Digital Forensic on MTK-based Shanzhai Mobile Phone with NAND Flash

Mengfei He[*], Junbin Fang[#&], Zoe L. Jiang[*1], S.M. Yiu[#], K.P. Chow[#], Xiamu Niu[*]
[*]Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
[#]Department of Computer Science, The University of Hong Kong, Hong Kong
[&]Jinan University, Guangzhou, China

Abstract
Mobile phone has become a necessity of our life. There exist hundred kinds of Chinese Shanzhai mobile phones and they had an important impact on the mobile industry and the society. There is also a trend that Shanzhai phones are used in crimes as they are much cheaper and hard to be traced. The adverse impact on forensic is the difficulty of obtaining useful evidence from these phones due to the absence of system manuals and knowledge of the memory layout. In this paper, we attempt to provide some important information of how the phone book, phone call records, SMS, web browser record etc. are stored inside a MTK-based Shanzhai phone with NAND flash and how this kind of Shanzhai phone handle these important data. This information can help investigators understand the working mechanisms of Shanzhai phone and analyze the problems encountered during investigation.

Keywords
Chinese Shanzhai mobile phone, NAND flash, phone book, phone call records, SMS, web browser

1. Introduction

The use of mobile phone has increased dramatically in the last decade. Globally, the number of mobile cellular subscriptions reached 5.3 billion by the end of 2010, reported by the International Telecommunications Union (ITU)[1]. Mobile phones have been part of people's daily life. With the improvement of performance and functionality, activities can be engaged by mobile phone increase rapidly, from making a phone call to browsing webpage, reading email, enjoying multimedia etc., which inevitably keep records of people's actions, whereabouts, habit, and intentions. Particularly, it can also be used as a criminal tool anytime and anywhere, which leads to the necessity of mobile phone forensics. Benefit from the integrated development environment provided by MediaTek (MTK) [2] and Spreadtrum [3], *Chinese Shanzhai mobile phone* (Shanzhai phone for short) has had a huge commercial market in China and overseas in recent years due to its high price/performance ratios. There is an increasing trend that these Shanzhai phones are found to be used in many crime cases. However, there has been little published research on Shanzhai phone forensics due to the lack of system manuals and knowledge of the memory layout.

Over 90 percent of Shanzhai phones are using the integrated platform developed by MediaTek (MTK) or Spreadtrum, including the core processor, the peripheral hardware prototype, the software platform and the SDK (Software Development Kit). Similar to other smart phones, it uses flash memory as the internal data storage, which is currently the most dominant

[1] Corresponding author. Email: zoeljiang@gmail.com

non-volatile solid-state storage technology for mobile phone.

In the paper, we provide important information of how a MTK-based Shanzhai phone with NAND flash stores the phone book, the phone call records, SMS, web browsing record etc. in its internal flash memory. This information can help investigators understand the working mechanisms of Shanzhai phone and analyze the problems encountered during investigating. The rest of the paper is organized as follows. Section 2 reviews the current work related to mobile phone forensics. Section 3 describes the format of phonebook, phone call record, SMS, web browser record etc. and their addition/deletion characteristics. Section 4 concludes the paper.

2. Related work

There has been some research on mobile phone forensics since early 2000s. From the operating system point of view, there have been various forensic software or tools aiming at dedicated operating systems, such as Symbian [4], Windows mobile [5], Android [6]. Since these tools are operating system dependent, they cannot be used to acquire data from Shanzhai phones. Zhang [7] proposed a method to recover MTK mobile phone flash file system, however, no detailed information is given. Fang et al.[8] analyzed the phone book, phone call record of a MT6253 chip based Shanzhai phone. However, since the phone under test is a low-end model and equipped with a NOR flash of 16MB, which is somewhat backward people’s demand for capacity. In our paper we analyze the phone book, phone call record, SMS, web browser etc. of MT6235 chip based Shanzhai phone, which uses NAND flash as basic storage medium and has larger capacity.

3. Digital forensics procedure

Our work is carried out on a model of Shanzhai phone which is a fake version of Apple’s iphone4. This model is equipped with a MediaTek MT6235 processor and a 132MB NAND flash chip (HY27xA081G1M/A). NAND flash is another type of flash different from NOR flash. The NAND type is primarily used in memory cards, USB flash drives, solid-state drives, and similar products, for general storage and transfer of data. Our first task is to retrieve a data image of the internal memory chip. Then the data dump will be analyzed to extract the information for forensic investigation.

3.1 NAND flash image acquisition

Basically, there are three methods for acquiring binary image from mobile phone [8, 9], Flasher Tools, JTAG, and Physical Extraction. Considering the complexity, reliability and other reasons, we choose the first approach to acquire data.

3.2 Phone book storage structure and characteristics analysis

Phone book is a basic data type in mobile phone. We first inserted a phonebook entry with the name of “ANDY1” and the number of “8976357”. Then used hex editor WinHex to investigate the image and found a phonebook entry stored in the following format, as show in Figure 1.

```
073198E0  00 00 00 00 00 00 04 A4  41 00 4E 00 44 00 59 00  .......  .N.D.Y.
073198F0  31 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  1...............
07319900  00 00 00 00 00 00 00 00  38 39 37 36 33 35 37 00  ........8976357.
07319910  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
07319920  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
07319930  00 00 36 A4 00 00 00 00  00 00 00 00 00 00 00 00  ..6?..........
```

Figure 1. An example of phonebook entry in the binary image

The length for one phonebook record is 74 bytes and is different from that in Ref. [8]. As shown in Figure 1, beginning at address 0x073198E8, 10 bytes of UCS2 characters are used to record the name of the phonebook entry (ANDY1, “41 00” is the UCS2 code for “A” etc). At address 0x07319908, 7 bytes, indicating the phone number, with an ASCII coding scheme.

In Ref. [8], Fang et al.’s found that the characteristics of wear leveling of NOR flash will lead to many snapshots of the historical operations. Referred to their experiment, we designed the following experiment for MT6235 with NAND flash.

Table1. The operations performed on the phonebook data

Step*	Operation	Name	Phone number
1	Add one entry	ANDY	8976356
2	Add one entry	ANDY1	8976357
3	Add one entry	ANDY2	8976358
4	Delete one entry	ANDY1	8976357
5	Add one entry	ANDY3	8976359

*we acquire the image after each step

However, the results are very different from Fang et al.’s. In our results only one snapshot can be found. The following binary images record our experimental results. Image 1 and Image 2 corresponds to memory dumps after Step 4 and 5, respectively. In the fourth step, we delete a phonebook entry. The experiment shows that the phonebook entry still exists with the first letter of “ANDY1” filled by 0x00 in Image 1. After the fifth step, the newly added phonebook entry, “ANDY3”, overwrite the previously deleted one, “ANDY1”, as shown in Image 2. However, we cannot find the phonebook data which was appeared in Image 1 at address 0x073CF090, where has been filled by 0xFF in Image 2. This indicates the previous operation trace has been erased.

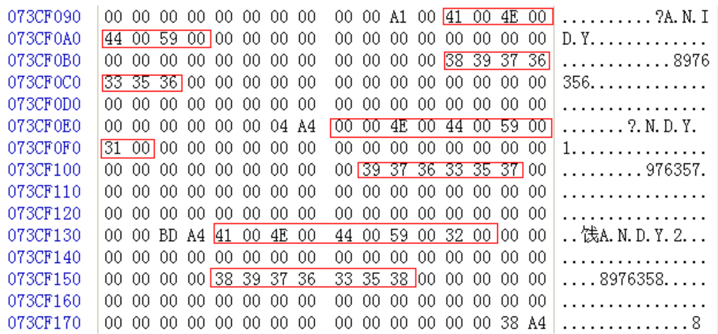

Image 1

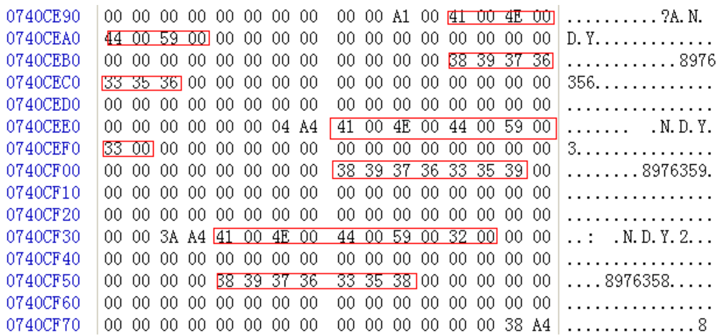

Image 2

Figure 2. Binary images of phonebook experiment

From the experiment above, we have the following observations.

(1) Deleted phonebook entry will not be overwritten until a new phonebook entry is added;

(2) Newly added phonebook entry is stored just behind the previously added entries;

(3) Any modification on phonebook will lead it to update its storing position and the previous one will be emptied. We do not find any snapshot related to our historical operation.

All these indicate that the mechanism of MT6235 is indeed different from MT6253.

3.3 phone call record storage structure and characteristics analysis

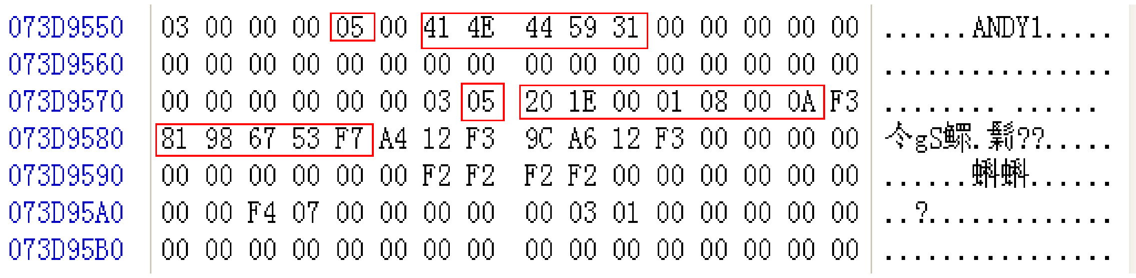

Figure 3. Examples of calling logs in the binary image

Phone call record is another basic data type in mobile phone. After adding the phonebook entry “ANDY1”, we make a call using it, and search the phone call record related to this phonebook entry.

Beginning from the address 0x73D9554, one byte indicates the length of bytes for storing phone name. The following one byte is the encoding mechanism of phone name, followed by the exact phone name. Beginning from the address 0x73D9577, one byte indicates the length of bytes for storing phone number. The following seven bytes represent the time and date, followed by the exact phone number with BCD coding scheme. The length for one phone call record entry in MT6235 is 92 bytes.

Similar to the analysis of phonebook, we designed an experiment shown in Table 2. Due to the similarity to phonebook, we ignore the images.

Table 2: The operations performed on the phone call record

Step*	Operation	Phone name	Phone number
1	Dial a phone	ANDY1	8976357
2	Dial a phone	ANDY2	8976358
3	Dial a phone	ANDY3	8976359
4	Delete a phone call record	ANDY2	8976358
5	Dial a phone	ANDY4	8976360

* We acquire the image after each step

From the above experiment designed, we observe that

(1) When deleting one phone call record, all other below it will be moved up one position;

(2) Newly added phone call record will be placed to the topmost

(3) Any change to the phone call record will lead the entire call log change its storage position and the previous one will become empty. Similar to the phonebook, no snapshot appears in our experiment.

3.4 SMS storage structure and characteristics analysis

SMS contains important information and is an essential part of mobile phone forensics. SMS uses the standard PDU format. Received SMS and sent SMS are in different formats.

```
0733EA50  03 08 91 68 31 08 70 55  05 F0 24 0D 91 68 51 18  ..愸1.pU.?.愸Q.
0733EA60  55 94 84 F9 00 00 21 70  10 90 94 93 23 07 7A F3  U攋?.!p.恨?.z
0733EA70  19 BD 36 E7 01 FF FF FF  FF FF FF FF FF FF FF FF  .??
0733EA80  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0733EA90  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0733EAA0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0733EAB0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0733EAC0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0733EAD0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0733EAE0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0733EAF0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0733EB00  00 00 00 00 86 C2 00 00  00 00 00 00 00 00 00 00  ....喀..........
```

Figure 4. Binary image of a received SMS

Beginning at 0x0733EA50 in Figure 4, one byte indicates the status of SMS (“03”, not viewed; “01”, viewed; “05” sent SMS), followed by one byte indicating the length to store SMS Center information that stored right behind. As shown in Figure 5, “91 68 31 08 70 55 05 F0” represent the SMS Center information, “91” is an international phone indicator and “68 31 08 70 55 05 F0” are the SMS Center phone number with BCD coding scheme. At address 0x0733EA5B, one byte indicates the length of sender’s phone number, denoted as “Address_Len”. According to this value, we can calculate the bytes for storing the sender phone number (the bytes for storing sender phone number equals to (Address_Len+1)/2). The sender number stored at address 0x0733EA5C with BCD coding scheme. Beginning at 0x733EA65, one byte indicates the SMS data coding scheme of SMS (“00”, bit-7; “04”, bit-8; “08”, ucs2), denoted as TP_DCS. The following seven bytes are the time stamp information. Beginning at 0x733EA6D, one byte indicates the length of SMS data (if TP-DCS field indicates 7-bit data, the length here is the number of septets. If the TP-DCS field is set to indicate 8-bit data or Unicode, the length would be the number of octets). After that is the SMS data. There is no SMS Center information and time stamp information in the format of sent SMS.

We designed the following experiment for SMS.

Table 3. Experiment on SMS

Step*	Operation	Phone number
1	Receive a SMS	15815549489
2	View a SMS	15815549489
3	Receive a SMS	1065712035030104
4	Receive a SMS	10086
5	Delete a SMS	1065712035030104
6	Receive a SMS	18718672692
7	Send a SMS	15815549489

* We acquire the image after each step

From the above experiment designed, we observe that no snapshot generated during the whole experiment, all the SMS are stored together (received SMS, sent SMS, draft SMS).

3.5 Web browser record analysis

With the popularity of mobile internet, people's habits are changing. People have become accustomed to using mobile phones to browse the Web, so forensic work toward web browser could make a difference. Generally, the website we visited will be recorded in two places, one only record the user typed website, we denote as P1, the other record both user typed and go through the hyperlink, we denote as P2. There is some difference between the record format in these two places, the following two figures give an example of “www.baidu.com” stored in P1 and P2.

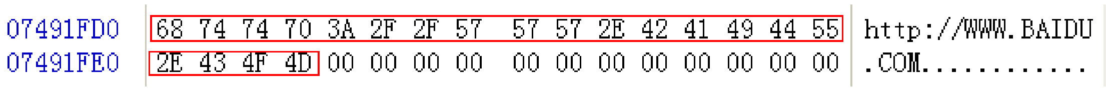

Figure 5. “www.baidu.com” in P1

Record in P1 only contains the URL, and ends with null character.

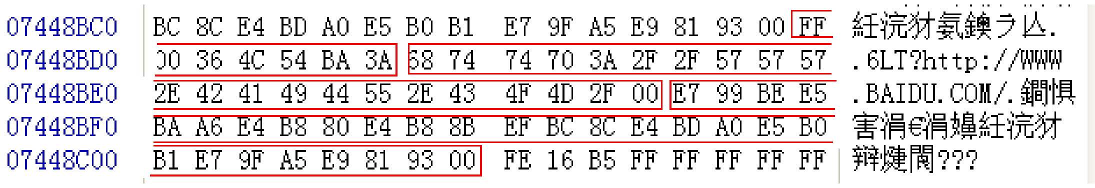

Figure 6. “www.baidu.com” in P2

A record in P2 can divided into the header, website and caption three parts. The header, 7 bytes, the third byte of the header indicate the length of bytes from the fourth byte of the header to the end of the record, we denote its value as VH3. Then we can calculate that the size of a total record equals to VH3+3.The header is stored at 0x07448BCF~0x07448BD5 in Figure 6. The website part stored right behind the header, end with null character. The last part is the caption, use utf8 coding scheme. In Figure 6, starting from 0x7448BEC to 0x7448C7 is the utf8 code of “百度一下，你就知道”.

We design the following experiment to investigate the addition/deletion characteristics of web browser record in P2.

Table 4. Experiment on web browser record in P2

Step*	Operation	Website
1	Visit	http://www.baidu.com
2	Visit	http://www.soso.com:80/?t=04964
3	Visit	http://wap.sogou.com/sogou/?fr=s-sogou&clk=s-sogou
4	Delete	http://www.soso.com:80/?t=04964
5	Visit	http://dh.sogou.com/guide?m=cla&nid=1&cl=soxs&from=sogou&v=2&uID=JkmvQBDK299yhm9h

* We acquire the image after each step

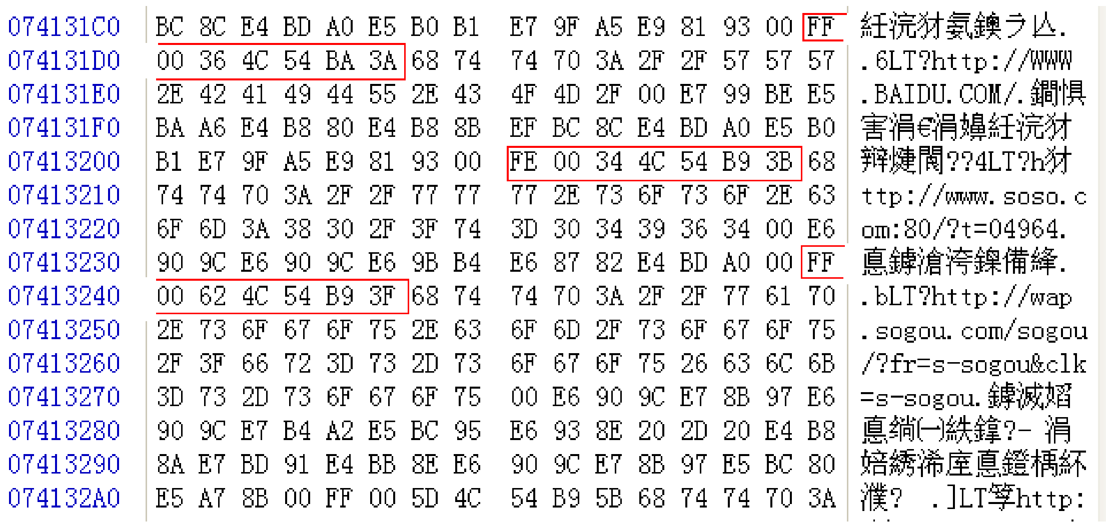

Image 3

Image 4

Figure 7. Binary images of web browser record experiment

Image 3 and 4 in Figure 7 correspond to memory dumps after Step 4 and 5, respectively. We marked the header of each web browser record with rectangle. Note that the locations of all the web browser records are changed without snapshots kept. Still we cannot find any snapshot related to historical operation in Image 3. The fourth step is to delete a web browser record. Then we can see the deleted web browser record keeps unchanged with its first byte replaced by 0xFE in Image 3. At last we visit a website, as shown Image 4, the newly visited web site is just placed in the bottom, but not overwrites the deleted one.

From the experiment above, we can get the conclusion that

(1) When we delete a web browser record, only its first byte is replaced with 0xFE;

(2) The newly generated web browser record is just placed in the bottom

(3) Any operation could lead all the record in P2 change its location and no snapshot is generated.

3.6 Analysis of operations on files

The storage area of Shanzhai phone is divided into system area and user area. The user file area is directly accessible for normal users through the OS of the mobile phone and is used to store the photos taken by the phone camera, the files downloaded using the mobile network, etc. When the mobile phone is connected to a PC with a data cable, the user file area works as an external storage in Windows OS. As shown in Figure 8, this area in the device under test is about 58.5M bytes.

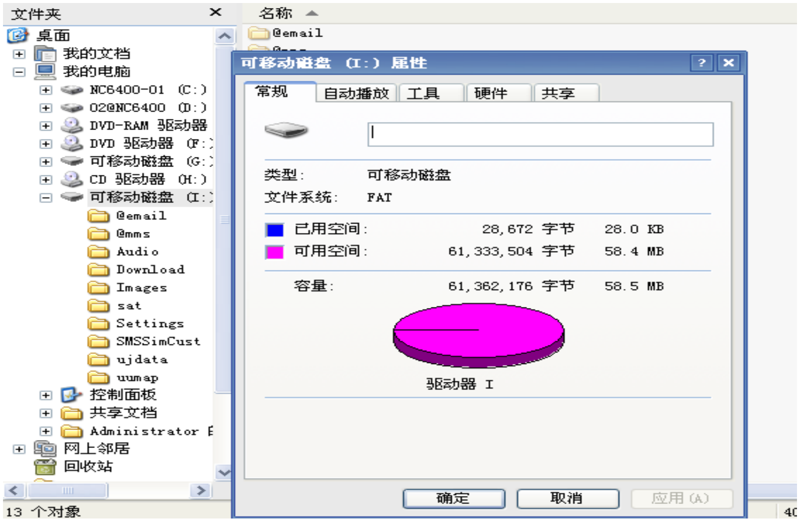

Figure 8. The directory of the user file area of the Shanzhai phone

View the DBR (Dos Boot Record) of this external storage with WinHex as shown in Figure 9.

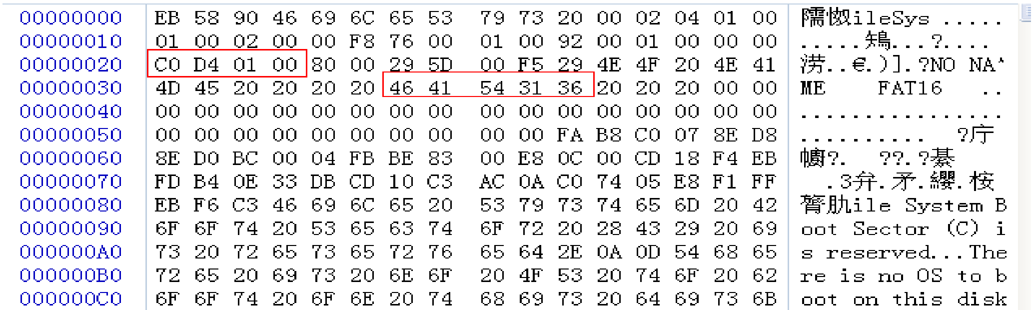

Figure 9. DBR in user file area

Starting at 0x00000020 in Figure 9, four bytes indicate the number of sectors and stored in reversed manner. So,“C0 D4 01 00”represent 0001D4C0. According to this, we can calculate that the storage capacity of this external storage equals to 0x0001D4C0*512=58.5MB, which is consistent with the data present in Figure 8. At address 0x00000036, five bytes show the file system description.

We designed the following experiment. Note that the size of the four files is all 920 bytes.

Table 5. Experiment on file operation

Step	Operation	File description & operation description
1	Add TestA.txt Add TestB.txt Add TestC.txt Add TestD.txt	The content of TestA.txt is “WHYA***AWHY”(914 A) The content of TestB.txt is “WHYB***BWHY”(914 B) The content of TestC.txt is “WHYC***CWHY”(914 C) The content of TestD.txt is “WHYD***DWHY”(914 D)
2	Delete TestA.txt Overwrite TestB.txt Modify TestC.txt Modify the attribute of TestD.txt	 Overwrite TestB.txt with 0xFF Replace ’C’ in TestC.txt with ’F’ Change the creation time of TestD.txt to20/3/2010

* We acquire the image after each step

```
045F9DA0  54 45 53 54 41 20 20 20  54 58 54 20 00 28 5C 4B  TESTA   TXT .(\K
045F9DB0  DD 40 DD 40 00 00 E7 4A  DD 40 02 00 98 03 00 00  軥軥..鍷軥..?..
045F9DC0  41 54 00 65 00 73 00 74  00 42 00 0F 00 37 2E 00  AT.e.s.t.B...7..
045F9DD0  74 00 78 00 74 00 00 00  FF FF 00 00 FF FF FF FF  t.x.t...    ..
045F9DE0  54 45 53 54 42 20 20 20  54 58 54 20 00 33 5C 4B  TESTB   TXT .3\K
045F9DF0  DD 40 DD 40 00 00 04 4B  DD 40 03 00 98 03 00 00  軥軥...K軥..?..
045F9E00  41 54 00 65 00 73 00 74  00 43 00 0F 00 13 2E 00  AT.e.s.t.C......
045F9E10  74 00 78 00 74 00 00 00  FF FF 00 00 FF FF FF FF  t.x.t...    ..
045F9E20  54 45 53 54 43 20 20 20  54 58 54 20 00 3E 5C 4B  TESTC   TXT .>\K
045F9E30  DD 40 DD 40 00 00 1C 4B  DD 40 04 00 98 03 00 00  軥軥...K軥..?..
045F9E40  41 54 00 65 00 73 00 74  00 44 00 0F 00 DF 2E 00  AT.e.s.t.D...?.
045F9E50  74 00 78 00 74 00 00 00  FF FF 00 00 FF FF FF FF  t.x.t...    ..
045F9E60  54 45 53 54 44 20 20 20  54 58 54 20 00 49 5C 4B  TESTD   TXT .I\K
045F9E70  DD 40 DD 40 00 00 31 4B  DD 40 05 00 98 03 00 00  軥軥..1K軥..?..
```

Image 5. FAT information with four test files

```
035A3180  57 48 59 41 41 41 41 41  41 41 41 41 41 41 41 41  WHYAAAAAAAAAAAAA
035A3190  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA.
```

Image 6. Storage location of TestA.txt

```
035A39C0  57 48 59 42 42 42 42 42  42 42 42 42 42 42 42 42  WHYBBBBBBBBBBBBB:
035A39D0  42 42 42 42 42 42 42 42  42 42 42 42 42 42 42 42  BBBBBBBBBBBBBBBB:
```

Image 7. Storage location of TestB.txt

```
035AC600  57 48 59 43 43 43 43 43  43 43 43 43 43 43 43 43  WHYCCCCCCCCCCCCC
035AC610  43 43 43 43 43 43 43 43  43 43 43 43 43 43 43 43  CCCCCCCCCCCCCCCC
```

Image 8. Storage location of TestC.txt

```
035ACE40  57 48 59 44 44 44 44 44  44 44 44 44 44 44 44 44  WHYDDDDDDDDDDDDD
035ACE50  44 44 44 44 44 44 44 44  44 44 44 44 44 44 44 44  DDDDDDDDDDDDDDDD:
```

Image 9. Storage location of TestD.txt

Figure 10. File storage location and FAT after Step 1

```
04616BA0  E5 45 53 54 41 20 20 20  54 58 54 20 00 28 5C 4B  錏STA   TXT .(\K
04616BB0  DD 40 DD 40 00 00 E7 4A  DD 40 02 00 98 03 00 00  軥軥..鍷軥..?.._
04616BC0  E5 31 00 30 00 37 00 30  00 45 00 0F 00 C4 30 00  ?.0.7.0.E...?...
04616BD0  38 00 46 00 37 00 31 00  32 00 00 00 00 00 FF FF  8.F.7.1.2.....
04616BE0  E5 30 37 30 45 30 7E 31  20 20 20 20 00 33 5C 4B  ?70E0~1    .3\Kr
04616BF0  DD 40 DD 40 00 00 43 54  DD 40 03 00 D1 5E 00 00  軥軥..CT軥..裇..
04616C00  41 54 00 65 00 73 00 74  00 43 00 0F 00 13 2E 00  AT.e.s.t.C......
04616C10  74 00 78 00 74 00 00 00  FF FF 00 00 FF FF FF FF  t.x.t...    ..
04616C20  54 45 53 54 43 20 20 20  54 58 54 20 00 3E 5C 4B  TESTC   TXT .>\K
04616C30  DD 40 DD 40 00 00 38 54  DD 40 04 00 98 03 00 00  軥軥..8T軥..?..n
04616C40  41 54 00 65 00 73 00 74  00 44 00 0F 00 DF 2E 00  AT.e.s.t.D...?..
04616C50  74 00 78 00 74 00 00 00  FF FF 00 00 FF FF FF FF  t.x.t...    ..
04616C60  54 45 53 54 44 20 20 20  54 58 54 20 00 00 5D 4B  TESTD   TXT ..]K
04616C70  74 3C DD 40 00 00 31 4B  DD 40 05 00 98 03 00 00  t<軥..1K軥..?...
```

Image 10. FAT information with four test files

```
0412C980  57 48 59 41 41 41 41 41  41 41 41 41 41 41 41 41  WHYAAAAAAAAAAAAA
0412C990  41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
```

Image 11. Storage location of TestA.txt

```
04603200  57 48 59 46 46 46 46 46  46 46 46 46 46 46 46 46  WHYFFFFFFFFFFFFF
04603210  46 46 46 46 46 46 46 46  46 46 46 46 46 46 46 46  FFFFFFFFFFFFFFFF
```

Image 12. Storage location of TestC.txt

```
04603A40  57 48 59 44 44 44 44 44  44 44 44 44 44 44 44 44  WHYDDDDDDDDDDDDD
04603A50  44 44 44 44 44 44 44 44  44 44 44 44 44 44 44 44  DDDDDDDDDDDDDDDD
```

Image 13. Storage location of TestD.txt

Figure 11. File storage location and FAT after Step 2

From the experiment above we can see the deleted files still exist in image, but the first letter of its record in FAT is changed to 0xE5. Any modification on file can lead the file to change its storage position. There is no snapshot generated in our experiment. Sometimes reboot can also lead the files to change its storage position. This may be caused by the wear leveling characteristics of flash. Because of the FTL, files that logically contiguous are always not contiguous in our physical image dump.

4. Conclusion

This paper presents a preliminary work on the investigation of how phone call records, phone book entries, SMS, web browser, etc. are stored in a MT6235-based Shanzhai phone with NAND flash and their addition/deletion characteristics. We have seen the differences between MT6235 and MT6253 in processing data. MT6235 does not generate snapshots. The investigation will be helpful when we encounter to this type of chip during forensic investigation. Future work includes (1) trying to get a more detailed allocation architecture of the system for phone calls, phone book entries, SMS, and other related information; and (2) further analysis on the Spreadrum-based Shanzhai phone which is another popular platform for Shanzhai phone.

Reference

[1] International Telecommunications Union. (2010). The world in 2010: ICT facts and figures. http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf (http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf)

[2] MediaTek. http://www.mediatek.com/en/index.php

[3] Spreadtrum. http://www.mediatek.com/en/index.php

[4] P. Mokhonoana and M. Olivier. Acquisition of a Symbian Smart Phone's Content with an On-Phone Forensic Tool, *Southern African Telecommunication Networks and Applications Conference 2007 (SATNAC 2007) Proceedings*, 2007, pp. 24-32.

[5] C. Klaver. Windows Mobile advanced forensics, *Digital Investigation*, Issue 6, 2010, pp. 147-167.

[6] T. Vidas, C. Zhang and N. Christin. Toward a general collection methodology for Android devices, Digital Inverstigation, Vol. 8, supplement, 2011, pp. S14-S24.

[7] Zhi-wei Zhang. The research of MTK mobile phones flash file system recovery, *Netinfo Security*, issue 11, 2010, pp. 34-36.

[8] Junbin Fang, Zoe Jiang, Kam-Pui Chow, Siu-Ming Yiu, Lucas Hui and Gang Zhou. MTK-based Chinese Shanzhai Mobile Phone Forensics, *Eighth Annual IFIP WG 11.9 International Conference on Digital Forensic*, 2012, pp. 1-9.

[9] Marcel Breeuwsma, Martien de Jongh, Coert Klaver, Ronald van der Knijff and. Mark Roeloffs. Forensic Data Recovery from Flash Memory, Small Scale Digital Device Forensics Journal, Vol. 1, No. 1, 2007, pp. 124-132.