domain will appear as a single spike, whereas the signal shown has 2 spikes at similar frequencies (440 Hz and 485 Hz) and also have a mirror image of itself (7515 Hz and 7560 Hz), with even symmetry around the centre point of half the sampling frequency, 4 KHz.

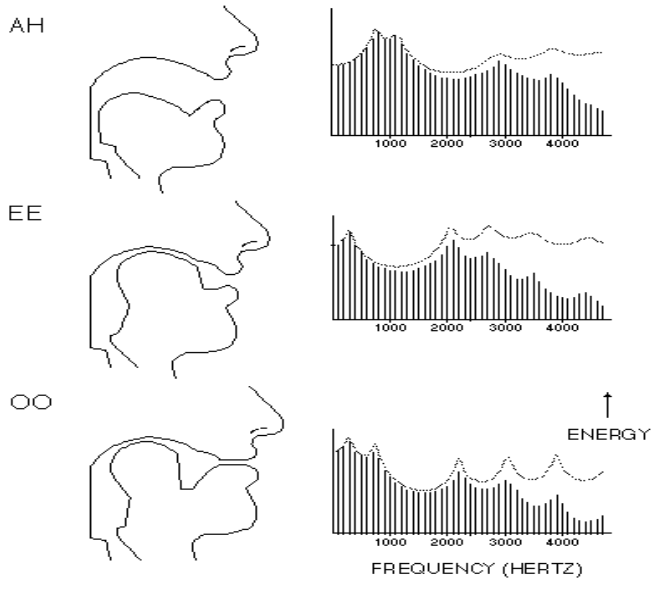

Figure 16 – Energy distribution versus frequency of human speech.

The phenomenon of reflection around the point of half the sampling frequency for periodic signals is counteracted by the software via automatically removing the part of the spectrum for frequencies exceeding half of the sampling frequency. The resulting spectrum can now be compared against frequency domain plots of sections of human speech where most of the energy is concentrated below 4 KHz e.g. as shown in Figure 17. Signals that do not fall within the range of pattern representing human speech can therefore be removed.

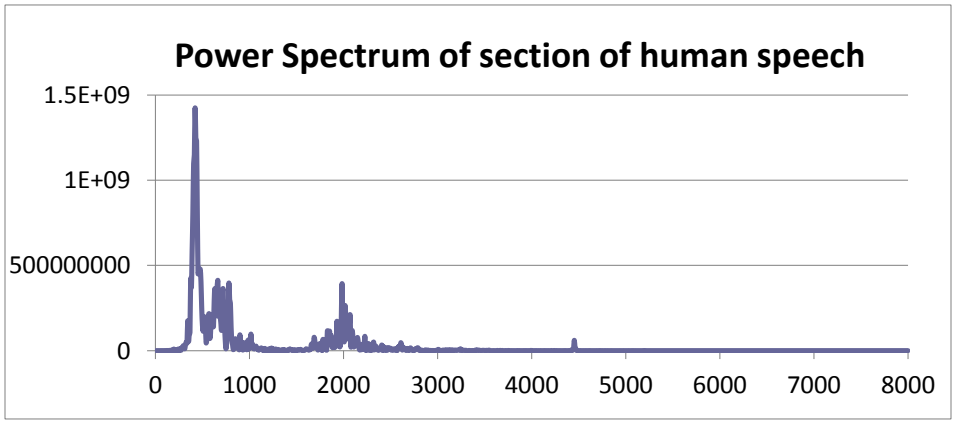

Figure 17 - Frequency domain analysis of speech (x-axis: Frequency (Hz), y-axis: Raw amplitude)

The resulting collection of signals that remain after the removal of non-speech signals based on both time domain and frequency domain analysis are believed to be fragments of human speech which have been digitised after processing by the sound card.

4. CONCLUSIONS & FUTURE WORK

The techniques described in this research have been applied only to RAM captures. The RAM captures discussed in experimental setup have been forensically acquired using X-Ways Forensics. The research has aimed to introduce novel techniques for the analysis of physical memory such as graphical visualisation (albeit time consuming) and the development of automatic algorithms to identify possible audio fragments based on their symmetrical appearance.

The use of visual inspection aimed to review the full RAM capture visually to avoid developing an algorithm that would detect false positives and omit false negatives. This was aided by the introduction of known segments into the RAM capture out with a VoIP call to test the detection properties of the algorithms.

The use of digital signal processing techniques to view possible audio fragments in the frequency domain is also novel. However, this research is ongoing to develop further algorithms that will inspect the leading and trailing edges of the suspect audio fragments to see if they can be joined together to identify different call streams and form continuous segments of audio.

Once segments of continuous audio have been reconstructed, it is anticipated that they would provide a high degree of probability that a particular individual has had access to a specific computer and made a VoIP call by matching their voice against the recovered audio from memory.

5. ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the Australian Research Council in this work via Linkage Grant LP0989890 and additional scholarship contributions from the Australian Federal Police.

REFERENCES

Audacity (2011, June 19). Audacity application downloaded. Retrieved from http://http://audacity/sourceforge.net

Beebe, N.L., Clark, J.G., Deitrich, G.B., Ko, M.S., & Ko, D. (2011, November). Post-retrieval search hit clustering to improve information retrieval effectiveness: Two digital forensics case studies. *Decision Support Systems, 51*(4), 732-744.

Carrier, B. (2003, Winter). Defining Digital Forensic Examination and

Analysis Tools Using Abstraction Layers. *The International Journal of Digital Evidence, 1*(4). Retrieved from http://www.digital-evidence.org/papers/ijde_define.pdf

Carvey, H. (2007). *Windows Forensic Analysis DVD Toolkit*. Burlington, MA: Syngress Publishing.

Casey, E., Gordon, G., & Leeson, L. (2005, February). Origins and Progress. *Digital Investigation, 2*(1), 1-2.

Civie, V, & Civie, R. (1998). Future Technologies from Trends in Computer Forensic Science. Presented at the Forensic Science in Trial - Seventh Report of Sessions, London: House of Commons.

ESection (2010, November 5). ESection application downloaded. Retrieved from http://www.phon.ucl.ac.uk/resources/sfs/esection

European Telecommunications Standards Institute (ETSI). (2001). *Telecommunication Security - Lawful Interception - Issues on IP Interception*. TR 101 944 V1.1.2.

Garofolo, J. S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., Dahlgren, N.L., & Zue, V. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus, Linguistics Data Consortium.

Hibishi, H., Vidor, T., & Cranor, L. (2011). Usability of Forensics Tools: A User Study. In *Proceedings of the 2011 Sixth International Conference on IT Security Incident Management and IT Forensics (IMF)*, pp. 81-91.

Hornig, C. (1984). *A Standard for the Transmission of IP Datagrams over Ethernet Networks*. IETF RFC 894.

Jordan, M. (1990). Motor learning and degrees of freedom problem. In M. Jeannerod (Ed.), *Attention and Performance XIII*, pp. 221-229 (Hillsdale, NJ: Erlbaum).

Kawato, M. (1989). Motor theory of speech perception. In *Proceedings of the 8th Symposium on Future Electron Devices*, pp. 141-150.

Keller, E. (1994). *Fundamentals of Speech Synthesis and Speech Recognition*. Chichester: John Wiley & Sons.

McKemmish, R. (June 1999). *What is Forensic Computing?* The Australian Institute of Criminology.

Mohay, G. (2005). Technical Challenges and Directions for Digital Forensics. In *Proceedings of the First International Workshop on Systematic Approaches*

to Digital Forensic Engineering (SADFE), Washington, D.C.

Pirani, G. (1990). *Advanced Algorithms and Architectures for Speech Understanding*. London: Springer-Verlag.

Postel, J. (1980). *User Datagram Protocol*. IETF RFC 768.

Postel, J. (1981). *Internet Protocol*. IETF RFC 791.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., & Schooler, E. (2002). *SIP: Session Initiation Protocol*. IETF RFC 3261.

Saltzman, E.L., & Munhall, K.G. (1989). A dynamic approach to gestural patterning in speech production. *Ecological Psychology*, *1*(4), 333-382.

Schatz, B. (2007). *Digital Evidence: Representation and Assurance*. Information Security Institute, Queensland University of Technology.

Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (2003). *RTP: A Transport Protocol for Real-Time Applications*. IETF RFC 3550.

Solomon, D., & Russinovich, M. (2005). *Microsoft Windows Internals*, 4th ed. Seattle, WA: Microsoft Press.

Skype. (2009, August 22). Skype application downloaded. Retrieved from http://www.skype.com

The Open Group. (2010). DD. Retrieved from http://pubs.opengroup.org/onlinepubs/009604499 /utilities/dd.html

VmWare. (2009, July 15). VM Workstation application downloaded. Retrieved from http://www.vmware.com

X-Ways Forensics. (2009, July 18). X-Ways Forensics application downloaded. Retrieved from http:// www.x-ways.net

Yasinsac, A, Manzano, Y. (June 2001). "Policies to enhance computer and network forensics," IEEE Workshop on Information Assurance and Security.