University of South Australia

School of Computer and Information Science

Mawson Lakes

Forensic Analysis of Mobile Phones

A thesis submitted for the Bachelor of Computer and Information Science (Honours) Degree

October 2005

Paul McCarthy

Supervisor: Dr Jill Slay

Declaration

I declare that the information contained within this thesis is my own, and except where noted, does not contain any work which has been previously written or published.

Paul McCarthy Date

Abstract

By the end of 2005, it is expected that more than 90% of Australians will own a mobile phone (AMTA 2005). With the exceedingly large amount of data held in an individual's pocket, the potential for mobile phones to contain information which can be used as evidence within a criminal or civil context continually grows. Call logs, SMS messages, calendar entries, images and audio and video recordings may all play a part in a criminal investigation.

When dealing with a digital device, such as a mobile phone, computer or otherwise, the methods used to acquire data stored in the device must have as little impact on the device's memory as possible. This is important, to ensure that the integrity of the acquired data can be verified.

As such, the methods used to acquire data from a mobile phone will play an important part in an investigation which requires evidence to be extracted from a mobile phone. The investigator must be sure that the information acquired actually reflects that which is stored in the phone, and that the procedures used do not adversely affect the integrity of the information, or any other data stored in the device.

There is currently no accepted forensic procedure for acquiring data from mobile phones. There are a wide range of software applications which claim to extract information from phones in a forensically sound manner; that is, without making any changes to the phone's memory. However, such claims cannot be verified, as these applications treat phones as a black box, using simple command – response protocols to acquire the data.

This thesis attempts to provide an overview of the methods commonly used to acquire data from mobile phones in a forensic manner. Limitations and issues inherent in software based data acquisition are discussed, as is the legal admissibility of information acquired using these methods. By doing so, the need to verify the methods currently in use is highlighted.

Table of Contents

1 Glossary 6
2 Introduction 9
2.1 Motivation 11
2.1.1 Traditional crimes and mobile phones 12
2.1.2 Electronic crimes and mobile phones 13
2.2 The problem at hand 14
2.3 Thesis Aim 16
2.4 Method of research 16
3 Literature Review 18
3.1 Data stored in mobile phones 18
3.1.1 Data stored in SIM cards 18
3.1.2 Data stored in phone memory 19
3.1.3 Data stored by the service provider 20
3.2 Extracting data from mobile phones 20
3.2.1 Extracting data from SIM cards 22
3.2.2 Extracting data from phone memory 23
3.2.2.1 Manufacturer and third party software 24
3.2.2.2 Forensic mobile phone software 25
3.2.2.3 PC connectivity SDKs 26
3.2.2.4 AT commands 26
3.2.2.5 OBEX 27
3.2.2.6 Nokia FBUS 28
3.2.2.7 JTAG 29
3.3 Legislation regarding electronic evidence 31
4 Results 33
4.1 Methods of acquiring data 33
4.1.1 AT Commands 33
4.1.2 OBEX 36
4.1.3 Nokia FBUS 39
4.2 Review of existing applications 45
4.3 Research application 48
5 Discussion 53
5.1 Limitations in software based data acquisition 53
5.2 Admissibility of evidence acquired from a mobile phone 55
5.3 Feasibility of a universal software application 57
6 Conclusion 59
6.1 Summary 59
6.2 Future directions 60
7 References 61
Appendix A: Email correspondence 65
Appendix B: Research application UML class diagram 67
Appendix C: OBEX over FBUS session initialisation 68
Appendix D: List of known Nokia FBUS commands 71

Table of Figures

Figure 1: *Mobile phone penetration in Australia 2001-2005* 9

Figure 2: *Software based acquisition of data stored in a mobile phone* 10

Figure 3: *Generic data stored in a Nokia 6225* 20

Figure 4: *Nokia 3220, Nokia 6225 and Sony-Ericsson f500i* 25

Figure 5: *Nokia 3220 JTAG interface* 30

Figure 6: *Nokia 6225 JTAG interface* 30

Figure 7: *Nokia 3315 JTAG interface* 30

Figure 8: *Sony-Ericsson f500i JTAG interface* 30

Figure 9: *AT command communication* 34

Figure 10: *Phone denying access to information over AT commands* 35

Figure 11: *SMS message status codes* 35

Figure 12: *SMS status changing in reaction to AT commands* 36

Figure 13: *OBEX communication* 37

Figure 14: *OBEX packet structure* 37

Figure 15: *OBEX session initialisation* 38

Figure 16: *OBEX XML folder listing* 39

Figure 17: *Phone denying access to a file over OBEX* 39

Figure 18: *FBUS communication* 40

Figure 19: *FBUS packet structure* 41

Figure 20: *Nokia FBUS session initialisation* 42

Figure 21: *FBUS received message time representation* 43

Figure 22: *FBUS sent message time representation* 43

Figure 23: *FBUS time retrieval command* 44

Figure 24: *Phone denying FBUS session initialisation* 44

Figure 25: *SMS status code changing in reaction to FBUS commands* 45

Figure 26: *Research application structure* 49

1 Glossary

3G (3rd Generation) A generic term referring to any of the recent wireless communication networks, providing high speed data transmission services such as video calling and broadband internet access.

ASCII (American Standard Code for Information Interchange) A standard method of storing text in electronic form. ASCII uses seven bits to store characters, giving a total of 128 different characters in the set. The eighth bit of each character byte can be used for parity (error) checking, ignored, or used to extend the standard character set.

AT Commands Also known as Hayes commands, are a set of commands which were originally developed for controlling modems. The ‘AT’ refers to the process where two devices determine the correct speed at which to communicate with each other.

Bluetooth An ad-hoc wireless communication standard built into the majority of new mobile phones. In its most common form, Bluetooth provides direct communication between devices to a range of approximately 10 metres.

CDMA (Code Division Multiple Access) A 2G wireless communication network standard, originally implemented by telecommunications service provider Qualcomm.

ESN (Electronic Serial Number) A unique identifier assigned to every ME within a CDMA network.

FBUS Nokia’s proprietary protocol which enables a PC to access the data stored in a Nokia mobile phone. FBUS also provides the ability to use the phone’s network functionality, for example, to send and receive SMS messages.

GSM (Global System for Mobile Communications) A 2G wireless communication network standard, originally developed in Europe to provide a single standard across the entire continent.

IMEI (International Mobile Equipment Identity) A unique identifier assigned to every ME within a GSM network.

IMSI (International Mobile Subscriber Identity) A unique identifier assigned to every SIM card within a GSM network.

IRMC (Infrared Mobile Communications) A synchronization protocol, originally designed for use over Infrared, which enables information stored in a mobile device, such as calendar entries and contacts, to be synchronized with that stored in a PC application such as Microsoft Outlook.

JTAG (Joint Test Action Group) An IEEE standard specifying an interface which can be used to test the hardware components which form an electronic device.

ME (Mobile Equipment) A term used to refer to a mobile device (i.e. a mobile phone) operating in a wireless communication network.

MMS (Multimedia Messaging System) A messaging service similar to SMS which enables messages comprising of images, audio and/or video to be sent over a wireless communication network.

OBEX (Object Exchange) A transport protocol, originally developed for use over Infrared, which enables generic transport of data over a communication medium.

PDU (Protocol Description/Data Unit) A standard used by mobile phones in a GSM network for storing and sending SMS messages.

PIN (Personal Identification Number) A number which must be given to a mobile phone / SIM card before it will allow access to its features and/or connect to the network.

PUK (Personal Unblocking Key) A number which unlocks a SIM card in the event that the incorrect SIM PIN is entered three times in succession. The PUK is stored by the service provider.

SD (Secure Digital) A form of removable storage, commonly used in mobile phones, cameras and MP3 players.

SIM (Subscriber Identity Module) A smartcard which identifies subscribers within a GSM network. The SIM card is placed within a GSM mobile phone, and is required to join the network.

SMS (Short Message Service) A messaging service, originally implemented for use in GSM networks, which enables short text messages to be sent between subscribers.

SyncML (Synchronization Markup Language) A synchronization protocol which is replacing IRMC as the standard for phone – PC synchronization.

TDMA (Time Division Multiple Access) A 2G wireless communication network standard used in GSM networks.

WCDMA (Wideband Code Division Multiple Access) A 3G wireless communication network standard which uses the same techniques as CDMA to transmit information, at much higher speeds. UMTS (3G GSM) is based on WCDMA.

2 Introduction

Acquiring data from a mobile phone in a forensic manner is an important issue. Information acquired from mobile phones is increasingly required as evidence in criminal investigations. Figure 1 shows mobile phone penetration in Australia from 2001 to 2005 (AMTA 2004, 2005). These statistics highlight the continual growth in Australia, of people who own a mobile phone.

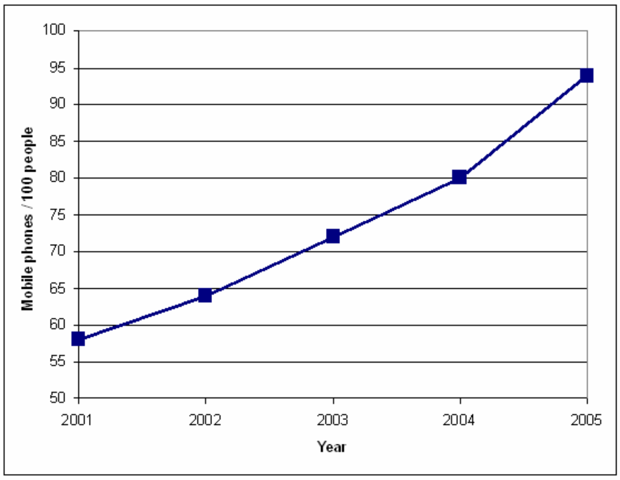


Figure 1: *Mobile phone penetration in Australia 2001-2005 (*Australian Mobile Telecommunications Association (AMTA) 2005 *Australian Mobile Telecommunications Industry Economic Significance*, online accessed 6th August 2005 http://www.amta.org.au/amta/site/amta/downloads/pdfs_2005/AMTA%20Industry%20Report%20%20Exec%20Summary%202005.pdf), (Australian Mobile Telecommunications Association (AMTA) 2004 *Industry Statistics Snapshot*, online accessed 20th May 2005, http://www.amta.org.au/default.asp?page=327).

A mobile phone can potentially contain a large amount of information related to the user's actions, determined by their communication patterns, and information such as images, video and audio recordings. As such, the information stored in a mobile phone may be important in proving or disproving theories and allegations.

The Australian state and federal legislation treats a mobile phone in the same manner as a computer, or any other electronic device. Before information produced by such a device can be admitted as evidence, it must be shown that the device is functioning correctly, and the procedures used to obtain the information do not adversely affect the validity of the information.

Hence, the methods in which information is obtained from a mobile phone may have a direct effect on whether that information will be admissible as evidence. If a certain method can be shown to alter data on the phone, the integrity of that data may be questioned, and even shown to be inaccurate. The desired situation would occur when a method can be proven to acquire data without making any changes to the phone's memory; information acquired using such a method will be admissible as evidence.

There are a number of different methods of acquiring information from a mobile phone. The most convenient, however, is to use a software application running on a desktop computer to send commands to the phone, the response to which contains information stored in the phone's memory. Such an application communicates with some form of software or hardware contained in the phone, which retrieves the data on behalf of the desktop application. This process is described graphically in Figure 2.

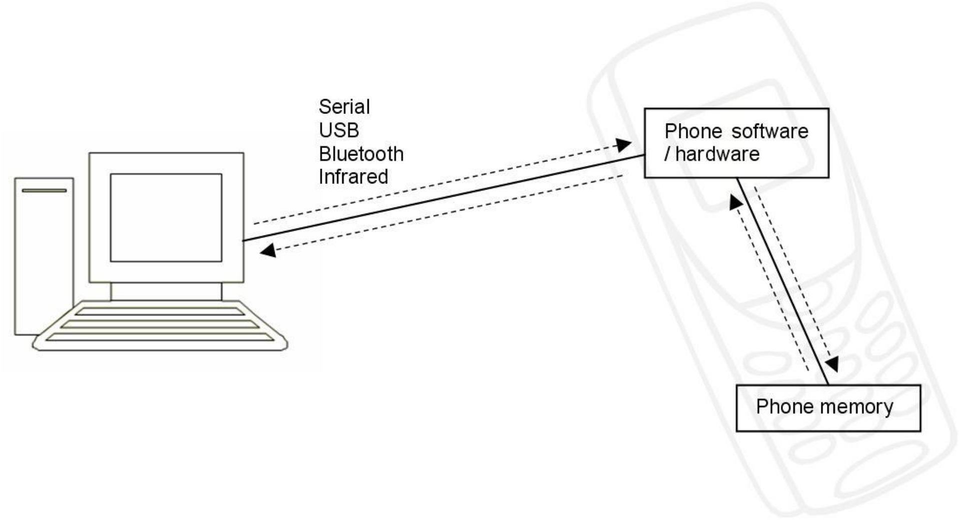


Figure 2: *Software based acquisition of data stored in a mobile phone*

There is obviously a layer of indirection in this procedure. The desktop application used by an investigator does not directly access the phone's memory; rather, it relies on the logic stored in the phone to retrieve the information and pass it back along the communication channel. There is a problem when considering this situation from a forensic perspective; both the desktop application and the investigator are assuming that the phone's logic is not making any changes to other areas of the phone's memory. However, this assumption cannot be verified without the source code and circuit schematics of the phone's software and hardware, which are rarely, if ever publicly available.

This thesis aims to provide an overview of the methods commonly used to acquire information from mobile phones, and forensic issues which are raised by using such methods. A software application was developed for use as a research tool, using common software based methods of acquiring information from mobile phones. This application was used to analyse both the effects that the methods have on phones, and the amount of information which is actually obtainable using the methods. Legislation as it relates to evidence obtained from mobile phones is discussed, as is the admissibility of information acquired from a mobile phone using the software methods described.

Chapter 2 provides some examples of the ways in which mobile phones can be involved in crime, as well as a more detailed discussion of the problems involved in acquiring data from a mobile phone. Additionally the thesis aims, and the methods used to undertake the research are clarified. Chapter 3 provides an overview of the existing academic literature in the field, the various methods of acquiring data from mobile phones, and the Australian legislature as it relates to evidence acquired from mobile phones. Chapter 4 discusses the software application which was developed as part of this thesis, along with more detailed descriptions of the software methods used by the application to acquire data from mobile phones. A discussion of the admissibility of information, which is acquired using these methods, as evidence is provided in Chapter 5, along with the potential problems which could be faced when using acquired information to build a case. Finally, Chapter 6 concludes and gives possible directions for future research.

2.1 Motivation

The examples discussed in the following sections indicate that mobile phone technology has become of great importance to law enforcement, in tracking illegal activities, and in providing evidence for convictions and acquittals. The information found in mobile phones has great potential for use as evidence in criminal investigations.

2.1.1 Traditional crimes and mobile phones

The following examples illustrate some of the possible ways in which a mobile phone can be involved in criminal activities.

- The Australian Institute of Criminology found that mobile phones are the most common form of communication for people purchasing heroin, methamphetamines and cocaine (Makkai, Milner & Mouzos 2004, pg. 6).

- Mobile phones are common targets for thieves. As of August 2002, around one hundred thousand mobile phones were stolen in Australia each year (SA Police 2002).

- A survey of prosecutor offices in America, undertaken by the American Prosecutors Research Institute, found that telecommunication service theft (i.e. mobile phone theft, SIM cloning, etc.) make up a significant portion of telecommunications fraud, with 26 percent of respondents having to investigate some form of service theft (Fanflik et. al. 2004, pg. 7).

- The relatively large storage space of modern phones makes them a useful tool for data theft. An employee could steal sensitive corporate information by uploading it onto their phone (Black 2005).

- Ekblom and Tilley (2000, pg. 386) discuss some hypothetical examples of how a mobile phone can be involved in criminal situations. For example, a criminal could call a security guard, to provide a diversion while his/her accomplices commit the crime.

- Another example is the use of a mobile phone for harassment. A case held in the District Court of South Australia involved the accused sending threatening SMS messages and making abusive phone calls to the victim. The call records, and SMS messages between both parties played a significant part in the case (AUSTLII 2003a).

- A case held in the Court of Appeal of New Zealand concerning a jewellery store robbery, found that the perpetrators used mobile phones extensively to organise and coordinate the crime (AUSTLII 2000).

- A more disturbing incident occurred in South Korea, late 2004, when subscriber information was stolen from a mobile phone carrier, and used to create duplicate phones. These phones were then used to purchase goods from the internet, using the original subscribers' accounts (Jae-hyun 2004). Users are generally placing a high level of trust in their mobile phones (Harkin 2003), which is a serious issue, as someone who steals a mobile phone could potentially pose as the legitimate owner of the phone (Masnick 2004) without having to find further personal details.

2.1.2 Electronic crimes and mobile phones

The global integration and interoperability of society's communication networks (i.e. the internet, public switched telephone networks, cellular networks etc.) means that anyone with a laptop and / or a modern mobile phone has the potential to commit a crime, without any limitations on mobility or time (Etter 2001, pg. 3).

In addition to this, the fast paced development of the mobile phone market ensures that mobile phones are continually becoming more powerful, and offering increased functionality (WAPForum 2000). Therefore the ability to use a mobile phone for malicious purposes is becoming more and more apparent.

The ability to install and execute applications, in the same way as on a traditional computer, is appearing in new phones. For example, the majority of new Nokia phones have the ability to download and execute Java applications (Nokia 2005a). Applications which can be used as part of a computer attack will soon be able to run on a mobile phone.

The increasing complexity of mobile phones also means that they are becoming susceptible to attacks through various software and protocol vulnerabilities. For example, a vulnerability in the Nokia 6210 handset allows an attacker to crash the phone via a business card SMS (@Stake 2003).

Viruses and worms targeting mobile phones are also appearing. A report released by Symantec describes a worm, SymbOS.CommWarrior.A, which targets phones using the Symbian 60 operating system, replicating through MMS messages and Bluetooth, and potentially crashing the phone (Symantec 2005).

2.2 The problem at hand

As discussed in Section 2.1, information found in mobile phones has great potential for use as evidence. However, there are no accepted standards to store or acquire this information. This is due to the wide variety of mobile phone hardware and software designs. A data cable which works with one phone model will rarely work with another, even if both phones are made by the same manufacturer. The same problem occurs with phone software interfaces. Each phone model must be treated differently. Despite this, a software application which could extract information from any mobile phone in a forensically acceptable manner would be an essential element of an investigator's toolkit.

However, the ability to extract that information in a manner that will not significantly change the mobile phone's memory is another obstacle to the development of such an application. There are applications available which claim to do so (described in Section 4.2), however these applications do not directly access the memory; rather, they use commands provided by the phone's software and / or hardware interfaces, and as such are placing a significant amount of trust in the phone software.

Another problem arises concerning the different network technologies in use today. The mobile networks in use today are based on a range of different technologies, including CDMA, TDMA, and 3G technologies such as WCDMA, and handsets based on the different technologies must conform to different specifications. An application designed for use with a GSM phone, would therefore not be expected to work with a CDMA phone.

Taking into account the problems discussed above, and the arguments made in the literature review, the most feasible method of forensic analysis of mobile phones is via a software application. A 'universal' application, which works with any mobile phone, is simply impossible, due to the different standards used by manufacturers, and the different mobile networking technologies in use.

The GSM AT commands (described in Section 4.1.1) ensure that basic information is accessible from any GSM mobile phone. Therefore these commands could be used as a basis for the application. This has the advantage that the code to execute AT commands for GSM phones will only need to be written once, but will work with every GSM phone. The same concept can be applied to other protocols, such as OBEX.

Many AT commands are optional, and many manufacturers implement their own commands or protocols. Therefore most information will not be obtainable through a standard set of commands. Obtaining this information requires each phone to be considered individually, using the extended command sets particular to that model.

This hybrid approach will have some of the benefits of a ‘universal’ application, as certain information will be available from any phone. However, support for a particular phone will need to be added to obtain all of the information from that phone.

Actually acquiring the information from a mobile phone is not the main problem. Using the phone’s interface to acquire data is essentially treating the phone like a black box. A command is sent, and a response is received; how the phone interprets the command, and forms its response is unknown, and cannot be determined. This has the effect that an application using the methods discussed cannot claim to be able to acquire the information stored in a phone in a forensically sound manner.

Despite this, law enforcement agencies require quick and easy methods, without the need for specialist knowledge, to acquire information stored in mobile phones. A software application is the best way to meet these requirements. Therefore, the methods used by such an application must be verified to ensure that they perform as promised, and do not make any inadvertent changes to the phone’s memory. As already mentioned, however, this would require access to proprietary information regarding the phone's inner workings.

2.3 Thesis Aim

The aim of this thesis has changed over the course of the year. Originally, the thesis was based around the development of an application for acquisition of data from mobile phones. However, this has been accomplished by several companies in a commercial setting, and as such would have been redundant work. Of what little academic work is available, no research has attempted to assess the forensic soundness of the underlying methods used to acquire the data. Hence, the focus of the thesis has shifted from the development of the application to an analysis of the methods used by such an application, and the legal admissibility of information acquired by such methods as evidence in a court of law.

More concisely, the aims of this thesis are:

- To provide an overview and analysis of the methods commonly used to forensically acquire data from mobile phones.
- To determine the limitations of the methods which are discussed, when considering their use in a forensic context.
- To assess the legal admissibility of data, acquired from mobile phones using the methods discussed, as evidence in a court of law.

2.4 Method of research

The preliminary work to this thesis forms the literature review in Chapter 3. The previous academic research in the area was discussed. In addition, a brief overview of the methods commonly used to acquire data from mobile phones, and the procedures which should be followed when forensically analysing a mobile phone was formed.

Much of the work performed was in software development, and analysis of mobile phone – PC communication. Whilst the application which was developed is no longer a primary aim of this thesis, many hours were spent working on the application, and deciphering and debugging the protocols used to communicate with the phones. Testing of the application with a number of phones brought about the discovery of a number of problems and limitations inherent in using these methods to acquire data; these issues are discussed in Chapter 4.

The Australian and South Australian legislation was analysed to determine how the law applies to information obtained from a mobile phone. There are only a few sections which specifically relate to the admissibility of electronic information as evidence; Sections 146 and 147 of the Evidence Act 1995 (Commonwealth of Australia) relates to proving that evidence produced by a device is correct. Section 59B of the South Australian Evidence Act 1929 relates to the same topic, although places more burden on proving that the device is functioning correctly at the time that the evidence was produced.

3 Literature Review

The range of information which can be obtained from mobile phones is discussed first, followed by a discussion on how this information can be obtained. There is little academic literature on this topic; as such, a range of sources have been used to form the discussion, including media releases, standards publications, and white papers.

3.1 Data stored in mobile phones

Data on a mobile phone can be found in a number of locations:

- The SIM card (if present).
- The phone's embedded memory.
- The phone's removable memory (i.e. SD card), if present.

In addition to this, subscriber and call related information is also stored by the service provider (Willassen 2005).

3.1.1 Data stored in SIM cards

The Subscriber Identity Module, or SIM card, used in GSM phones, is a smart card which enables connection to GSM networks, and enables the subscriber to be uniquely identified in the network. The SIM card contains a number of files, which contain the user's subscriber information, and personal information, such as:

- The International Mobile Subscriber Identity (IMSI), which is the SIM card's globally unique identifier.
- Language preferences and network (service provider) information.
- Currency information, such as call charge counters.
- Information about the current (or most recent) location of the mobile phone.
- Phone book entries.
- Sent and received SMS messages.
- Recently dialled numbers.

Many of the features available on a SIM card are optional, and therefore may not be implemented by every handset or service provider (ETSI 2004a).

3.1.2 Data stored in phone memory

In addition to the SIM memory, memory is available within the phone to store phone software, and additional data. This space can be used to extend the SIM memory, to store additional phone book data, call logs and so forth.

The following are some examples of the additional information which may be found in a phone's memory:

- Phone settings.
- Calendar information.
- SMS / MMS messages.
- Call log entries
- Time and date.
- Ring tones.
- Data required for / produced by the phone's extra features, such as audio and video recordings, and images.
- Generic data stored in the phone's memory.
- Application executables (Willassen 2005, ETSI 2004b).

Many modern phones come with a relatively large amount of onboard memory. For example, the Nokia 9300 has 80MB onboard memory, with the option to extend it with a removable memory card (Nokia 2005b).

These removable cards are generally used to store multi-media files, such as audio, video, images and MMS messages, and are not used for phone related information (phone book entries, etc) (Willassen 2005). However, the cards can be used for transfer and storage of any form of data.

OBEX capable phones will typically allow the user to store any form of data in the phone's memory. Figure 3 shows various forms of data stored in a Nokia 6225. While the phone may not be able to recognise or display the data, it can be used as a generic storage device.

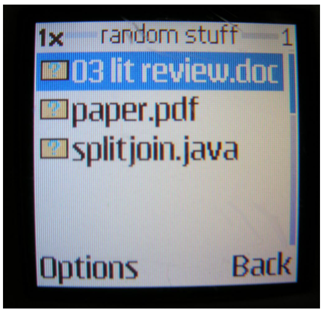


Figure 3: *Generic data stored in a Nokia 6225*

3.1.3 Data stored by the service provider

Data retained by the service provider includes subscriber information, location information, and call and billing information (Mellars 2004, pg. 267).

Whenever a call is made or a text message is sent, a 'call data record' is created and stored, containing, amongst other information, the sending and receiving phone numbers, the length of the call, and the initial and final location of the two parties (Willassen 2003, pg. 13, Goode 2003, pg 9/2). This information is available from the service provider, and therefore is not discussed as it is outside the scope of this thesis.

3.2 Extracting data from mobile phones

It is important to realise that there are a number of basic requirements for a handset and / or SIM card to interface with a computer. Once these requirements are met, the manufacturer is free to implement any other features in any way it wishes. This means that a forensic tool for mobile phones will not be able to target every feature on every type of phone (Willassen 2003, pg. 10).

Furthermore, the different technologies upon which mobile networks are based, such as TDMA, CDMA, and more recent 3G technologies, such as WCDMA, each require the handsets to implement different standards. For example, CDMA phones do not require a SIM card; rather, the software required to connect to the network is in the phone itself (ACA 2005).

Data will either be found in the SIM card if present, or in the phone's memory. There are a number of different methods for obtaining the data. However, there is no accepted standard, as the freedom which manufacturers have when designing their phones means that every phone must be considered separately.

Willassen (2005) proposes the following methodology for forensic analysis of phones:

- Turn the phone off as soon as possible.
- Obtain access codes from the phone owner / service provider
- Analyse the SIM card.
- Analyse any removable memory.
- Analyse the phone memory.

This methodology raises some issues. Mellars (2004, pg. 267 – 268) and Goode (2003, pg. 9/2) state the importance of isolating the phone from the network, so no new information is received. Turning the phone off has the potential to alter data on the phone, but leaving the phone on raises the possibility of new information arriving over the network.

There are ways around this issue; for example, Forensic Telecommunications Services (FTS 2004) offers a 'radio screened foil bag' which isolates the phone from the network. In the absence of such equipment, turning the phone off should be the preferred choice; however, a certain level of trust must be placed in the phone's operating system.

Mellars (2004, pg. 267 – 268) and Goode (2003, pg. 9/2) also note that the order in which data is extracted is important, as removal of the SIM card or battery from some phones will modify the contents of the phone memory. This raises the question of whether the SIM card should be removed from the phone before analysis / imaging. Many phones require the battery to be removed to access the SIM card. This has the potential to alter information in the phone.

Mellars provides an example of this with regard to the Nokia 3310 handset, which loses time and date information as soon as the battery is removed. Removing or replacing the SIM card may also have an effect on the phone's memory (Willassen 2003, pg. 11).

Willassen notes that another method of obtaining data from both the phone and the SIM card is to simply use the phone’s keypad to browse through the phone. However, certain information such as deleted text messages would not be accessible via this method, and the process is time consuming and prone to errors, as pushing the wrong keys could destroy information.

Phone and SIM access codes will generally be needed before certain information can be accessed. These will either be obtainable from the phone’s owner (PIN codes), and / or the service provider (PUK codes) (Willassen 2005, Mellars 2004, pg. 268).

3.2.1 Extracting data from SIM cards

Forensically acceptable extraction of data from the SIM card can potentially be accomplished in two ways. Directly analysing the contents of the SIM card is outside of the scope of this thesis, and hence will not be discussed in a detailed manner.

The first way is through a smart card reader, which are cheap and easy to obtain. The SIM card is accessed and controlled by commands specified in the ETSI ‘TS 31.101’ and ‘TS 51.011’ standards (ETSI 2000, 2004). A terminal program such as Microsoft HyperTerminal can be used to send commands to the SIM card. A number of software applications are also available which perform these tasks, such as Sim Manager (TX Systems 2004) and SIMCon (insideout Forensics 2005).

Another method of accessing the SIM card is through the mobile phone. GSM phones conform to the ETSI ‘TS 27.007’ standard, which specifies a command set. This set includes a command which allows a SIM card command to be embedded, and passed to the SIM card. Responses from the SIM card are passed back in a similar manner. This is effectively identical to directly accessing the SIM card. This command is an optional implementation, however, so there is no guarantee that every GSM phone supports it (ETSI 2004, pp. 88 - 90). If this method were available on every mobile phone, it would reduce the difficulty in analysis, as the SIM card would not need to be removed and analysed separately.

3.2.2 Extracting data from phone memory

The data to be extracted will reside in the phone's embedded memory, and / or in a removable memory card. Data stored in the latter is examinable using a forensic tool such as Encase (Willassen 2005), and therefore is not discussed.

Extracting data from the phone's embedded memory is more complex. Willassen (2005) proposes two forensically sound methods:

- Taking the phone apart and accessing the memory chip directly.
- Tapping in to the phone's motherboard to access the memory chip.

These two methods bypass the phone's operating system and access the memory directly; hence, an exact memory image can be obtained. The only way to directly access the phone's memory is through one of these methods. The use of the JTAG interface (described in Section 3.2.2.7) may also allow a complete memory image to be obtained in a non-destructive manner. Due to their technical nature however, it is currently infeasible to expect the average investigator to use these, or any similar methods.

Therefore, an exact image of a phone's memory cannot be obtained by a non - technical investigator. The methods described above are infeasible, as they would involve detailed knowledge of the phone's design. Thus, the only acceptable method of data extraction is through the phone's software interface. However, analysis using this method places trust in the phone software, that it does not alter the phone's memory. This trust problem is confirmed by Willassen (2005), who also notes related issues, namely that deleted information will not be accessible via a software interface.

Best practice guidelines from the 2000 International Organization on Computer Evidence conference (IOCE 2000) state that phones and other electronic devices should be examined with 'methods that minimise loss / change of data'. It is simply too much trouble to obtain an exact memory image. Therefore, the phone's (and SIM card's) operating system must be trusted not to alter the memory when read commands are executed.

This idea is supported by Robinson & Smith (2001), who acknowledge that the current trend in verifying mobile phone evidence in Britain is based more on the approach used by the investigator to obtain the evidence; as opposed to verifying the correct operation of the phone. The responsibility in this case is on the opposition to prove that the mobile phone is not operating correctly, hence invalidating the information it produces.

3.2.2.1 Manufacturer and third party software

Software packages for data synchronisation between phone and computer are generally available from the phone manufacturers. The phone is usually connected via a data cable, or by infra-red or Bluetooth. Some examples of such software are:

- Nokia PC Suite (Nokia 2005c).
- Sony Ericsson Sync Station (Sony Ericsson 2005a).
- Sony Ericsson File Manager (Sony Ericsson 2005b).

Third party software which performs the same function is also available, such as MightyPhone (FusionOne 2005). These applications are not designed for forensic analysis; hence there is a risk in altering the data on the phone through improper use of such an application.

Nokia PC Suite and Sony Ericsson Sync Station were reviewed using a freeware serial port monitor, Portmon (Russinovich 1999), and a USB monitor, SourceUSB (SourceQuest 2004). This review was performed to determine the methods these applications use to communicate with the phones. It should be noted that wherever communication between a mobile phone and a PC has been referred to, the assumption has been made that the output from these monitor applications reflects the communication which is actually taking place.

Nokia PC Suite was tested with a Nokia 3220 and Nokia 6225 shown in Figure 4, using an official Nokia CA-42 USB cable. There are a number of different versions of the software, however in this case, the same version was recommended for both phones. Predictably, the software uses Nokia's proprietary FBUS protocol for all communication with the phones, and is able to extract the phone book, call logs and calendar entries. OBEX is used to extract media files, ringtones and downloaded applications. PC Suite could extract SMS messages from the 3220, but not from the 6225.

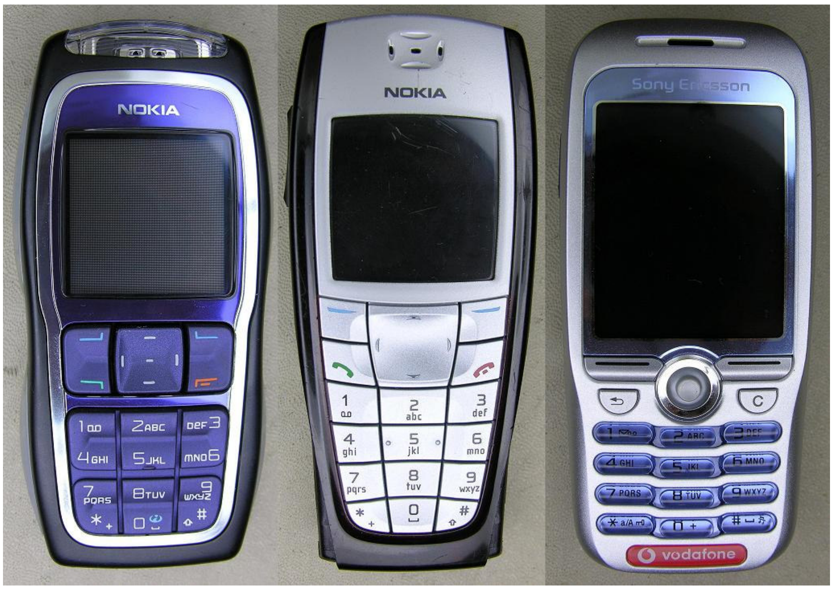


Figure 4: *Nokia 3220, Nokia 6225 and Sony-Ericsson f500i*

Sony Ericsson Sync Station and File Manager were tested with a Sony Ericsson f500i, also shown in Figure 4, using an official Sony Ericsson DRS-11 serial cable. Sync Station is an application used for synchronizing the contents of the phone with an application such as Microsoft Outlook. Data such as contacts and calendar entries can be copied between the phone and an application to ensure both data sets are identical. Sync Station used SyncML over OBEX to extract the phonebook and calendar entries from the phone.

Sony Ericsson File Manager is an application which allows a limited portion of the phone's file system to be accessed. The application used OBEX to access user created / downloaded media files. Only a subset of the phone's file system was accessible.

3.2.2.2 Forensic mobile phone software

There are a number of third party applications which have been designed for forensic analysis of mobile phones, such as:

- PhoneBase (Envisage Systems 2005).

- Oxygen Phone Manager II (Forensic version) (Oxygen Software 2005).
- XRY (Micro Systemation 2005).
- Cell Seizure (Paraben Forensics 2005).

These applications (and a number of others) support a wide range of phones, and claim not to alter any data on the phone; however, they use the same software interfaces as the non-forensic applications, and hence are placing trust in the phone's operating system. Section 4.2 describes these applications in more detail.

3.2.2.3 PC connectivity SDKs

Some phone manufacturers offer SDKs for connectivity with the phone operating system. For example, an SDK is provided by Nokia, providing access to phone information such as the phonebook, sent, received and saved SMS messages, and call logs (Nokia 2003). The SDK supports a very limited range of phones, however, and some data remains inaccessible through the SDK, including images, video and other user files.

Symbian is an operating system found in many modern phones from a wide range of manufacturers, including Nokia, Sony-Ericsson and Panasonic (Symbian 2005). An SDK is provided for PC connectivity with any phone using the Symbian operating system. This SDK allows access to portions of the file system of the phone (Symbian 2004).

3.2.2.4 AT commands

AT, or Hayes commands were originally designed by Hayes Microsystems for modem control (Hayes Microsystems 2005). A standard for GSM phones was developed which includes commands to access information such as phone book entries, call logs and SMS messages.

AT commands were designed to control a modem from a PC, and when communicating with a mobile phone, the phone's internal logic receives and parses the commands. AT commands were originally designed in the early 1980s, by Hayes Microsystems, and were taken up by other manufacturers as a standard for modem control (Durda 2004).

The AT commands specified in the TS 27.007 and TS 27.005 standards theoretically provide access to a large amount of information available in GSM phones and SIM cards (ETSI 2004b, ETSI 2005). The TIA/EIA/IS-707 standard provides a set of AT commands for CDMA phones (TIA 1999). Unfortunately many of the commands cannot be relied upon, as they are optional implementations, and cannot be guaranteed to be present on every phone. In practice however, information such as the following are generally available on GSM phones through AT commands:

- The phone's manufacturer, model, and version information.
- The phone's International Mobile Equipment Identity (IMEI).
- The SIM card's International Mobile Subscriber Identity (IMSI).
- Phone book entries.
- Call log entries.
- Sent and received SMS messages.

(ETSI 2005).

AT commands for CDMA phones will generally provide access to a much more limited set of information:

- The phone's manufacturer, model and version information.
- The phone's Electronic Serial Number (ESN).

(TIA 1999).

3.2.2.5 OBEX

OBEX (OBject EXchange) is a communication protocol originally designed for use in infrared devices. OBEX has been incorporated into the specification for Bluetooth devices, and can also be used over cable connections. The OBEX specification has deliberately been left open to interpretation, to ensure that it can be used in a wide variety of applications (IrDA 2003). OBEX is similar to HTTP, in that it is a transport mechanism, over which different types of data can be transported.

One of the most common uses of OBEX is for remote file browsing, for example, of a mobile phone's media gallery. Only a subset of the phone's entire file system will be made available, and will provide access to items such as images, audio and video recordings, ringtones and downloaded applications.

IRMC (Infrared Mobile Communications) is a data synchronization standard which can be used over OBEX. It allows the synchronization of items such as calendar and phone book entries between a mobile phone and a PC (ETSI 2003).

The OBEX specification also includes support for SyncML, another synchronization protocol which is slowly replacing IRMC. These protocols are typically used to synchronize data between a mobile phone and an application such as Microsoft Outlook (OMA 2002). While the concept of synchronization implies data being changed in the memory of both participants in a SyncML session, the specification includes a 'Get' command which allows one way retrieval of data from a phone.

3.2.2.6 Nokia FBUS

FBUS is Nokia's most recent proprietary protocol for phone PC communication. There are a number of different versions of FBUS, all of which are based on a similar structure. Nokia has not published any documentation regarding FBUS, and the only publicly available work has been as a result of monitoring and reverse engineering communication between Nokia phones and software.

Open source projects Gnokii (Gnokii Project 2005) and the related Gammu (Wiacek 2005) form the largest public body of work on FBUS. Both projects are developed primarily for the Linux operating system, and attempt to enable full access and control over the functionality and data provided by mobile phones. Many Nokia models are supported, in addition to a limited range of other manufacturer models.

FBUS will generally allow the access of basic phone information such as phonebook entries, call logs, SMS messages and calendar entries. In addition to this, limited file system access is accessible on modern Nokia phones which support OBEX, however the OBEX communication must be encapsulated within FBUS communication, and requires FBUS commands for initialisation.

3.2.2.7 JTAG

JTAG (Joint Test Action Group) (IEEE 2001) is a widely used standard developed by the IEEE in the early 1990s. The JTAG standard specifies an interface and commands which can be used for testing and debugging of the hardware components in an electronic device. JTAG works by performing a boundary scan on a given component; this is essentially a test of the input and output pins connected to that component. JTAG can test the correct functioning of an individual component, and the correct interconnections and interactions between components. Official JTAG cables for mobile phones are not publicly available, nor are JTAG interface specifications for particular phones.

JTAG may be of interest in a forensic context, as it could theoretically provide direct access to a mobile phone's memory. Memory in any device is typically accessed through a hardware memory controller. If this memory controller can be tested with JTAG, the memory it manages will be accessible.

The JTAG standard requires a four pin connection onto the device circuit board (an optional fifth pin can be added, to provide a system reset function). Shown in Figures 5 to 8 are the JTAG connections from a variety of mobile phones. As can be seen from the four examples given, JTAG connections are generally different for every phone model. The Sony Ericsson f500i connection in particular is quite unusual. The pins can be in arbitrary positions on the phone, of different sizes, and unlabelled. These examples show that one JTAG cable would most likely not be able to be used with more than a few phones.

For every phone shown, the battery must be removed to access the JTAG connection. In addition, the SIM card must be removed from the f500i.

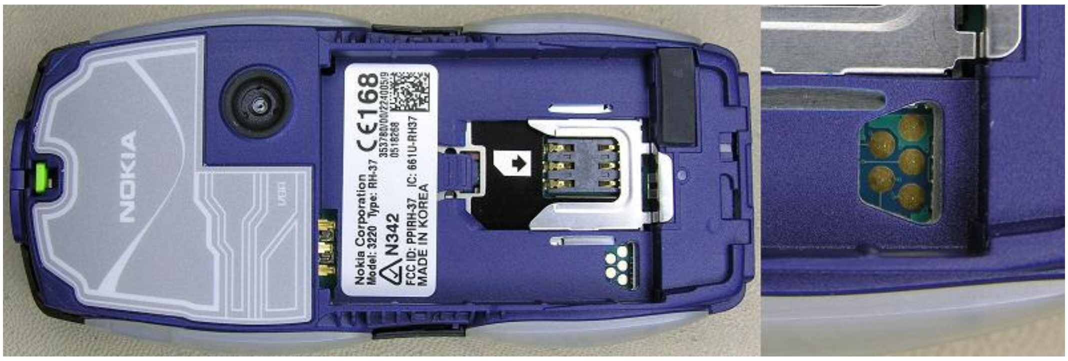


Figure 5: *Nokia 3220 JTAG interface*

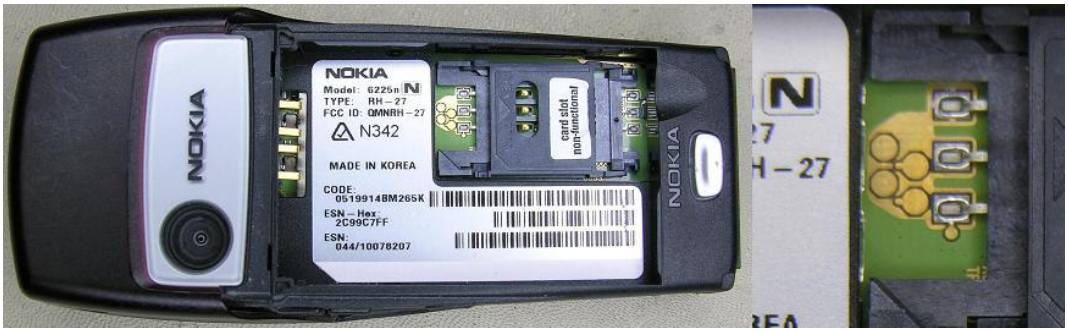


Figure 6: *Nokia 6225 JTAG interface*

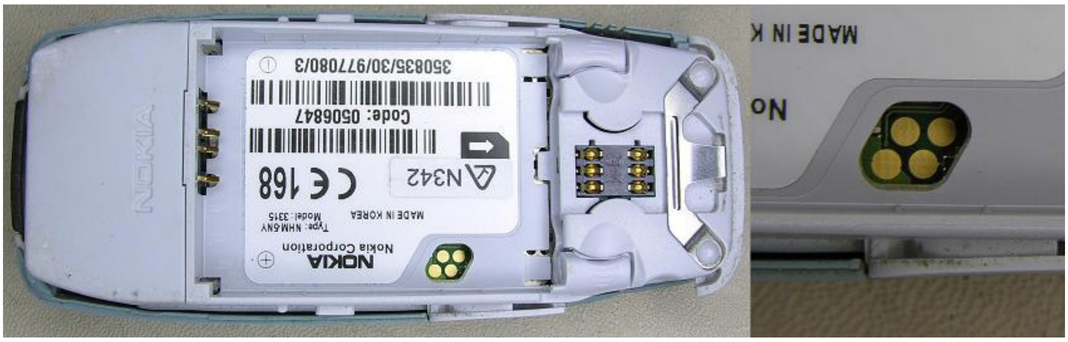


Figure 7: *Nokia 3315 JTAG interface*

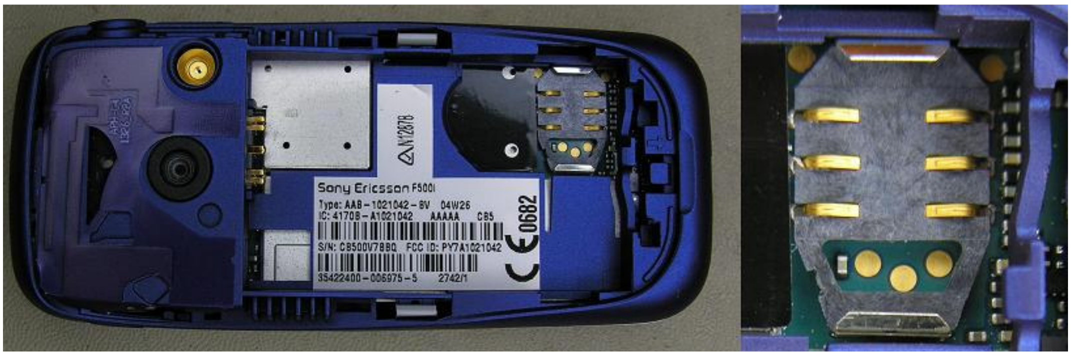


Figure 8: *Sony-Ericsson f500i JTAG interface*

Testing could not be performed using the phones’ JTAG interfaces. No information is available on device specific implementations of JTAG, and official cables are not (publicly) available. Despite this, the JTAG interface may provide a forensically sound method of accessing the phone's memory, and if so, would be able to obtain a complete memory image without requiring the phone to be taken apart.

3.3 Legislation regarding electronic evidence

The Evidence Act 1995 (Commonwealth of Australia) outlines what constitutes evidence, and the types of information which can be admitted as evidence. Section 146 of the Act implies that any device from which information is obtained can be assumed to be working correctly; hence its output is correct. Only if evidence is presented showing that the device is not functioning normally, will the information produced by the device be considered inadmissible as evidence. This essentially means that if data produced by a mobile phone is presented as evidence, the onus is on the opposition to show that the mobile phone was not functioning or outputting data correctly when the data was acquired. Otherwise the data will be accepted as evidence.

Section 48 of the Evidence Act is also relevant to electronic evidence. When information acquired from a computer or electronic device is presented in a court as evidence, it must be presented as a human readable or understandable document. A document which was produced from information acquired from an electronic device can be thought of as a reproduction of information stored in the device. Therefore, by Section 48 of the Act, information which is acquired from a mobile phone (or any other electronic device) is admissible as evidence.

The South Australian Evidence Act 1929 (Parliament of South Australia) takes a different approach with regard to information obtained from a computer or electronic device. Section 59B relates to the admissibility of data generated by a computer, where a computer is defined to be any device capable of recording, processing and producing data. Rather than assuming a computer to be in correct working order, the Act requires the court to be satisfied that the computer is functioning correctly, and that there is no evidence to suggest otherwise. This essentially shifts the onus of proving the correct functionality of a computer or device on to the party which wishes to use computer generated information as evidence. If the party can prove that the computer / device is functioning correctly, the information will be admissible as evidence.

Subsection 2g of Section 59B is specifically related to the procedures used to obtain the evidence, by stating that the court must be satisfied that the processes used to acquire the information do not affect the accuracy of the information itself. This implies that the processes used must have been shown to produce accurate information.

4 Results

The various methods of acquiring data from mobile phones were analysed in a number of ways. The existing mobile phone forensic applications were reviewed, to determine the methods they use to acquire data. In addition, the methods were tested with a number of phones, in an attempt to discover any limitations in their use to acquire data.

The research application, discussed in Section 4.3, was also used to determine any limitations in the methods used, and to determine the level of difficulty involved in implementing these methods with no commercial support.

4.1 Methods of acquiring data

The methods of acquiring data from a mobile phone mentioned below are commonly used by forensic applications as discussed in Section 4.2. Also mentioned is the fact that using these methods requires trust to be placed in the phone's software or firmware. As the internal workings of a mobile phone are proprietary information, there is no way to verify that the software which receives requests to access data is working as it should, or as it appears.

The only argument which could be used to confirm the correct operation of these methods is that they are in common use, and are regularly relied upon to acquire data, forensically or otherwise. This argument is insufficient for these methods to be scientifically acceptable, however. Each manufacturer has their own potentially different implementation of the standards, and different phones may handle the same command in different ways.

4.1.1 AT Commands

Where the original AT command set consisted of commands for dialing, answering and controlling the ways in which data is transferred, the ETSI TS 27.007 and TS 27.005 specifications add the ability to access the phonebook, call logs and SMS messages stored in a GSM phone (ETSI 2004b, ETSI 2005). Whether these additional commands are implemented in the phone's internal modem or in software is up to the manufacturer, but at some point these commands will cause the phone's memory to be accessed.

Communication via AT commands takes the form of a command response protocol. Over a serial connection, the client sends a command followed by a carriage return character, and optionally with a new line character, and receives a formatted response. An example of AT command communication with a Sony Ericsson f500i, is shown in Figure 9.

The command `AT+CGSN` is a request for the phone's IMEI. The phone also echoes the command which was sent to it; this is not shown in any of the examples in this Chapter. In all of the figures in this Chapter, the symbol '\n' refers to a new line character, and the symbol '\r' refers to a carriage return character.

PC:	AT+CGSN \r
Phone:	354224000069755 \r\n \r\n OK \r\n

Figure 9: *AT command communication*

Most data obtainable via AT commands will require the user to be verified to the SIM card and / or phone. Figure 10 shows the f500i denying access to information as the SIM PIN has not been entered.

PC: This command tells the phone to return an error code when an error occurs.	`AT+CMEE=1`
	`\r`

Phone:	`OK` `\r\n`
PC: This command requests the ISMI, the SIM card's unique ID.	`AT+CIMI` `\r`

Phone: The error code `11` means that the SIM card PIN is required to access this information.	`+CME ERROR: 11`
	`\r\n`

PC: This command passes the SIM PIN to the phone.	`AT+CPIN="2580"`
	`\r`

Phone: The phone indicates that the PIN was accepted.	`OK` `\r\n`
PC:	`AT+CIMI` `\r`

Phone: The information is now accessible.	`505010388205398`
	`\r\n`
	`\r\n`
	`OK`
	`\r\n`

Figure 10: *Phone denying access to information over AT commands*

As described in ETSI 27.005 (ETSI 1999), an SMS message can be in one of four states; this is shown in Figure 11.

SMS status	GSM representation	FBUS representation	Meaning
RECEIVED / UNREAD	0	03	The message has been received but not yet read.
RECEIVED / READ	1	01	The message has bee received and read.
STORED / UNSENT	2	07	The message has been written and saved, but not sent.
STORED / SENT	3	05	The message has been written, saved and sent.

Figure 11: *SMS message status codes*

When an SMS message is accessed via AT commands, its status may be inadvertently changed, as a result of the message being accessed. An example of this, taken from communication with a Sony Ericsson f500i is shown in Figure 12. The command is a request to retrieve an SMS message; this command has been repeated twice, and the response from the phone differs slightly each time. The message status has changed from RECEIVED / UNREAD to RECEIVED / READ.

PC:	`AT+CMGR=10`
	`/r`

Phone: The value `0` on the third line gives the status of this message.	`+CMGR: 0,,35`
	`/r/n`
	…

PC:	`AT+CMGR=10`
	`/r`

Phone: Note that the status of the message has changed from `0` to `1`.	`+CMGR: 1,,35`
	`/r/n`
	…

Figure 12: *SMS status code changing in reaction to AT commands*

The GSM Nokia phone which was used for testing, the 3220, does not support retrieval of SMS via AT commands. Hence the only way to retrieve SMS from this phone is through Nokia FBUS, described in Section 4.1.3.

4.1.2 OBEX

OBEX can be thought of as a binary version of HTTP. The most important functions in an OBEX session are ‘Get’, ‘Put’ and ‘SetPath’. The Get operation allows access to an object which is stored on a device. Put allows an object to be copied to the remote device. SetPath enables the folder system on the device to be traversed. A number of different services can be run on top of OBEX, including the folder browsing service, and a SyncML session. A typical example of OBEX communication with a Sony Ericsson f500i is shown in Figure 13.

PC: The first byte indicates that this is a SetPath operation.	`85 00 08 02 00 01 00 03`
Phone: The first byte indicates that the previous command was a success.	`A0 00 03`
PC: This is a request to get the folder listing of the current folder.	`83 00 24 CB 00 00 00 01 01 00 03 42 00 19 78 2D 6F 62 65 78 2F 66 6F 6C 64 65 72 2D 6C 69 73 74 69 6E 67 00`
Phone: The folder listing is returned in XML.	`A0 01 C5 CB 00 00 00 01 49 01 BD …[XML folder listing]…`

Figure 13: *OBEX communication*

An OBEX packet consists of an operation code (opcode), the packet length, command specific bytes, and optional fields called headers. This structure is shown in Figure 14. Each OBEX command has a unique structure. For example, in addition to the opcode, packet length and optional headers, a connect request packet must contain information such as the OBEX version number and the maximum packet length which will be accepted.

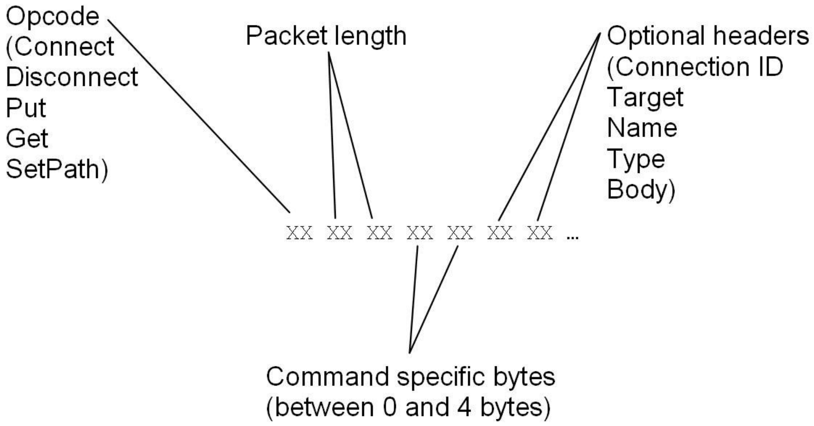


Figure 14: *OBEX packet structure*

The method of initializing an OBEX session varies from phone to phone. An OBEX session with a Sony Ericsson phone must be initialised via an AT command, `AT*EOBEX`. In contrast, OBEX communication with a Nokia phone must occur within an FBUS session, Nokia’s proprietary protocol.

Once an OBEX session has been initialised, the client (the PC application) must connect to the OBEX service running on the phone, and in this case, specify the file browsing service as the target. Once the connection has been made, the client is able to retrieve folder listings, change the current folder, and extract files. Only a subset of the phone's entire file system is made available via OBEX, typically the phone's media gallery. In addition to this, file permissions may be present on certain files, restricting their access via OBEX.

An example of OBEX session initialisation with a Sony Ericsson f500i is shown in Figure 15.

PC: The OBEX session must be initialised via an AT command.	`AT*EOBEX` `\r`

Phone: The response indicates that a connection has been made to the OBEX server.	`AT*EOBEX`
	`\r\n`
	`\r\n`
	`CONNECT`
	`\r\n`

PC: This is an OBEX connect request, setting the target as the file browsing service. This is done by specifying the file browser's unique ID (the last 16 bytes of the data).	`80 00 1A 10 00 FF FF 46`
	`00 13 F9 EC 7B C4 95 3C`
	`11 D2 98 4E 52 54 00 DC`
	`9E 09`

Phone: This is a success response, giving a unique connection ID (`00 00 00 03`), and repeating the target ID.	`A0 00 1F 10 00 04 00 CB`
	`00 00 00 03 4A 00 13 F9`
	`EC 7B C4 95 3C 11 D2 98`
	`4E 52 54 00 DC 9E 09`

PC: This is a set path command, requesting the path to be set to the root folder.	`85 00 08 02 00 01 00 03`
Phone: This is a success response.	`A0 00 03`

Figure 15: *OBEX session initialisation*

OBEX communication with Nokia phones must occur within a FBUS session. This has the effect that the initialisation process is considerably more complex than the previous example. Additional FBUS specific commands are required to initialise an OBEX session, and the client – server relationship between the phone and PC appears to be reversed during the initialisation. OBEX over FBUS session initialisation is discussed in more detail in Appendix C.

Folder listings are returned by the phone in XML, as shown in Figure 16.

PC:	[setpath to "pictures"] [get folder listing]
Phone:	… <folder-listing version="1.0"><parent-folder/> <file name="Startup.gif" size="28075"/> <file name="Shutdown.gif" size="28296"/> <file name="Screensaver.gif" size="28094"/> … <file name="Tunnel Music.jpg" size="11904"/> <file name="Vacation.gif" size="53986"/> <file name="Winter.gif" size="2900"/> <folder name="camera_semc"/> </folder-listing>

Figure 16: *OBEX XML folder listing*

Access to a file may be prohibited via OBEX, by way of copyright or distribution flags present in the file itself. For example, the distribution of 3GPP video files can be limited to certain devices, as described in the Open Mobile Alliance Digital Rights Management specification (OMA 2005). Figure 17 shows a Sony Ericsson f500i denying access to a file, 'Surfing.3gp'.

PC: This is a GET command, requesting a file 'Sufting.3gp'.	83 00 23 CB 00 00 00 04 01 00 1B 00 53 00 75 00 72 00 66 00 69 00 6E 00 67 00 2E 00 33 00 67 00 70 00 00
Phone: The value `C1` indicates that access to this file is not allowed.	C1 00 03

Figure 17: *Phone denying access to a file over OBEX*

OBEX provides methods for authentication between devices and / or PCs. Authentication is based on a shared secret, such as a password or PIN. Although none of the phones tested required authentication for an OBEX session, if a phone did implement the authentication procedure, the secret would be required to acquire data from the phone via OBEX.

4.1.3 Nokia FBUS

FBUS is a command response protocol, although is slightly more complex than AT commands and OBEX. An example of typical FBUS communication, with a Nokia 6225 is shown in Figure 18. FBUS packets are typically referred to as frames; in this section, the two terms are interchangeable.

As a result of the unreliable nature of information on FBUS, there are conflicting opinions on the amount of data which is actually obtainable via the protocol. The fact that different phones may use different implementations of FBUS throws more doubt on the topic. For example, two of the phones used in testing, the Nokia 6225 and Nokia 3220, use the same version of FBUS. However, there are slight differences in the two implementations, such as different memory identifiers, and slightly varying commands. In particular, the 6225 appears to provide no method of obtaining SMS messages. Nokia's official synchronization software, PC Suite, also fails to extract SMS from the 6225, indicating that SMS messages cannot be obtained from this particular phone via any known method. The FBUS commands which are known, and were implemented in the research application, are listed in Appendix D.

PC: This command tells the phone to set all communication parameters to their default values.	`AT&F` `\r`

Phone:	`AT&F`
	`\r\n`
	`\r\n`
	`OK`
	`\r\n`

PC: This command is required to initialise an FBUS session.	`AT*NOKIAFBUS` `\r`
Phone:	`[FBUS initialisation]`
…	…
PC: This is a request for a calendar entry from memory location `00 00 00 85`	`1E 00 10 13 00 10 00 01 00 7D 55 55 00 00 00 85 FF FF FF FF 01 46 5A E9`

Phone: The first frame is the acknowledgement frame for the command received from the PC. The next frame is the phone's response, which contains a calendar entry.	`1E 10 00 7F 00 02 13 06 0D 6B`
	`1E 10 00 13 00 62 01 60 00 7E 00 00 00 00 00 00 00 00 00 85 00 00 00 00 80 00 00 64 FF FF FF FF 02 00 07 D5 0A 0E 11 00 07 D5 0A 0E 11 00 00 00 FF FF 20 00 00 00 00 00 00 15 00 00 00 6D 00 69 00 73 00 73 00 79 00 20 00 68 00 69 00 67 00 67 00 69 00 6E 00 73 00 20 00 74 00 6F 00 6D 00 20 00 36 00 70 00 6D 01 44 43 45`

PC: This is an acknowledgement frame for the phone's response.	`1E 00 10 7F 00 02 13 04 1D 79`

Figure 18: *FBUS communication*

The structure of an FBUS packet is shown in Figure 19. Every FBUS packet over a cable connection begins with the frame ID `0x1E`. The phone and PC are both given identifiers, which constitute the source and destination bytes. The 'frames to go' byte indicates whether the current response consists of more frames. The body length byte is the length of the packet body, and the frames to go, sequence number and padding byte, if present. Sequence numbers in an FBUS session increment from `0x40` to `0x48` and then loop back to `0x40`, with the exception that the first sequence number is `0x60`.

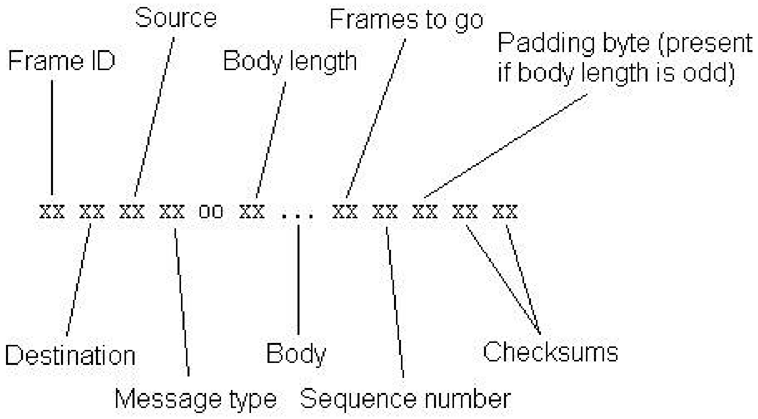


Figure 19: *FBUS packet structure*

Whenever a frame is sent in an FBUS session, the receiving party must send an acknowledgement frame back to the sender. Acknowledgement frames have the same structure as normal frames, with the exception that there is no frame body, and the frames to go and sequence number bytes are replaced with the message type and the least significant four bytes of the sequence number of the frame being acknowledged, respectively.

An FBUS session is initialised via AT commands. After initialisation, the only communication understood by the phone is FBUS; AT commands will not be accepted during an FBUS session. However, there appears to be no method of ending an FBUS session. Once started, the only way to end the FBUS session, and resume using AT commands, is to either unplug the cable from the phone and then plug it back in, or to turn the phone off and then back on. Nokia PC suite also fails to reset FBUS sessions; When PC Suite is exited, the phone remains in FBUS mode, and will not accept any other form of communication until the session is reset via one of the above methods.

An FBUS session initialisation procedure with a Nokia 6225 is shown in Figure 20.

PC: This command ensures that AT commands are supported.	`AT` `\r\n`
Phone:	`OK` `\r\n`
PC: This command resets any session specific parameters to their defaults.	`AT&F` `\r\n`
Phone:	`OK` `\r\n`
PC:	`AT*NOKIAFBUS` `\r\n`

Phone: The phone begins responding to the AT command normally (by echoing the command), but is interrupted arbitrarily by the FBUS service.	`AT*NOKIAFB`
	`F8 55 55 55 55 55 55 55 55`
	`1E FF 00 D0 00 03 01 01 E0 00 FF 2D`

PC: The `0x55` characters are used by the phone and PC to synchronize with each other.	`55 55 55 55 55 55 55 55`
	`[normal FBUS command]`

Figure 20: *Nokia FBUS session initialisation*

There are a number of omissions from information retrieved via FBUS. For example, an SMS message can be acquired from the phone in Unicode, or in PDU, which is the encoding used by phones when sending SMS messages (ETSI 1999). If a particularly long SMS message is retrieved in Unicode format, the end of that message may be omitted, so that the message can fit within a predefined packet size. The end of the message will not be accessible. This problem can be overcome by retrieving the message in encoded PDU form, however the message must then be converted into Unicode to be human readable.

If an SMS message is received from someone in the phone's phonebook, there will be a name associated with the number the message originated from. However, the phone will not return this association via an FBUS command to retrieve the message; only the number is returned. This is further complicated by Nokia's lack of support for the SMS AT commands, which do return the name associated with the number. While this can be overcome, it means having to retrieve and manually search through the phonebook to find the name associated with the number.

There are a number of issues regarding the methods in which Nokia phones store time. SMS message times can be stored in different formats. For example, shown in Figures 21 and 22 are two methods which a Nokia 3220 used to store the sending time of two different messages. The time in Figure 21 corresponds to a received message, and the time in Figure 22 to a sent message. Both messages were acquired in PDU format; the time is represented in the same way when the message is retrieved in Unicode format.

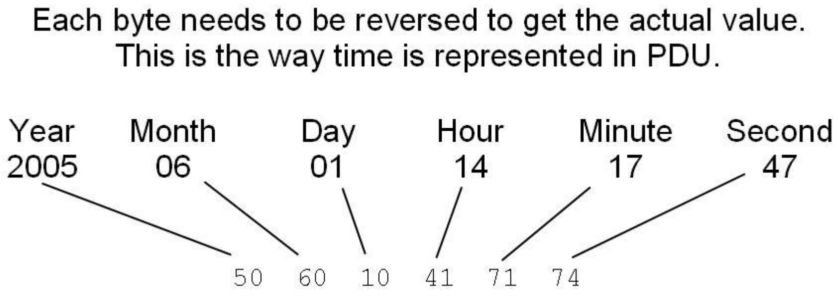

Each byte needs to be reversed to get the actual value.
This is the way time is represented in PDU.
Year 2005
Month 06
Day 01
Hour 14
Minute 17
Second 47
50 60 10 41 71 74

Figure 21: *FBUS received message time representation*

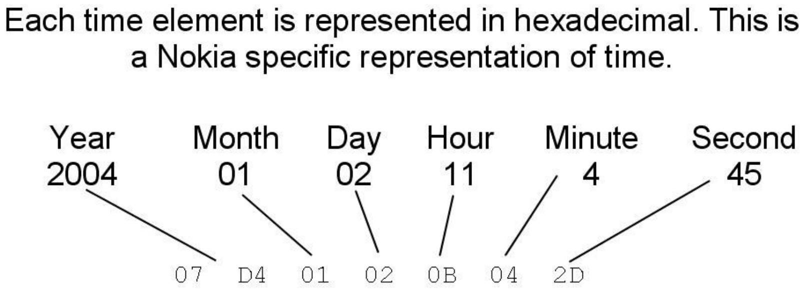

Each time element is represented in hexadecimal. This is
a Nokia specific representation of time.
Year 2004
Month 01
Day 02
Hour 11
Minute 4
Second 45
07 D4 01 02 0B 04 2D

Figure 22: *FBUS sent message time representation*

This example highlights another issue, in that the timezone to which the phone is set is not present in either representation of time. Despite requiring the user to set the timezone, the Nokia 3220 does not give any method of accessing the timezone. Additionally, timestamps for data such as call log entries and calendar entries omit the timezone. In contrast to the 3220, the Nokia 6225 does not use any timezone setting. Figure 23 shows the response from the 3220 for two FBUS commands to retrieve the phone's time. In the first response, the phone's timezone has manually been set to GMT 0; in the second response, the timezone is GMT +2 hours. Neither the 6225 nor the

3220 support the optional AT command to retrieve the phone's time; hence, this FBUS command is the only way to do so.

[timezone is manually set to GMT 0]	

PC:	`1E 00 10 19 00 06 00 01`
	`00 0A 01 60 0F 74`

Phone: The first 7 bytes of the third row corresponds to the time (2005-01-01 22:23:16).	`1E 10 00 19 00 18 01 32`
	`00 0B 01 02 01 0C 01 03`
	`07 D4 01 01 16 17 10 00`
	`04 04 01 00 01 41 1A A2`

[timezone is manually set to GMT +2 hours]	

PC:	`1E 00 10 19 00 06 00 01`
	`00 0A 01 60 0F 74`

Phone: Note that the bytes which represent time have changed, but the rest of the response is exactly the same (The last two bytes are checksums over the entire packet, so they have changed to reflect the change in the time bytes).	`1E 10 00 19 00 18 01 32`
	`00 0B 01 02 01 0C 01 03`
	`07 D4 01 02 00 18 0A 00`
	`04 04 01 00 01 41 16 AE`

Figure 23: *FBUS time retrieval command*

Some more discrepancies arise when considering the security PINs required to use a phone. The Nokia 3220, being a GSM phone, may require a PIN to identify the user to the SIM card. The output in Figure 24 shows that if this PIN has not been entered, the phone will not allow an FBUS session to be initialised.

PC:	`AT&F` `\r`

Phone:	`AT&F`
	`\r\n`
	`\r\n`
	`ERROR`
	`\r\n`

PC:	`AT*NOKIAFBUS` `\r`

Phone:	`AT*NOKIAFBUS`
	`\r\n`
	`\r\n`
	`ERROR`
	`\r\n`

Figure 24: *Phone denying FBUS session initialisation*

The CDMA 6225 allows the use of a security PIN, which must be entered when the phone is turned on, so the user can be identified to the phone. In contrast to the 3220, however, when this PIN has not been entered into the 6225, the phone will still allow all information to be accessed via FBUS.

Data on the phone can be modified when using FBUS, in the same way with AT commands as described in Section 4.1.1. Consider the data shown in Figure 25, taken from communication with a Nokia 3220. The command sent by the PC is a request to retrieve an SMS message. This command has been repeated twice, and the response from the phone differs slightly. The status of the SMS message has changed from RECEIVED / UNREAD to RECEIVED / READ.

PC:	`1E 00 10 14 00 0C 00 01 00 02 01 02 00 04 01 00 01 43 0F 5E`
Phone: The twelfth byte gives the status of the SMS message.	`1E 10 00 14 00 49 01 67 00 03 00 03 01 02 00 04 00 …`
PC:	`1E 00 10 14 00 0C 00 01 00 02 01 02 00 04 01 00 01 43 0F 5E`
Phone: Note that the twelfth byte has changed from `03` to `01`.	`1E 10 00 14 00 49 01 67 00 03 00 01 01 02 00 04 00 …`

Figure 25: *SMS status code changing in reaction to FBUS commands*

4.2 Review of existing applications

As discussed in Section 3.2.2.2, a number of applications exist which claim to acquire data from mobile phones in a forensically sound manner.

A limited number of these applications were tested to determine the methods used to acquire data. Only a small number of these applications were able to be obtained; the applications tested were either trial versions, or freely available. The applications were tested with a number of mobile phones. The communication occurring between the phones and the software was monitored, again using Portmon and SourceUSB.

Envisage System's Phonebase (Envisage Systems 2005) can be used to acquire SMS messages and phone book entries from SIM cards, and a wide variety of mobile phones. An optional 'memory module' is available which can extract memory from Nokia phones; however, no information is provided on what this memory may contain.

The Phonebase website stresses that the software will not allow any data to be changed on the phone / SIM card. However, Envisage provides no information on the methods used to acquire data from mobile phones, and an email sent to the company was left unanswered (Appendix A).

Micro Systemation's XRY (Micro Systemation 2005) allows data to be acquired from a wide variety of mobile phones. Similar to Envisage Systems, Micro Systemation claims that no data on the phone is modified, and that there is no indication that information on the phone has been read. Email correspondence with Micro Systemation (Appendix A) confirmed that XRY uses AT commands, OBEX, Nokia FBUS and IRMC over OBEX, in addition to other proprietary methods developed by Micro Systemation.

TULP2G (NFI 2005) is an open source project initiated by the Netherlands Forensic Institute. The aim of TULP2G is to provide a framework for the forensic acquisition of data from electronic devices; this is accomplished by the use of a plugin architecture. A number of plugins are provided with the release for use with mobile phones.

As TULP2G is open source, the source code can be viewed. TULP2G uses the GSM AT commands, OBEX and IRMC for data acquisition. Additionally, TULP2G implements the Samsung and Siemens extensions to the GSM AT command set, and hence is able to extract additional data from these models.

Oxygen Phone Manager (Oxygen Software 2005) is a phone synchronization program designed for use with Nokia phones. Oxygen software has made a forensic version of the application which claims to prevent any changes in the data on the phone. This forensic version is a free download, and was tested with a Nokia 6225 and Nokia 3220.

Oxygen uses Nokia FBUS and OBEX commands embedded in FBUS packets to extract data from phones. Using these methods, the application was able to extract most data from the 6225, but was unable to extract SMS. Oxygen began communicating with both phones as soon as the application was loaded. No indication of this activity was given, however.

Some strange behaviour was exhibited when the 3220 was tested. As shown in Section 4.1.3, three AT commands should be sent to initialise an FBUS session. The USB monitor output indicated that Oxygen attempted to send all of these commands simultaneously, without waiting for responses from the phone. After these commands were sent, Oxygen proceeded to send a number of FBUS commands to retrieve data such as battery and signal levels.

The phone responded with jargon, as the FBUS session had not been properly initialised.

If left for a period of time with no communication, an FBUS session must be re-initialised by sending a series of ‘Q’ (`0x55`) characters. This is only necessary if no commands are sent to the phone for more than one or two seconds. Oxygen was re-initialising the FBUS session for every command sent, despite delays of only a few milliseconds. While this may not have any adverse effects on the data stored in the phone, when compared with the correct behaviour exhibited by Nokia PC Suite, it may indicate that the protocol has not been implemented correctly.

Like Oxygen, Compelson Labs’ Mobiledit (Compelson Labs 2005) has been designed for phone synchronization purposes. A forensic version has been released, which claims to prevent data changes on the phone. Mobiledit supports a broad range of phones from many manufacturers. A demonstration version of Mobiledit Forensic was tested, with a Nokia 6225, Nokia 3220 and Sony Ericsson f500i.

Mobiledit uses AT commands, OBEX and Nokia FBUS. It may use other proprietary protocols for other phones, but only the three phones were tested. The application worked with the f500i, extracting the phonebook, call logs, SMS messages and media files from the phone using AT commands and OBEX. Additionally, Mobiledit was able to extract the same types of data from the Nokia 3220 using AT commands, Nokia FBUS, and OBEX packets embedded within FBUS.

Mobiledit failed to recognise the Nokia 6225. An FBUS session was successfully established, and the application was sending commands to retrieve information such as battery and signal level, model and version information, and was receiving valid responses to these commands. The user interface failed to reflect this, however. These commands were repeated regularly; 256 kilobytes of information were sent and received between the phone and application in approximately 20 seconds.

Paraben Forensics Cell Seizure (Paraben Forensics 2005a) is an application which has been designed for forensic extraction of data from mobile phones.

A demo version is available, however could not be tested, as it did not seem to support the available phones.

Paraben outline the methods used by Cell Seizure on the product website (Paraben Forensics 2005b). The primary method of extracting data is via AT commands. Some manufacturers have developed extensions to the AT command protocol which provides access to additional information. Cell Seizure also uses FBUS for Nokia phones, and some other proprietary protocols for other models. OBEX is used for some phones.

Paraben have developed some physical methods of data extraction for some Siemens and Samsung phones, which can be used via Cell Seizure. These plugins can acquire a full dump of the phone memory.

With the exception of TULP2G, all of the companies which develop the applications discussed claim that their products do not modify any data on the phone. However, all of these applications use standard connections (Infrared, Bluetooth, USB or serial cable) to communicate with phones. This implies that the applications are using the phone's software interface which, as shown in Section 4.1, has severe limitations and issues.

4.3 Research application

A data acquisition application for use with mobile phones was developed, primarily to test the methods discussed, and to see how much data can be acquired via a software interface.

The application was primarily written in Java, with one component written in C. A serial or virtual serial connection to the phone is required for communication. The methods used to extract data include AT commands, OBEX and Nokia FBUS. Due to time restrictions, no other proprietary protocols could be targeted. An implementation of SyncML would also have provided access to more information. These methods are the same used by the forensic applications discussed previously.

An overview of the application architecture is shown in Figure 26, and a UML diagram is given in Appendix B.

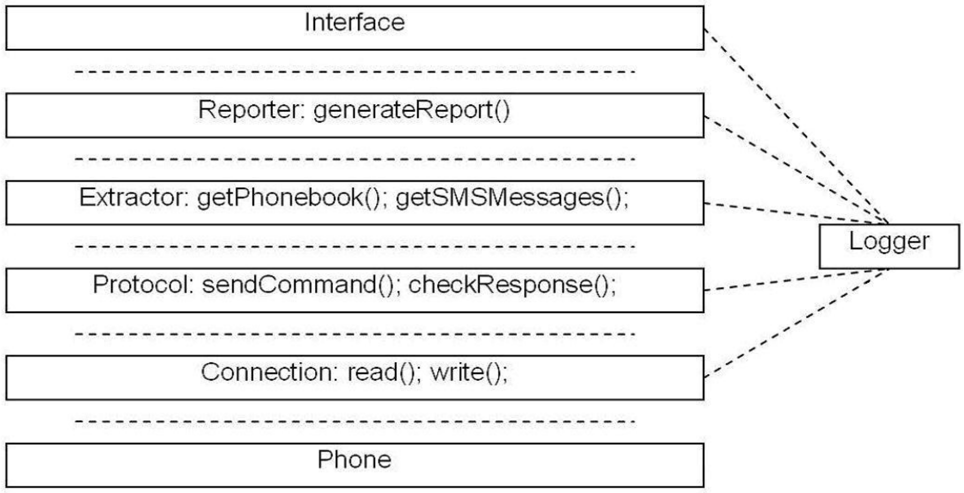

Interface
Reporter: generateReport()
Extractor: getPhonebook(); getSMSMessages();
Protocol: sendCommand(); checkResponse();
Connection: read(); write();
Phone
Logger

Figure 26: *Research application structure*

The application interfaces with the phone via a Connection object, which allows the sending and receiving of byte arrays. The connection to the phone is made possible via the Java COMM API (Sun Microsystems 2005).

The Connection object is encapsulated within a Protocol object, which manages protocol specific aspects of the various methods of communication. For example, a NokiaFBUSProtocol object calculates sequence numbers and checksums for every packet sent to the phone.

An Extractor object uses a Protocol object to send and receive data to and from the phone. The extractor object provides a high level interface, with methods such as getPhonebook to extract the entire phone book from the phone. For example, an ATCommandExtractor implements the commands in the GSM AT command specification.

Note that a specific Extractor object does not have to use its matching Protocol object. This is necessary, as a NokiaFBUSExtractor object may need to use OBEX, which requires an OBEXProtocol object. The OBEXProtocol object must then use a NokiaFBUSProtocol object to embed the OBEX communication within FBUS.

The Extractor layer is the highest implemented so far. However a design has been developed for the higher layers. An Extractor object will be encapsulated within a matching Reporter object, which calls the methods provided by the Extractor, and organizes the information for display to the user. There are a number of important issues raised:

- A generic method must be used to display information, and be applicable to all forms of information which may be extracted from a mobile phone. The current plan is to use a table which enables information to be listed, with arbitrary use of columns for different forms of information. Items such as images and video will be represented in the table via the file metadata, and the file itself will be made accessible by clicking on the corresponding table entry. By displaying information in this generic way, the actual form of information being displayed will be decoupled from the way in which it is to be displayed.
- Another issue which must be considered is that the user may not want all information to be extracted. For example, if an investigator wants to determine whether a suspect made a call at a particular time, the images stored on the suspect's phone are irrelevant, and need not be acquired. Hence, the ability to selectively choose which information is to be extracted is essential.
- Finally, powerful techniques of organizing and searching through data would be required for this application to be useful, to enable an investigator to quickly find the desired information.

As Figure 26 shows, all elements of the application use a Logger to record important information and events. The logger records any errors which occurred, events such as opening and closing connections, and a listing of all information, in ASCII and hex, which was sent and received to and from the phone(s).

A Sony-Ericsson f500i, Nokia 3220 and Nokia 6225 were used for testing during the development of the application. These phones are shown in Figure 4. Data is retrieved from the f500i using a combination of AT commands and OBEX. This enables the following data to be retrieved:

- General phone information, including manufacturer, phone model, IMEI (the phone's unique ID), IMSI (the SIM card's unique ID) and hardware / software version information.
- The phonebook in both the phone memory and SIM card.

- Sent, received and saved SMS messages in both the phone memory and SIM card.
- Dialled numbers from the phone memory and SIM card, and received and missed calls from the phone memory (this data is not stored in SIM cards).
- Limited portions of the file system. Any user created / downloaded / stored media files (e.g. via the phone camera) will be accessible, along with downloaded ringtones and media.

The application cannot retrieve calendar entries from the f500i, however this is possible via SyncML over OBEX.

Methods used for the Nokia phones are the same, with the addition of Nokia FBUS. The application is able to retrieve the following data from the 3220:

- General phone information.
- The phonebook from both the phone memory and SIM card.
- Sent, received and saved SMS messages from both the phone memory and SIM card.
- Dialled numbers and missed and received calls from both the phone memory and SIM card.
- Limited portions of the file system. As with the f500i, user created / downloaded / stored media and applications will be accessible.
- Calendar entries from the phone memory.

Much the same information can be retrieved from the Nokia 6225; however:

- As the 6225 is a CDMA phone, it does not have a SIM card, and does not support the GSM AT command specifications.
- The 6225 provides no method of obtaining SMS messages.
- CDMA phones are uniquely identified by an ESN, rather than an IMEI.

Most information from the three phones used during the application's development can be acquired. There are, however, some important omissions which illustrate the limitations of a software based data acquisition application, namely that there will always be some data which remains inaccessible:

- Deleted information.
- Operating system files.

The inability of any known method to obtain SMS messages from the Nokia 6225 further highlights this point. Due to the lack of standardization, manufacturers are free to use their own implementations for data access, and leave some data inaccessible. Software based data acquisition will only ever be able to obtain a subset of all the data stored on the phone.

Whether further development of the application will occur remains to be seen. As mentioned in Section 2.3, the original aim of this thesis was the application itself. However, the fact remains that a commercial product, such as Cell Seizure, XRY or otherwise, will provide much better performance, stability and support than what is essentially a prototype, relying on an interpretation of a proprietary protocol (FBUS).

5 Discussion

The results discussed in Chapter 4 highlight a number of limitations and problems inherent in software based data acquisition using the methods discussed:

- Data can be indirectly altered, when using AT commands or Nokia FBUS.
- Important data may be omitted from the phone's response to a command.
- Some data will never be accessible over a software interface.
- Data which is accessible on one phone may not be accessible on other, similar phones, using the same commands.

These issues are discussed in more detail below.

5.1 Limitations in software based data acquisition

As described in Sections 4.1.1 and 4.1.3, when a previously unread SMS message is accessed via AT commands or FBUS, its state will change to indicate that it has been read. This example highlights the fact that the change of data in the phone is not under the PC's control. The same command, issued twice over a small period of time, produces two different results, and the state of the SMS message in question has been permanently changed.

This particular issue could invalidate evidence which relies on a suspect having received and read an SMS message. The suspect could argue that they never actually read the message that was received. While the original evidence would indicate that the message has been read, the suspect could argue that the procedures used to acquire the SMS message modified the data in this way.

This issue arose during a case held in the District Court of South Australia (AUSTLII 2003b). The accused's counsel argued that a received text message found on the accused's mobile phone was prejudiced and should not be admitted, as there was no evidence to suggest that the accused had read or responded to the message. In this case, the text message was allowed to be admitted as evidence, as the fact that the message had been received at all corresponded with other evidence in the case. In the event that knowledge of whether the message was or was not read is important, this problem can be overcome, as the phone's response to the first command shows the original state of the message. Hence, if a log of the communication is used alongside the data obtained from the phone, the data will be shown to be valid.

When using FBUS to retrieve a call log entry, SMS message, or any information which has an associated timestamp, there is no indication of the timezone to which the phone is set, as described in Section 4.1.3. The lack of any indication of the timezone is a serious problem, as it would bring into question the validity of any timestamps retrieved from the phone. For example, if a call log entry indicates that a suspect made a call at 18:25, and a call was received by another suspect at 18:25, the suspect(s) could argue that their phones were set to different timezones (for their own reasons); hence the call times do not match up. The only way of disproving this claim would be through external information, for example from the service provider(s). Weil (2002) stresses the need for multiple, independent sources of time to verify system events; the more sources of time used, the lower the likelihood that the time is incorrect.

The lack of standardisation of Nokia's FBUS protocol is a hindrance to the development of a software application which will work with every Nokia phone. Nokia's official software, PC Suite, comes in several different versions, as two Nokia phones may not be compatible. In addition, some information on a Nokia phone cannot be acquired via FBUS, as mentioned in Section 4.1.3, with regard to the Nokia 6225.

OBEX adds more limitations to the data which can be acquired, as discussed in Section 4.1.2. If a file is flagged with limited distribution rights, it will not be accessible via OBEX. The only method of acquiring files such as images, videos and other user files is through OBEX; if a file is not accessible in this way, the only other method of accessing it will be to manually view it through the phone's user interface. In addition, the authentication mechanism in OBEX, also mentioned in Section 4.1.2, would present another difficulty. If a particular mobile phone requires authentication to carry out an OBEX session, a PIN or password will be required to do so, without which the user files stored on the phone will only be accessible through the phone itself.

The examples given above show that the methods commonly used to acquire data from a mobile phone have some serious limitations. These limitations may prevent all data from being acquired, and may allow arguments to be made against the integrity of data which can be acquired. However, these methods are currently the only way to automate the acquisition of data, which is an essential requirement of the process. Manual browsing of the information stored in the phone can only be used as a last resort, as it is too time consuming to be of practical use.

5.2 Admissibility of evidence acquired from a mobile phone

The legislation outlined in Section 3.3 allows the use of information stored in a mobile phone as evidence. However, the mobile phone may need to be shown to be functioning correctly, and the procedures used to acquire the information must be shown to be accurate. When data stored in a mobile phone is required as evidence, it will typically be obtained via the software methods described in Section 3.2.2. These methods cannot easily be verified to be forensically sound, however, as a mobile phone's software / firmware is proprietary information.

This raises questions as to whether data obtained using these methods is legally admissible in a court of law. It may seem obvious that if the phone returns a certain set of data, then that is what is stored in the phone's memory. However, if it cannot be verified that the phone's software / firmware is accessing the correct area of memory, and is not making changes to that, or other areas of memory, an argument could be made against the integrity of the data acquired from the phone.

It has been shown in Section 4.1 that the use of these methods to acquire data can change information in a mobile phone. In addition to this, important information may not be accessible via these methods. Despite this, if the methods can be validated, and used alongside other sources of information, the information obtained from a mobile phone can be shown to be valid.

In the case regarding the text message found on the accused's mobile phone, mentioned previously, the defense also argued that the message be ruled inadmissible, as, along with other reasons, the message was produced by a computer (either the mobile phone, a SMS message call centre, or otherwise), and the prosecution gave no evidence that the computer(s) involved were operating correctly at the time the message was produced, as required by Section 59B of the Evidence Act 1929 (Parliament of South Australia). The presiding judge made an important decision; that the message could be admitted without conformance to the requirements of Section 59B, and could be admitted at common law.

Common law essentially refers to the interpretation of laws made by judges in past decisions, which then sets precedence for future decisions (Hirst 1998). Therefore, the judge accepted the prosecution's assumption that the text message was valid, as there was precedence from past cases to do so. Such a decision has the effect that when presenting evidence which has been generated by a computer the requirements specified in Section 59B do not necessarily have to be fulfilled. Only when doubt is raised about the correct operation of the computer do the requirements need to be addressed. Therefore, if the computer can be assumed to be operating correctly, only the procedures used to acquire the data can be questioned.

The procedures used to acquire forensic evidence affect the admissibility of the evidence itself. A case held in the Supreme Court of South Australia (AUSTLII 2001) involved the use of DNA analysis to acquire evidence. The defence challenged the admissibility of the DNA evidence on the basis that the software tools used to analyse the DNA samples had not been recognised by the relevant scientific community as reliable, and the expert who presented the evidence was not qualified to use the tools or interpret the results. Direct comparisons cannot be made between physical and computer forensic procedures; however, similar problems may arise when considering evidence from a mobile phone. There are no guidelines on what tools or procedures can be considered reliable, or what constitutes an expert in the area.

From the discussion above, it is clear that the methods of acquiring data from mobile phones need to be verified. This is necessary, in order to prove the accuracy of data obtained from mobile phones using these methods. While the methods will make changes to the data on the phone, it is important to know what changes will occur. If it can be shown that the changes made are unrelated to the data acquired from the phone, the methods used can be shown to be valid.

Verification of these methods could be done in two ways. The first would require phone manufacturers to either release the technical details of their products; this is extremely unlikely to occur. The second approach would require phone manufacturers to test their own products, and release the results of the tests, showing the effects which the acquisition methods have on the phone memory.

There are also other ways to ensure the validity of evidence obtained from a mobile phone. Manufacturers could develop tools which acquire data in a forensically sound manner. This would avoid the need to release proprietary information. If a court case relies on the integrity of information obtained from a mobile phone, an expert witness who has knowledge of the proprietary protocols used, i.e. an employee of the phone manufacturer, to acquire the information could explain why the information is or is not valid.

5.3 Feasibility of a universal software application

The elements of the application for which specifications were available (AT commands and OBEX) were successfully completed, and are able to work with any phone which supports the GSM AT commands or OBEX, respectively. However, as specifications were not available for Nokia's FBUS protocol, this section of the application is lacking. While the FBUS extractor is compatible with the two phones used for testing (the 3220 and 6225), it is unknown whether the range of phones extends beyond them. In addition, other information stored on the phone may be available via FBUS, but the complete set of commands is unknown.

An application which acquires as much data from a mobile phone as is possible via software would be quite easy to develop, provided that the specifications for each communication protocol used are available. Consequently, such an application would not be able to be developed or distributed cheaply, due to licensing restrictions and requirements imposed by the phone manufacturers. It would be extremely unwise to rely on an application which is based on an uninformed interpretation of a proprietary communication protocol, as is the case with the research application's implementation of Nokia FBUS.

This is also the case when considering JTAG. Specifications for a mobile phone's JTAG interface do exist, but are unavailable. Hence, the potential of JTAG for data acquisition is unknown.

6 Conclusion

As stated in the Introduction, the aims of this thesis were as follows:

- To provide an overview and analysis of the methods commonly used to forensically acquire data from mobile phones.
- To determine the limitations of the methods which are discussed, when considering their use in a forensic context.
- To assess the legal admissibility of data, acquired from mobile phones using the methods discussed, as evidence in a court of law.

6.1 Summary

The methods commonly used to acquire data from mobile phones are software based implementations of communication protocols such as AT commands, OBEX, SyncML and Nokia FBUS. An overview of these methods was given. These methods are commonly used as a forensic basis for acquiring data. There are a number of applications which are used to forensically acquire data from mobile phones, however all of the applications found are based on the methods mentioned above.

The main issue with these methods is that they have not been shown to be forensically sound; therefore the integrity of any evidence which is acquired using them may be questioned. Additionally, a number of limitations were discussed, such as the lack of timezone information in Nokia phones, and the lack of standardisation from model to model.

In Australia, information obtained from a mobile phone is admissible as evidence, unless the phone is shown not to be operating correctly. Mobile phone evidence is generally not questioned in court; only a few cases were found which involve any of the issues discussed. However, the procedures used to obtain the evidence can be questioned, and there is currently no scientific basis to rely on these procedures. This may prove to be a hindrance, as in reality, there is currently no way to conclusively prove that information obtained from a mobile phone actually reflects what is stored in the phone.

Nor is there any way to show that the procedures used to acquire the information did not alter any other information stored in the phone.

6.2 Future directions

The main obstacle to this research is the tendency of mobile phone manufacturers to prefer proprietary protocols for phone – PC communication. While there are standards, many phones only provide partial implementations, and in some cases, break rules imposed by the standards. In addition, there is no standard which provides the ability to access all of the memory in a mobile phone.

Currently, a software application which implements the methods discussed is the only way of automating the acquisition of data from a mobile phone. This approach has been shown to have serious limitations, not to mention that the methods used cannot be guaranteed to be forensically sound.

Future research in this area will be limited by the fact that the protocols used for phone – PC communication are unpublished, and not standardized. Industry co-operation may help to overcome this hindrance. Nevertheless, possible areas of research could be assessing the usefulness of the JTAG interface in a forensic context, the development of a communication protocol for complete memory access, and formal verification of protocols such as OBEX and AT commands over a wide range of mobile phones.

Verification of the methods used to acquire data needs to occur, so that the data can be shown to be accurate. This will only occur with cooperation from phone manufacturers. It is not necessary for proprietary information to be released; a phone manufacturer could develop their own tools which can be used to acquire data from their phones in a forensically sound manner. Another option is the use of expert testimony from the phone developers, guaranteeing that the procedures are valid.

7 References

@Stake 2003, 'Security Advisory: Nokia 6210 DoS SMS Issue', online accessed 21st March 2005, http://atstake.com/research/advisories/2003/a022503-1.txt.

Apache 2005 *Xerces* (computer program), http://xerces.apache.org/.

Australian Communications Authority (ACA) 2005, 'Consumer fact sheet: Mobile phone security', online accessed 7th April 2005, http://internet.aca.gov.au/ACAINTER:STANDARD::pp=DIR2_10,pc=PC_1718.

Australian Mobile Telecommunications Authority (AMTA) 2004, 'Industry Statistics Snapshot', online accessed 20th May 2005, http://www.amta.org.au/default.asp?page=327.

Australian Mobile Telecommunications Authority (AMTA) 2005 *Australian Mobile Telecommunications Industry: Economic Significance*, September 2005.

Australasian Legal Information Institute (AUSTLII) 2000 'The Queen v Shawn Dean Roberts [2000] NZCA 235 (12 October 2000)', *The Court of Appeal of New Zealand*, online accessed 22nd March 2005, http://www.austlii.org/.

Australasian Legal Information Institute (AUSTLII) 2001 'R v Karger No. SCCRM-98-224 [2001] SASC 64 (29 March 2001)', *Supreme Court of South Australia*, online accessed 24th July 2005, http://www.austlii.org/.

Australasian Legal Information Institute (AUSTLII) 2003a 'R v Seminara No. DCCRM-02-383 [2003] SADC 56 (11 April 2003)', *District Court of South Australia*, online accessed 21st March 2005, http://www.austlii.org/.

Australasian Legal Information Institute (AUSTLII) 2003b 'R v Koliroff No. DCCRM-01-744 [2003] SADC 31 (4 March 2003)', *District Court of South Australia*, online accessed 23rd September 2005, http://www.austlii.org/.

Black J 2005 'Mobile Phones Ready to Jack Your Data', *Empire Security Corporation*, online accessed 30th March 2005, http://www.empiresecurity.com/hello/Article263.htm.

Commonwealth of Australia, *Evidence Act 1995*.

Compelson Labs 2005 *MobileEdit* (computer program), http://www.mobiledit.com/.

Durda F 2004 'The AT Command Set Reference', online accessed 3rd September 2005, http://nemesis.lonestar.org/reference/telecom/modems/at/history.html.

Ekblom P & Tilley N 2000 'Going Equipped: Criminology, Situational Crime Prevention and the Resourceful Offender', *The British Journal of Criminology*, June, vol. 40, pp. 376-398.

Engstrom M 2003 *pduconv* (computer program), http://www.nerdlabs.org/projects/index.php.

Envisage Systems 2005, *PhoneBase 2*, online accessed 5th April 2005, www.phonebase.info/.

European Telecommunications Standards Institute (ETSI) 1999 *Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS); (GSM 03.40 version 7.4.0 Release 1998).*

European Telecommunications Standards Institute (ETSI) 2003 *Universal Mobile Telecommunications System (UMTS); Discussion of Synchronization Standards (3GPP TR 27.903 version 4.0.0 Release 4).*

European Telecommunications Standards Institute (ETSI) 2004a *Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface (3GPP TS 51.011 version 4.11.0 Release 4)*.

European Telecommunications Standards Institute (ETSI) 2004b *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); AT command set for User Equipment (UE) (3GPP TS 27.007 version 6.7.0 Release 6)*.

European Telecommunications Standards Institute (ETSI) 2005 *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE-DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS) (3GPP TS 27.005 version 5.0.1 Release 5)*.

Etter B 2001 'The forensic challenges of e-crime', *Australasian Centre for Policing Research*, October 2001, no. 3, Adelaide.

Fanflik PL, Johnson JL, Miller ML, Morgan S & Muller S 2004 'If It Sounds Too Good To Be True: Local Prosecutors' Experiences Fighting Telecommunications Fraud', *American Prosecutors Research Institute*, September 2004.

Forensic Telecommunications Services (FTS) 2004 *FTS News*, October/November, Kent.

FusionOne 2005, *MightyPhone*, online accessed 5th April 2005, http://www.mightyphone.com.

Gnokii Project 2005 *Gnokii* (computer program), version 0.6.8, http://www.gnokii.org/.

Goode AJ 2003 'Forensic extraction of electronic evidence from GSM mobile phones', *IEE Seminar on Secure GSM and Beyond,* pp. 9/1-9/6.

Harkin J 2003 'Mobilisation: The growing public interest in mobile technology', *Demos*, June 2003, London.

Hirst J 1998 *Discovering Democracy – A Guide to Government and Law in Australia*, Curriculum Corporation.

Hayes Microsystems 2005, *Glossary*, online accessed 2nd September 2005, http://www.hayesmicro.com/Products/Glossary.htm.

Institute of Electrical and Electronic Engineers (IEEE) 2001 *IEEE Std 1149.1-2001 IEEE Standard Test Access Port and Boundary Scan Architecture*.

insideout Forensics 2005 *SIMCon – SIM Content Controller*, online accessed 4th April 2005, http://www.simcon.no/.

International Organization on Computer Evidence (IOCE) 2000, 'Good Practices for Seizing Electronic Devices', *IOCE 2000 Conference*, Rosny sous Bois, France, 13th – 15th December.

Infrared Data Association (IrDA) 2003, *Infrared Data Association (IrDA) Object Exchange Protocol*, January 3.

Jae-hyun H 2004 'Duplicate Phones Used to Buy Online Goods', *The Chosun Ilbo*, 7th April, Seoul, Korea, online accessed 22nd March 2005, http://english.chosun.com/w21data/html/news/200410/200410070048.html.

Makkai T, Milner L & Mouzos J 2004 '2003 Annual Report on Drug Use Among Police Detainees', *Australian Institute of Criminology*, no. 58, Canberra.

Masnick M 2004 'When your Mobile Phone Represents You', *The Feature*, online accessed 22nd March 2005, http://www.thefeature.com/article?articleid=101131.

Mellars B 2004 'Forensic examination of mobile phones', *Digital Investigation*, vol. 1, no. 4, pp. 266-272.

Micro Systemation 2005 *XRY*, online accessed 10th August 2005, http://www.msab.com/en/product.jsp?categoryId=25&productId=25.

Netherlands Forensic Institute 2005 *TULP2G* (computer program), http://tulp2g.sourceforge.net/.

Nokia 2003 'Nokia PC Connectivity SDK Introduction', *Forum Nokia*, 10th October 2003.

Nokia 2005a *Nokia – Java*, online accessed 14th March 2005, http://www.nokia.com/nokia/0,,402,00.html.

Nokia 2005b *Nokia 9300 SmartPhone*, online accessed 29th March 2005, http://nokia.com.sg/nokia/0,8764,68766,00.html.

Nokia 2005c *Nokia PC Suite* (computer program), http://www.nokia.com/nokia/0,,71516,00.html.

Open Mobile Alliance (OMA) 2002 *SyncML Representation Protocol, Data Synchronization Usage, version 1.1*.

Open Mobile Alliance (OMA) 2005 *Enabler Release Definition for DRM V2.0*.

Oxygen Software 2005, *Oxygen Phone Manager II (Forensic version)* (computer program), http://www.opm-2.com/forensic/.

Paraben Forensics 2005a *Cell Seizure* (computer program), http://www.paraben-forensics.com/.

Paraben Forensics 2005b *Cell Seizure Supported Manufacturers & Models*, online accessed 15th May 2005, http://www.paraben-forensics.com/cell_models.html.

Parliament of South Australia, *Evidence Act 1929*.

Robinson G & Smith G 2001 'Evidence from mobile phones', *Journal of the Institute of Legal Executives*, 1st July.

Russinovich M 1999 *PortMon* (computer program), ver. 3.02, http://www.sysinternals.com/Utilities/Portmon.html.

South Australia Police 2002 *Mind Your Mobile*, online accessed 21st March 2005, http://www.sapolice.sa.gov.au/crime/crime_reduction_section/docs/Mobile%20Phone%20Theft.pdf.

Sony Ericsson 2005a *Sony Ericsson Sync Station 1.5.5.3b* (computer program), http://www.sonyericsson.com/spg.jsp?cc=us&lc=en&ver=4000&template=ps1_1_3_4_1&zone=ps&lm=ps1_1&pid=10139&fid=6995&esi=true.

Sony Ericsson 2005b *Sony Ericsson File Manager 2.6.6.0* (computer program), http://www.sonyericsson.com/spg.jsp?cc=au&lc=en&ver=4000&template=ps1_1_3_1_1&zone=ps&lm=ps1_1&pid=10172&fid=6997&esi=true.

Sun Microsystems 2005 *Java Communications API* (computer program), http://java.sun.com/products/javacomm/.

SourceQuest 2004 *SourceUSB* (computer program), http://www.sourcequest.com/.

Symantec 2005 *Symantec security response: SymbOS.Commwarrior.A*, online accessed 22nd March 2005, http://securityresponse.symantec.com/avcenter/venc/data/symbos.commwarrior.a.html.

Symbian 2004 *Symbian Connect QI SDK* (computer program), http://www.symbian.com/developer/downloads/files/ScQiSdk24.exe.

Symbian 2005 *Symbian OS phones*, online accessed 12th April 2005, http://www.symbian.com/phones/index.html.

Telecommunications Industry Association (TIA) 1999 *Data Service Options for Spread Spectrum Systems: AT Command Processing and the Rm Interface (TIA/EIA/IS-707-A.3)*.

TX Systems 2004 *SIM Manager*, online accessed 4th April 2005, http://www.txsystems.com/sim-manager.html.

WAPForum 2000 *Wireless Internet Today*, June, Mountain View, California.

Weil MC 2002 'Dynamic Time & Date Stamp Analysis', *International Journal of Digital Evidence*, vol. 1, no. 2.

Wiacek M 2005 *Gammu* (computer program), http://www.gammu.net/projects/gammu.php.

Willassen SY 2003 'Forensics and the GSM mobile telephone system', *International Journal of Digital Evidence*, vol. 2, no. 1.

Willassen SY 2005 *Evidence in Mobile Phone Systems*, online accessed 24th March 2005, http://www.mobileforensics.com.

Appendix A: Email correspondence

Correspondence with Micro Systemation regarding XRY.

From: McCarthy, Paul David - mccpd001
Sent: Tuesday, 20 September 2005 1:13 PM
To: '<name withheld>'
Subject: RE: XRY data acquisition methods

Hi <name withheld>,

Thank you for your response, I appreciate it.
I am a graduate student completing my Honours; I have had no prior experience in digital forensics before this year, however it is definitely an interest of mine!
My thesis involves developing a software application, which may end up being used by the South Australian Police.
My thesis will (hopefully) be completed within the next two months, so I will send you a copy when it is complete.

Thanks again,

Paul McCarthy.

-----Original Message-----
From: <name withheld> [mailto:<name withheld>@msab.com]
Sent: Monday, 19 September 2005 6:04 PM
To: McCarthy, Paul David - mccpd001
Cc: <name withheld>@<withheld>
Subject: RE: XRY data acquisition methods

Hi

Pls contact <name withheld>, our partner in Australia. He can demonstrate the system for you. I´d be interested in a copy of your Thesis work when it´s finalized. Are you a forensic specialist working within the police?

We are using AT-Commands, OBEX, F-Bus, IRMC and some commands we have developed ourselves (these I cannot describe since it´s our property of course)

MfG/Best regards

<name withheld>
Area Manager Central Europe

Micro Systemation AB
Råsundavägen 1, Box 3053
SE-169 03 SOLNA, SWEDEN
PHONE: +46 8 739 02 70
FAX: +46 8 730 01 70
MOBILE: +46 70 5352045
E-MAIL: <name withheld>@msab.com
WEBSITE: www.msab.com

-----Original Message-----
From: Paul McCarthy [mailto:mccpd001@students.unisa.edu.au]
Sent: None
To: info@msab.com
Subject: XRY data acquisition methods

Kontaktformulär på www.msab.com

E-postadress: mccpd001@students.unisa.edu.au
För- och efternamn: Paul McCarthy
Tel:
Företag: University of South Australia
Meddelande:
Hi,

I am completing an Honours thesis on mobile phone forensics at the University of South Australia.

I am curious of the methds which your XRY software uses to acquire data from mobile phones (i.e. Nokia FBUS, AT commands, OBEX, JTAG etc).

Any information would be appreciated, however, i understand if you are unable to give this information out.

Regards,

Paul McCarthy.

Correspondence with Envisage Systems regarding Phonebase 2 (no response was received to this email).

From: McCarthy, Paul David - mccpd001
Sent: Wednesday, 17 August 2005 7:40 PM
To: 'info@phonebase.info'
Subject: Phonebase acquisition methods
Hi,

My name is Paul McCarthy, and I am a student at the University of South Australia.

I am doing research into mobile phone forensics, and have been evaluating a number of mobile forensics applications.

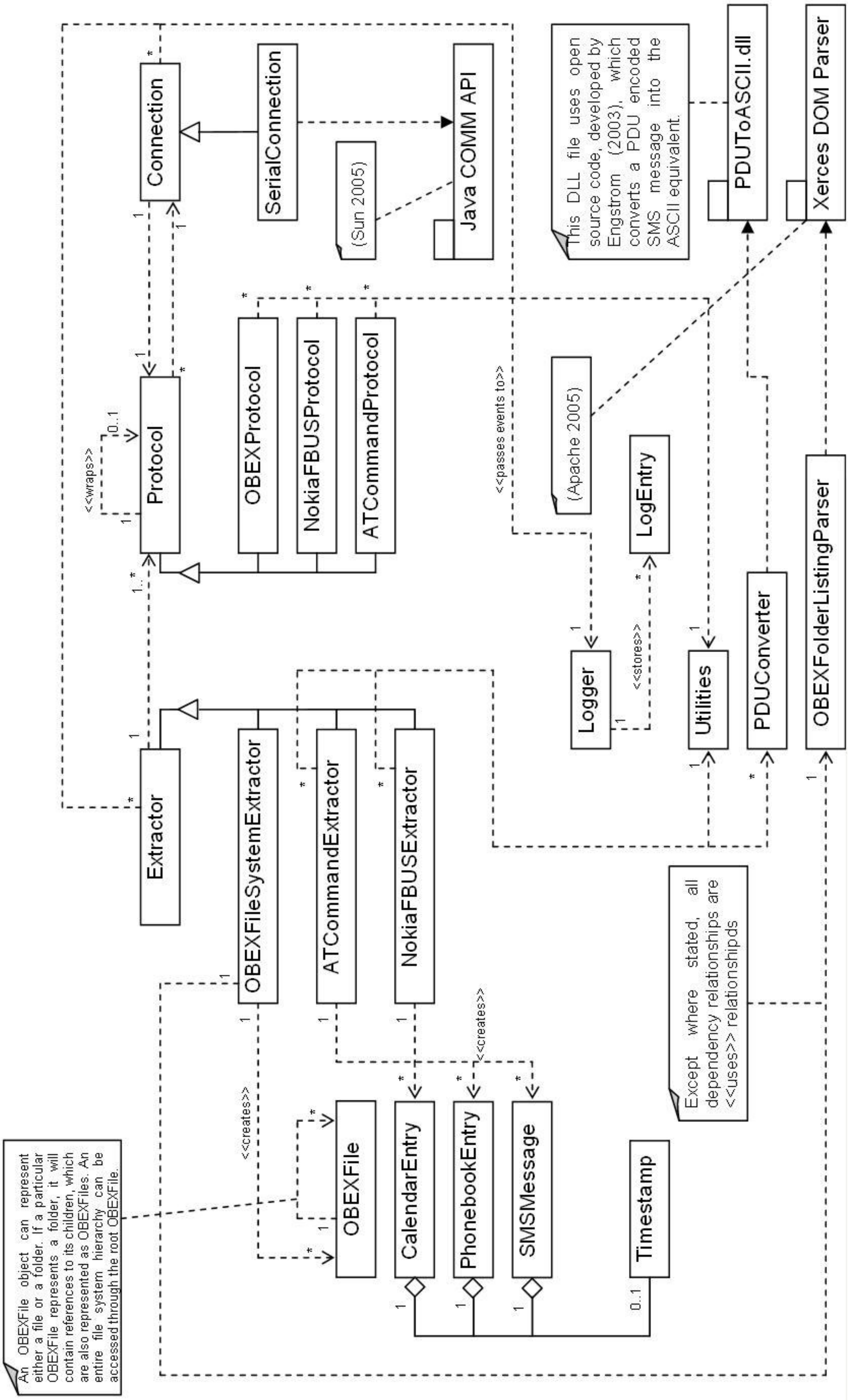
I am curious if it is possible to get some information on the methods that Phonebase uses to extract data from mobiles (i.e. Nokia FBUS, AT commands, OBEX etc).

Any information would be appreciated, however I will understand if you are unable to help me.

Regards,

Paul McCarthy.

Appendix B: Research application UML class diagram



This DLL file uses open source code, developed by Engstrom (2003), which converts a PDU encoded SMS message into the ASCII equivalent.

Except where stated, all dependency relationships are <<uses>> relationshipds

An OBEXFile object can represent either a file or a folder. If a particular OBEXFile represnets a folder, it will contain references to its children, which are also represented as OBEXFiles. An entire file system hierarchy can be accessed through the root OBEXFile.

Appendix C: OBEX over FBUS session initialisation

While not directly related to the thesis topic, the implementation of OBEX present in the two Nokia phones is worth discussing. The process of initialising an OBEX session with a Nokia phone is much more complex than the relatively simple method used by Sony Ericsson phones. The listing below is taken from communication between Nokia PC Suite and a Nokia 6225. Behaviour of the Nokia 3220 in this situation, while not shown, is similar. While this may simply be a misinterpretation, the phone's behaviour during the procedure appears to break several of the rules imposed by the FBUS protocol. This may imply that the implementations of the FBUS and/or OBEX protocols in Nokia phones are incorrect, or unstable. In the listing below, an FBUS session has already been established.

PC:	`1E 00 10 DB 00 10 00 03 00 05 00 00 01 0A 05 D9 10 03 00 00 01 41 1B 5C`

Phone: The first issue to note is that, for unknown reasons, the phone's and PC's sequence numbers are out of sync. The sequence number should increment by one for each frame sent by either party. In this case, they differ by four.	`1E 10 00 7F 00 02 DB 01 C5 6C`
	`1E 10 00 DB 00 07 03 00 00 06 00 01 45 00 58 CB`

PC:	`1E 00 10 7F 00 02 DB 05 D5 78`
Phone: Despite the 'frames to go' byte of the phone's last frame indicating that there were no more frames arriving, the phone has sent another frame. From this point on, it appears that the phone has become the client, and the PC the server, the opposite of a normal FBUS session.	`1E 10 00 D9 00 09 03 00 00 40 00 00 05 01 46 00 5E 81`

PC:	`1E 00 10 7F 00 02 D9 06 D7 7B`
	`1E 00 10 D9 00 08 00 03 00 41 00 00 01 42 0F D1`

Phone:	`1E 10 00 7F 00 02 D9 02 C7 6F`
	`1E 10 00 D9 00 08 03 00 00 61 00 59 01 47 1C BE`

PC:	`1E 00 10 7F 00 02 D9 07 D7 7A`
	`1E 00 10 D9 00 0D 00 03 00 60 00 05 01 00 00 FF FF 01 43 00 B3 4C`

Phone: Here, the phone fails to	`1E 10 00 D9 00 7A 03 08 00 60 00 05 01 00 00 27 D1 CE C0 DE 00 00`

acknowledge the PC's last frame, instead sending its own frame immediately.	00 00 2D 00 00 00 00 08 CB B0 00 00 00 00 01 10 0C 00 00 04 00 09 01 8E 03 28 00 00 00 00 00 22 00 08 00 00 00 43 00 44 00 4D 00 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 31 41 59 27 D1 CE C0 DE 00 00 00 00 2D 01 01 00 00 08 CB B0 00 08 CD BC 09 0C 00 01 00 02 00 00 00 08 02 40 BC F7
PC:	1E 00 10 7F 00 02 D9 00 D7 7D
Phone: The acknowledgement frame from the PC's previous frame has arrived here, out of order.	1E 10 00 7F 00 02 D9 03 C7 6E 1E 10 00 D9 00 5C CD C8 12 03 00 03 00 01 C5 98 00 08 CD D4 01 11 00 04 00 03 00 08 00 08 CD E0 12 03 00 07 00 01 C0 DE 00 00 00 00 09 18 00 08 00 04 00 00 00 09 2E 54 31 41 59 27 D1 CE C0 DE 00 00 00 00 2D 00 00 0A 00 08 CB B0 00 00 00 00 01 10 0C 00 00 04 00 09 01 8E 03 28 00 00 00 00 01 01 61 02
PC:	1E 00 10 7F 00 02 D9 01 D7 7C
Phone: Once again, the 'frames to go' byte of the previous frame indicated that there were no more frames arriving, but the phone has sent another frame.	1E 10 00 D9 00 7A 03 3D 00 60 00 05 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 31 41 59 27 D1 CE C0 DE 00 00 00 29 2D 2F E2 88 00 00 00 00 00 09 26 8C 00 01 FB 98 00 02 33 60 00 09 2F 8C 0C 03 00 03 00 01 00 34 00 00 00 00 09 0E 00 04 00 02 00 00 00 09 2F A4 12 03 00 07 00 01 59 27 00 00 00 00 09 18 02 42 00 01
PC:	1E 00 10 7F 00 02 D9 02 D7 7F
Phone:	1E 10 00 D9 00 5C 00 08 00 04 59 27 00 09 2E 54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 01 03 69 E0
PC:	1E 00 10 7F 00 02 D9 03 D7 7E
Phone: The 'frames to go' problem occurs again here.	1E 10 00 D9 00 08 03 00 FE 46 00 59 01 44 E2 9A
PC:	1E 00 10 7F 00 02 D9 04 D7 79 1E 00 10 D9 00 08 00 03 00 47 00 00 01 44 0F D1
Phone:	1E 10 00 7F 00 02 D9 04 C7 69 1E 10 00 D9 00 08 03 00 00 64 00

	00 01 45 1C E0
PC: The frame sent by the PC here is an OBEX connect request (embedded within an FBUS frame).	1E 00 10 7F 00 02 D9 05 D7 78 1E 00 10 D9 00 2C 00 03 00 20 00 80 00 25 10 00 27 00 46 00 13 F9 EC 7B C4 95 3C 11 D2 98 4E 52 54 00 DC 9E 09 4A 00 0B 50 43 20 53 75 69 74 65 01 45 15 39
Phone: The phone again fails to acknowledge the PC's previous frame.	1E 10 00 D9 00 7A 03 3D 00 60 00 05 00 00 00 02 00 00 00 00 80 00 00 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 64 00 00 00 00 02 01 00 00 01 00 31 41 59 27 D1 CE C0 DE 00 00 00 29 2D 00 00 00 00 00 00 00 00 09 2E 54 00 08 C4 CC 00 02 00 00 00 09 0D 2C 12 03 00 03 00 01 00 00 00 09 00 38 01 11 00 03 00 03 00 00 00 09 0D 44 12 03 00 07 31 41 59 27 D1 CE C0 DE 00 00 02 46 57 00
PC:	1E 00 10 7F 00 02 D9 06 D7 7B
Phone: The acknowledgement to the OBEX connect frame arrives here instead.	1E 10 00 7F 00 02 D9 05 C7 68 1E 10 00 D9 00 5C 00 00 2D 04 59 27 00 09 0D 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 07 66 BC
PC:	1E 00 10 7F 00 02 D9 07 D7 7A
Phone: Another discrepancy concerning the 'frames to go' byte.	1E 10 00 D9 00 26 03 3D 00 20 00 A0 00 1F 10 00 04 00 CB 00 00 00 01 4A 00 13 F9 EC 7B C4 95 3C 11 D2 98 4E 52 54 00 DC 9E 09 01 40 90 5D
PC: Normal, legal communication resumes here.	1E 00 10 7F 00 02 D9 00 D7 7D [OBEX setpath command]

Analysis of this situation can bring about three possible conclusions:

- There has been a severe misinterpretation of the Nokia FBUS protocol. While these discrepancies do not occur in any other situation, this is a possibility.
- The SourceUSB monitor output is inaccurate. This is possible, but unlikely.
- The phone's implementation of either FBUS or OBEX, or the combination of both, is incorrect.

Appendix D: List of known Nokia FBUS commands

The commands listed below have only been tested with two phones: the Nokia 6225, and the Nokia 3220. The commands work with both phones, unless otherwise stated. Whether these commands will work with any other Nokia models is unknown. Some of these commands were interpreted from the Gnokii source code (Gnokii Project 2005), and others were discovered independently.

FBUS commands are organised into different types, indicated by the message type byte. As shown below, there is a type for SMS related commands, phone book related commands and so on.

Message type	Command	Notes
1B	00 01 00 15 01 00	Retrieves the phone model.
1B	00 01 00 07 00 01	Retrieves the software / firmware version information.
1B	00 01 00 00 41	Retrieves the IMEI on GSM phones.
44	00 01 00 00 41	Retrieves the ESN on CDMA phones.
19	00 01 00 0A	Retrieves the phone's current time.
03	00 01 00 03 XX XX 55 55 55 00	Returns the memory status of the phone book (total space and used space). XX XX: memory ID, as follows: dialled numbers: 00 01 missed calls: 00 02 received calls: 00 03 phonebook (phone): 00 05 phonebook (SIM): 00 06
03	00 01 00 07 01 01 00 01 XX XX 00 00 00 00 YY YY 00 00	Returns a phone book entry from the given memory at the given index. XX XX: memory ID (as above). YY YY: index (accessed sequentially).
14	00 01 00 12 00 00	Not supported by the 6225. Returns the names and IDs of all the SMS folders in the phone. Memory IDs are as follows: SIM card: 01 Phone: 02 Default folder IDs are as follows:

		SIM inbox: 02 SIM outbox: 03 Phone inbox: 02 Phone outbox: 03 Phone saved: 04 Phone templates: 05 There may also be user defined folders present in the phone memory.
14	00 01 00 0C XX YY 0F 55 55 55	Not supported by the 6225. Returns the status of the given SMS folder, in the given memory. This command returns the indexes of every message in the folder, as they are not stored sequentially. XX: Memory (as above). YY: Folder (as above).
14	00 01 00 0E XX YY 00 ZZ 55 55	Not supported by the 6225. Returns an SMS message in Unicode from the given memory, in the given folder, at the given index. If the SMS text is longer than a certain amount, the end of the text will not be present in the response. XX: Memory (as above). YY: Folder (as above). ZZ: Index (obtained from previous command).
14	00 01 00 02 XX YY 00 ZZ 01 00	Not supported by the 6225. Same as previous command, but the message is returned in PDU format. The entire message is returned, regardless of the text length.
13	00 01 00 9E XX XX 00 00 00 00 00	Retrieves the memory indexes of calendar entries of the given type. XX XX: Entry type, as follows: 02 00: reminder 00 01: meeting 00 02: call 00 04: birthday 00 08: memo
13	00 01 00 7D 55 55 00 00 00 XX FF FF FF FF	Retrieves the calendar entry from the given index. XX: index (obtained from previous command).
7F		Message type for acknowledgement frame.