2008

Data recovery from PalmmsgV001

Satheesaan Pasupatheeswaran
Edith Cowan University

DOI: 10.4225/75/57b2740440cbf
Originally published in the Proceedings of the 6th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2008.

Data recovery from PalmmsgV001

Satheesaan Pasupatheeswaran

School of Computer and Information Science
Edith Cowan University
Perth, Western Australia
spasupat@student.ecu.edu.au
ptheesan@yahoo.com

Abstract

Both SMS and MMS data analysis is an important factor in mobile forensic analysis. Author did not find any mobile forensic tool that is capable of extracting short messages (SMS) and multimedia messages (MMS) from Palm Treo 750. SMS file of Palm Treo 750 is called PalmMgeV001 and it is a proprietary file system. A research work done to find a method to recover SMS data from PalmMsgV001 file. This paper is going to describe the research work and its findings. This paper also discusses a methodology that will help recover SMS data from PalmMsgV001. The PalmMsgV001 file is analysed using hex analysis method. Solutions were found to recover each message from every folder like Inbox, Outbox, Sentbox, Draft and Template. The research work partially contributes to improving mobile forensic analysis since the finding will be helpful to forensic tool developers. At this stage, this study will concern only the SMS part and not the MMS part.

Keywords

Mobile forensics, Palm Treo 750 SMS file, PalmmsgV001 data recovery

INTRODUCTION

Hardware and software architecture of smart phones, unlike computers, differ from manufacturer to manufacturer. Also the frequency of arrival of new models of mobile phones with new technology is increasing and this becomes a challenge to mobile forensic analysers and forensic tool developers. There is no mobile forensic tool that has the ability to analyse all brands of mobile phones all by itself (McCarthy, 2005). Beyond any argument, analysing SMS file is significant in mobile forensics because it may contain valid evidences.

Palm Treo 750 mobile phone uses a new SMS technology called threaded SMS technology. At the time of writing this paper, the author did not find any commercial or open source tool that helps to analyse message file of Palm Treo 750 also referred to as Palmmsgv001. Aim of the research is to find a method to retrieve SMS data from PalmMsgV001file. Binary analysis and binary comparison method is used during the research work. The research work is concentrated on SMS data recovery from all text message folders such as Inbox, Outbox, Sentbox, Draft and Template.

According to the ACPO (2003), any operation performed on original evidence should not alter the evidence. If alteration is necessary, it should be documented and its impact on evidence should be realized (ACPO, 2003). Due to some limitations in accessing mobile phone data acquisition tools and accessing mobile devices, the file acquisition was not done in a forensically sound manner. However, this did not significantly affect the research results because the goal of the research was not forensic analysis.

Significance

Lack of knowledge about messaging file (Palmmshv001) of Palm Treo 750 imposed limitations on mobile forensic analysis. Nowadays, communication via text messages and exchanging multimedia messages is quiet common because of its several attractive features like sending a single SMS to multiple recipient and low cost. Several crimes such as drug trafficking and pornography are also committed via text and multimedia messaging (Press, 2008). Therefore analysing messaging file during mobile forensic analysis is crucial.

Research questions

The analysis work is focused on answering the following questions.

i. How to identify an empty file?

ii. How to identify and retrieve read, unread, saved, deleted, drafted and template messages?

iii. How to identify and retrieve source of message and other relevant contact information?

iv. In some occasions there may be identical message from a particular source. If so how to distinguish messages?

SHORT MESSAGE SERVICE

SMS is an abbreviation for short message service, it first appeared in 1992 in Europe (Bodic, 2005). Since then it has become hugely popular among mobile users. SMS application is widely used for multiple purposes such as person to person communication, information services, email alerts, business cards, chat applications and other co-operate services such as sim updates, unlocking mobile to use on any network and remote monitoring (Bodic, 2005).

SMS provides a way to transfer short message between two entities via service centre (SC). SC provides relaying service for short messages. There are two types of SMS transport services. A point to point SMS service and point to Omni-point service. Point to point SMS is a service between two parties. Point to Omni-point (cell broadcast) SMS service is sending message to multiple recipients from a single source (ETSI, 1999). Point to point SMS service contains two types of basic services such as short message point to point mobile terminated (SM-MT) and short message point to point mobile originated (SM-MO).

SM-MT is a GSM service that handles delivery of a short message from SC to mobile station (MS). SM-MO is a GSM service that handles submition of short messages from MS to another mobile entity through SC. Both services are capable of providing delivery status report (ETSI, 1999). SC provides store and forward service so as to give a reliable service (Trosby, 2004).

SMS size is very limited; a single SMS size is 140 octets. It supports only 160 characters if 7 bit character encoding is used and 70 characters if 16 bit Unicode encoding is used (ETSI, 1999).

Chat Application

Chat application helps to chronologically display the communication history between two parties. Most of the mobile chat applications use SMS for message transport (Bodic, 2005).

APPARATUS

During the research process both software and hardware tools were used for data collection and analysis. Following paragraphs discuss each tool in detail.

Dopod 838 Pro smart phone

A Dopod smart phone was used for data collection. It was running on WM 5.0. With a few modifications this mobile was made to operate in threaded style SMS. These modifications make the Dopod's SMS application to replicate a Palm Treo 750. The message application file is placed in a windows folder.

'ExamDiff Pro' Software

ExamDiffPro is a hex file comparison tool. Version 4.0 of this tool was used for the research. This program supports opening two files in a window. This tool shows edited characters, deleted characters and added character in different colours in both files. It also provides a navigation facility that helps to jump to locations of modified data. It has facilities like file swapping, simultaneous scrolling of both files and hex to text/text to hex toggle ("Visual File And Directory Comparison Tool," 2008).

“UltraEdit” software

Version 14.10 of UltraEdit Professional was used in this analysis. UltraEdit is a powerful text, hex and software source code editor. During hex edit, it displays both its hex and its ASCII values. It supports toggling between hex to text and text to hex ("UltraEdit Text Editor Features," 2008).

PROCEDURE

Data collection

The Palmmsgv001 file is accessed from the desktop and copied from the mobile to the desktop by synchronising the mobile phone with the desktop.

Reference image

A reference point is necessary so as to make sure each test phase is started from the same environment. Starting each phase of test from a reference point will help to protect the file from previous data content. An empty Palmmsgv001 is kept as reference file. Each phase of test is started with empty file acquisition.

Sample data preparation

Several samples were created in a way to address the research questions.

ANALYSIS OF SAMPLE FILES

Analysis is performed in two stages. Each stage is discussed below. Analysis was done based on ETSI TS 100 901 short message standards (ETSI, 1999).

Stage1

At first empty file of each test phase was compared with the help of ‘ExamDiff Pro’. The aim of this comparison was to identify header bytes.

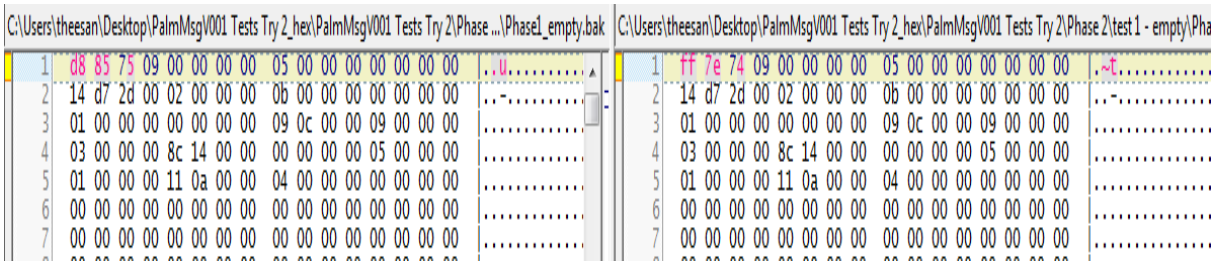

Figure1: Comparison of two empty file

The above figure 1 shows comparison of two empty files. Bytes, those are different between the two files are coloured in pink.

It is found that the first three bytes are changing from file to file. Observations in all phases, helped to identify two pairs of identical files.

Then empty file of each phase was compared with the relevant data files that contained known data such as unread file, read file, saved file and deleted file. First few lines were observed for changes. It was found that first three lines had changed bytes. Closer observation in all phases helped to identify a byte which is used to indicate file status ,i.e. whether the file is empty or not. There were some other bytes also changed from file to file but the changes were not consistent. Figure 2 shows a screen shot of changed bytes in the first three lines.

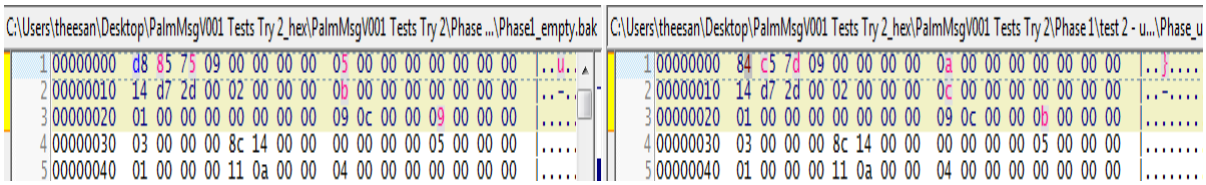

Figure2: File comparison shows changed bytes.

Further analysis helped to identify a set of four bytes that indicate whether the message data structure is empty or not. Figure3 shows the set of four bytes and its ASCII character in blue font.

Figure3: Empty message data structure and non empty data structure

Stage2

During this stage, each sample file was opened with 'UltraEdit' in order to perform pattern analysis. Each file was manually analysed. A column that represents ASCII characters of hex values was used to identify the known data. This analysis helped to find another data structure that contains all SMS data, MMS data and relevant folders. This data structure has a special signature. Also it was found that the data structure has several small data units each units has a header and a footer. Mostly all footers have the same bytes but some differ. Also a data unit was identified that is used as a separator and it is used to separate two different sections.

The footer was used to break the data structure into small data units, then the data structure was analysed. From the analysis it was found that, at first the data structure initialises all folders like 'Outbox', 'Sentbox', 'Savedbox', 'Draftbox' and 'Template'. Two repeating patterns were observed during each folder initialisation, those patterns were identified as start of folder and initialisation part. These patterns are included within blue coloured boxes in figure4. Figure4 shows a small data unit, its header and footer are highlighted. The header value is the signature of the data structure. This data unit includes Outbox initialization part also.

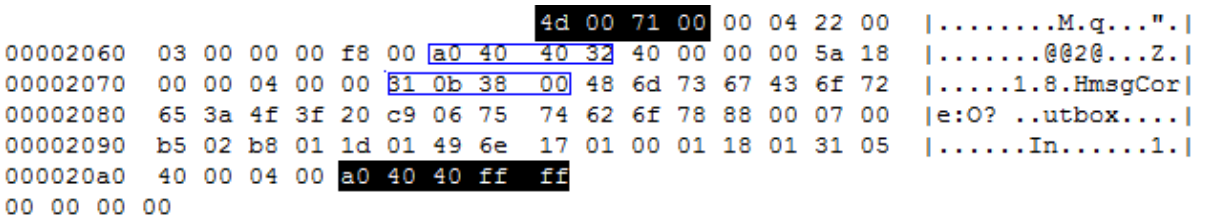

Figure4: Small data unit with header and footer that initialize 'Outbox'

Figure 5 shows the separator data unit that always start with a particular 4 bytes header and ends with footer. Separator data unit consists of 21 bytes.

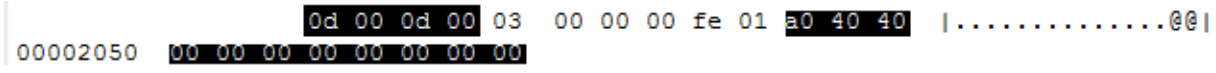

Figure5: Separator data unit

Further analysis helped to find the folder data structure. Each folder is also identified as a data unit. Folder data structures start with 4 bytes header. The header bytes are not always consistent, it takes different byte value for each messages and the reason was not found. A set of bytes, which are used to indicate the start and the end of SMS text, was identified. Similarly another set of bytes were identified, which are used to indicate the start and the end of the source mobile number. Two sets of unique bytes were identified in inbox folder, one set of the unique bytes was used to link data that is relevant to a particular SMS and the other set of bytes was used as a unique inbox message-id followed by a pattern that was used as end of SMS text part.

Further analysis helped to identify how to distinguish between messages from different folders. The information of the Folder, to which the SMS belongs to, was appended at the end of data unit.

Two different patterns, which were used to indicate starting of the source mobile number, were identified. One pattern was used to refer to the source within folders and the other pattern was used to refer the source in other data units such as contact list.

Another data unit was identified and that was used as contact list. The contact list data unit contains source mobile number, sender's first name, surname and others. The pattern used to separate each detail of the sender was also identified. A byte that was used to rank SMS data was also identified.

Figure6 shows message data structure of the inbox for a single SMS. Similar structure repeats for every SMS record.

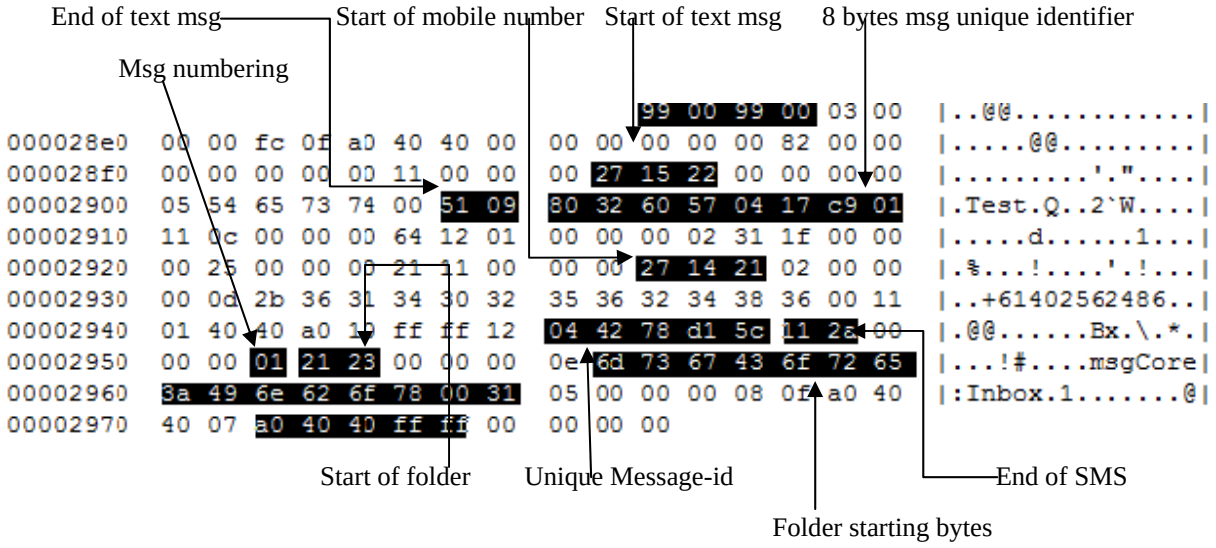

Figure6: Inbox data structure

The above figure shows that the message belongs to the inbox folder. The 'message numbering' part indicates position of message in inbox folder. Figure7 below shows detail description of a contact list.

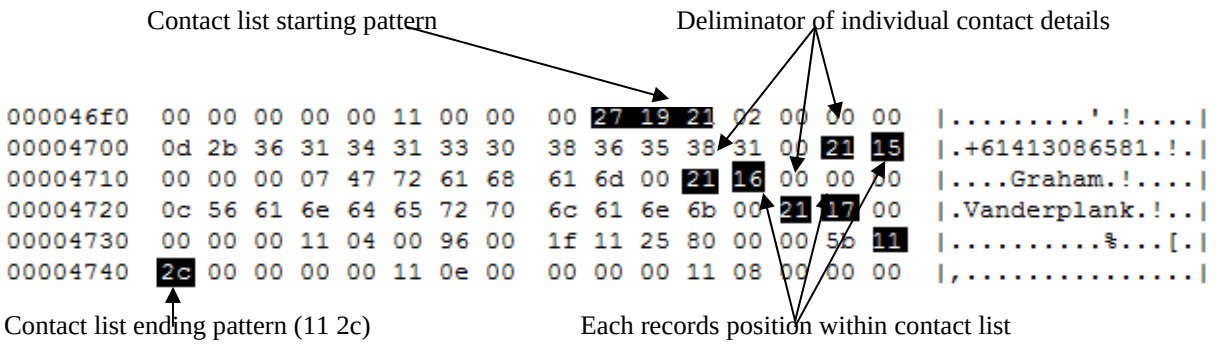

Figure7: Contact list structure

Abbreviations used in coming sections are listed in below table.

ABBREVIATIONS

Abbreviation	Full form	No of bytes
CD	Contact Detail	Varies (Specify by CDL)
CDL	Contact Detail Length	1
CLSP	Contact Detail Starting Pattern	3
DA	Destination Address	Varies (Specify by MNL)
DUSP	Data Unit Start Pattern	3
ECL	End of Contact List	2
EMID	End of Message Identifier	2
EMNP	End of Mobile Number Pattern	2
EOF	End Of File	5
EOR	End Of Record	2
FL	Folder Information	Varies
IMID	Inbox Message Identifier	4
LFL	Log File Length	1
Log	Log file	Varies (Specify by LFL)
LSP	Log Start Pattern	2
MC	Message Core: xxxx	Varies (Specify by MCL)
MCL	Message Core Length	1
MD	Message Data	Varies
MID	Message Identifier	8
MN	Mobile Number	Varies (Specify by MNL)
MNL	Mobile Number Length	1
MP	Message Position	1
PB	Padded Bytes	Varies(Specified at locations wherever necessary)
SFLP	Start of Folder Pattern	2
SMIDP	Start of Message-ID Pattern	2
SMNP	Start of Mobile Number Pattern	3
SNCD	Start of Next Contact Detail	2
UD	Used Data	Varies (Specify by UDL)
UDEP	Used Data End Pattern	2
UDL	User Data Length	1
UDSP	User Data Start Pattern	Either 1 or 3 depends on folder

Table1: Abbreviations with their number of bytes

RESULTS AND ANSWERS TO RESEARCH QUESTIONS

Stage1

i. How to identify an empty file?

File status byte is stored at Offset 0x18. If its value is 0x0b then the file is empty and value of a non empty file is 0x0c. In addition to the offset 0x18, an empty files contain a unique hex pattern “63 4d 6b 71” somewhere within the file.

Stage2

From stage2 analysis, the purpose of this research is achieved. Below paragraphs discuss all findings and methods of retrieval.

ii. How to identify and retrieve read, unread, saved, deleted, drafted and template messages?

File status flag, which indicates whether file is read or not, were not identified. However, it was observed that if the file was an unread SMS it keeps the database at offset 0x1000, once the SMS is read it moves it to different location, mostly to offset 0x2000. But these offsets were not consistent, and so it is not possible to exactly identify unread and read SMS messages.

Other activities like saved, deleted, drafted and template messages were identified and recovery methods also found. Below paragraphs discuss identification and recovery process.

Reading messages from inbox

Inbox data structure is comprised of message data (MD), source address (SA) and folder information (FL). Figure8 shows main parts of inbox data structure and each part is magnified.

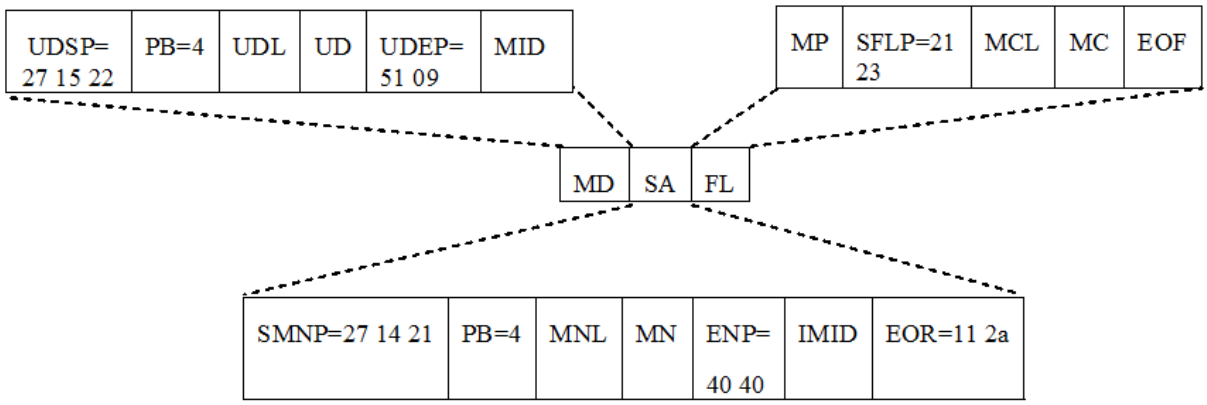

Figure8: Inbox data structure with magnified elements

a) User data (UD) starts with a hex pattern of “27 15 22” then neglect the next four bytes following it. Fifth byte denotes length of user data (UDL). The text message starts from the sixth byte. The text message ends with a hex pattern of “51 09”(UDEP).

b) After the end of text message, the next consecutive eight bytes are the unique message identifier (MID). It is used to refer this text message in other areas. This will help to identify the order of SMS and its relevant contact list data unit. And also these 8 bytes always end with a hex pattern of “c9 01”. This might contain time related information but at this time it was unable to confirm.

c) After the unique message identifier, the source mobile number starts with a hex pattern of “27 14 21”(SMNP). The next four bytes are padded with zeros. Then the fifth byte indicates the mobile number length (MNL) and the next consecutive bytes are the source mobile number. This mobile number ends with a hex pattern of “40 40” (ENP).

d) A hex pattern “11 2a” is used to indicate the end of message record (EOR).

e) A set of four bytes that forms a unique inbox message identifier (IMID) was identified. This IMID is located just before the end of message record (EOR) pattern. This IMID appears for messages within the inbox only.

f) After the end of the message, the record folder information starts with a hex pattern of “21 23”(SFLP). Next three bytes are neglected. Fourth byte indicates length of message core (MCL). Message core contains name of folder.

g) The byte before the pattern “21 23” indicates the message records position in inbox (MP).

h) Inbox folder ends with a hex pattern of “a0 40 40 ff ff”.

Reading message from outbox

Out box record structure consists of destination address (DA), message identifier (MID), message data (MD) and folder information (FL).

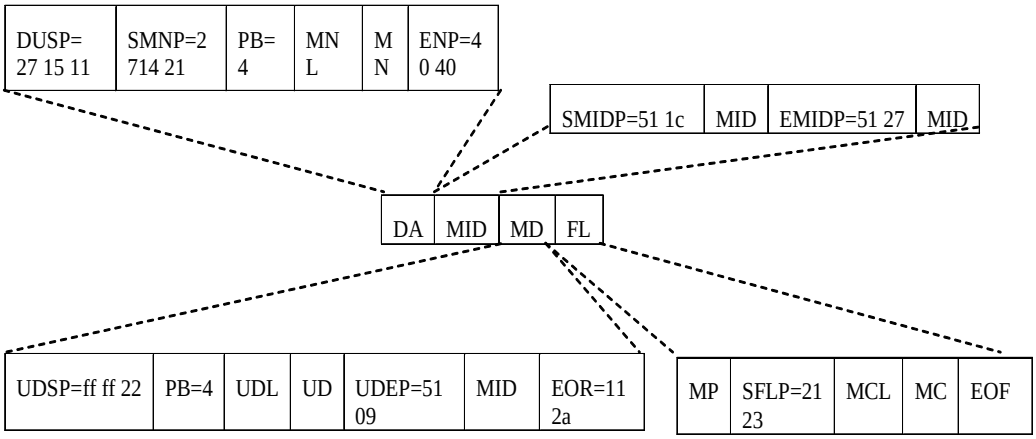

Figure9: Outbox data structure with magnified elements

a) Like inbox, destination mobile number starts with a hex pattern of “27 14 21” and fifth byte from SMNP denotes mobile number length then mobile number starts at sixth byte and ends with a hex pattern “40 40”.

b) Then the unique message identifier starts with a hex pattern of “51 1c”(SMIDP) and the next set of eight consecutive bytes is the unique message identifier. This identifier ends with a hex pattern “51 27” (EMIDP).

c) Then the user data starts with a hex pattern of “ff ff 22”. Next four bytes are padded with 0s or any other hex values and the fifth byte specify length of user data (UDL). The user data (UD) starts immediately after the UDL and ends with a hex pattern “51 09”.

d) Then next 8 bytes are the same unique message identifier followed with the end of record pattern “11 2a”.

e) Folder information starts and ends as stated in inbox part.

Reading saved message

Reading ‘saved folder’ is similar to reading inbox but there are a few difference like text message starting pattern and folder information are different and saved messages do not have unique inbox message identifier (IMID). Inbox data structure folder information part contains “msgcore:Inbox” but ‘saved’ data structure folder information part contains “msgcore:Saved”. User data starting pattern for saved message is “02 22”.

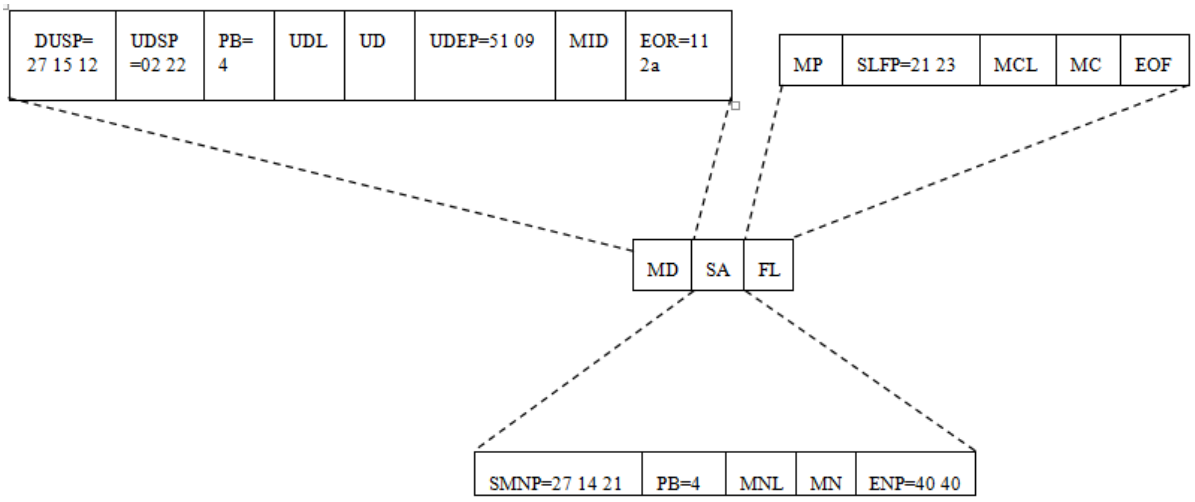

Figure10: Saved message data structure

Reading sent message

This folder contains log entry of sending attempt and relevant text message with destination mobile number. As outbox data structure, ‘sent message’ data structure also starts with destination mobiles number.

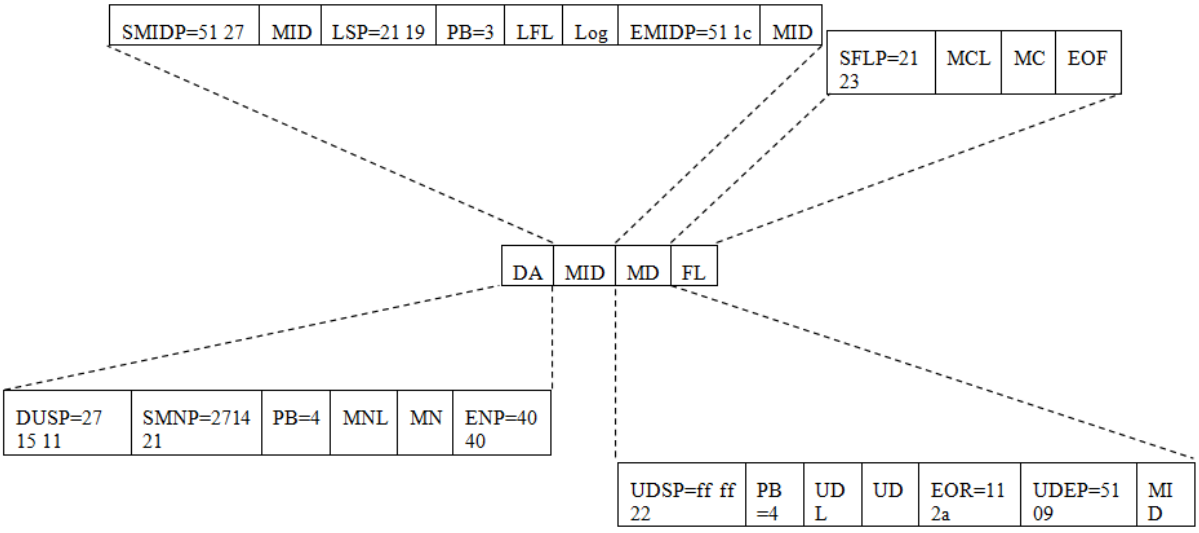

Figure11: Sentbox data structure with magnified elements

Destination mobile number start and end pattern are same as outbox structure.

a) Starting and ending pattern structure of MID of ‘sentbox’ are “51 27’and “51 1c” respectively. The MID section of ‘sentbox’ also contains log entries of sending attempts.

b) Next consecutive eight bytes from SMIDP are unique message identifier.

c) Log file starts (LSP) with a hex pattern of “21 19”. Then next three bytes are padded with 0s. Fourth byte shows log file length (LFL). Log information follows immediately after LFL. Eight bytes followed by EMIDP are message-id.

d) The user data starts with a hex pattern “ff ff 22”. Next four bytes are padded with 0s or any other hex values. Fifth byte denotes length of user data and the user data starts from 6th byte.

e) End of record (EOR) pattern appears before the end of user data pattern (UDEP).

f) The 8 bytes immediately after UDEP are MID.

g) Sent box does not have a byte to indicate a message position (MP) within ‘sentbox’ folder.

Reading drafted message

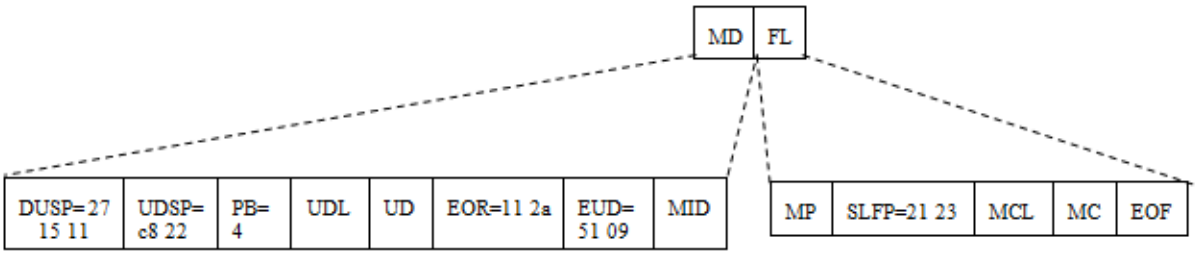

Figure 12: Drafted message format

a) Draft data structure is identified with a hex pattern “27 15 11”.

b) Similar to saved message its text UDSP is identified by the pattern “c8 22”.

c) End of text pattern “51 09” appears earlier than end of message record pattern.

d) The 8 bytes unique message identifier is placed before the starting pattern of folder information.

e) Folder information part (FL) does not have a message position byte (MP).

Reading message templates

Template data structure is exactly same as ‘draft’ data structure but it varies at folder information part only. Folder information part contains ‘msgcore: Template’.

Reading Deleted messages

Deleting operation does not actually delete text message from the message file. The message is still in the file but a flag is set to not to display the message in folders. But the flag was not identified during this research. Normal inbox reading procedure will read the deleted messages also. Following will help to identify deleted message.

a) Deleted message is still in inbox. So reading procedure is same but it lacks the source mobile number part.

b) The folder information part is repeated twice . As usual, first folder information ends with footer then again the folder information starts with a starter byte pattern “23” and ends with footer.

Reading contact list

iii. How to identify and retrieve source of message and other relevant contact information?

Contact list or phone book contains source or destination contact details such as mobile number, first name, last name and others. The first record of contact list is mobile number.

CLSP=27 19 21	PB=4	CDL	CD1	SNCD= 21 xx	PB=3	CDL	CD2	SNCD=27 xx+1	PB=3	CDL	CD3

SNCD=27 xx+n	PB=3	CDL	CDn	ECL=11 2c

Figure 13: Contact list format

Contact list starts with a hex pattern "27 19 21" (CLSP). The next 4 bytes are padded. The fifth byte is used to specify the first contact detail length (CDL). Then first contact detail starts from the sixth byte, usually the first contact detail is a mobile number.

Next contact detail starts (SNCD) with a hex pattern of "21 xx" where "xx" increments with number of contact details. The 'xx' is also used to arrange the contact details. In this analysis xx started from 0x15.

Then the next three bytes are padded with zeros. Again fourth byte is CDL then second contact detail. Then again start of third contact detail if available. Otherwise contact list is terminated with a hex pattern of "11 2c"(ECL).

iv. In some occasions there may be identical messages from a particular source. If so how to distinguish between messages?

Unique message identifier (MID) or unique inbox message-id (IMID) of SMS record in inbox can be used to distinguish messages, even if those messages were from same source and contains same data.

PRACTICAL DIFFICULTIES

Palmmsgv001 file repeats the message database at least twice and each SMS record several times within a database. Any attempt to read palmmsgv001 file will retrieve duplicate records of each SMS. Sometimes a SMS message may undergo several operations such as save and delete; if message retrieval procedure sorts out SMS records based on folder name then inbox folder, save folder and delete folder will have the same message. It is important to mark duplicate messages during retrieval procedure.

Unique message identifier will help to tag duplicate SMS records and also it will help to mark same SMS record in all folders. To find deleted messages form recovered inbox message this unique identifier can be used.

Encountered Issues

This analysis did not help to find timing parameters. It was assumed that timing parameter must hide within unique message identifier (MID). Time parameter is an important parameter in forensic analysis. Message identifier was analysed to find any time related information. Due to time restriction, message-id was not fully analysed. Analysing the file may reveal time related information.

It was identified in most of the samples, when the SMS was unread the message database appeared at offset 0x1000. Once it is read the message database was moved to offset 0x2000. But some sample files with unread SMS showed different behaviour, it stored message database to some other location. Moreover no any special flag were found to distinguish read message from unread message or vice versa.

It was observed that with multiple SMS records, message data base did not show its signature and in some cases even it did not include the folder initialisation part. These issues, however, did not affect the aim of the research.

CONCLUSION

Thread SMS is a new technology in mobile phone SMS message service and Palm Treo 750 SMS service. The research work on palmmsgv001, message file of Palm Treo 750, revealed that it is possible to recover all message data from Palmmsgv001 message file. A methodology, based on pattern search, to recover data from PalmMsgV001 is also proposed.

This research work did not find message timing parameters. It is suspected that the unique message identifier may be constructed with time information but not resolved. Timing parameters are important parameters in forensic analysis. This emphasizes a future research to determine timing parameters. There are still several information needs to be identified such as group-id, to identify a message from same source, and flag that indicates message status.

The research work was limited with SMS recovery but the message file contains MMS (Multi Media Service) data also. Future research is needed to study MMS parameters so as to recover MMS data.

REFERENCES

ACPO (2003). Good Practice Guide for Computer based Electronic Evidence. *Journal*. Retrieved from http://www.acpo.police.uk/asp/policies/Data/gpg_cpmputer_based_evidence_v3.pdf

Bodic, G. L. (2005). *Mobile Messaging Technologies and Services: SMS, EMS and MMS* (2nd ed.): John Wiley and Sons.

ETSI. (1999). *ETSI TS 100 901 V7.4.0 (1999-12): Digital cellular telecommunications system (Phase 2+);Technical realization of the Short Message Service (SMS);*

(GSM 03.40 version 7.4.0 Release 1998). Retrieved 18 July, 2008, from http://amber.feld.cvut.cz/user/pokorny/bpdp/GSM_0340.PDF

McCarthy, P. (2005). *Forensic Analysis of Mobile Phones*. Retrieved 18 Aug, 2008, from http://esm.cis.unisa.edu.au/new_esml/resources/publications/forensic%20analysis%20of%20mobile%20phones.pdf

Press, T. A. (2008). Dozens of college students busted in drug sting. Retrieved 28/9/2008, from http://www.msnbc.msn.com/id/21134540/vp/24487634#24487634

Trosby, F. (2004). *SMS, the strange duckling of GSM*. Retrieved 22 July,2008, from http://www.telenor.com/telektronikk/volumes/pdf/3.2004/Page_187-194.pdf

UltraEdit Text Editor Features. (2008). Retrieved 5 Aug,, 2008, from http://www.ultraedit.com/products/ultraedit/ultraedit_features.html

Visual File And Directory Comparison Tool. (2008). Retrieved 25 Aug,, 2008, from http://www.prestosoft.com/edp_examdiffpro.asp