Chapter 9

FORENSIC ANALYSIS OF PIRATED CHINESE SHANZHAI MOBILE PHONES

Junbin Fang, Zoe Jiang, Kam-Pui Chow, Siu-Ming Yiu, Lucas Hui, Gang Zhou, Mengfei He and Yanbin Tang

Abstract Mobile phone use – and mobile phone piracy – have increased dramatically during the last decade. Because of the profits that can be made, more than four hundred pirated brands of mobile phones are available in China. These pirated phones, referred to as "Shanzhai phones," are often used by criminals because they are inexpensive and easy to obtain. However, the variety of pirated phones and the absence of documentation hinder the forensic analysis of these phones. This paper provides key details about the storage of the phonebook and call records in popular MediaTek Shanzhai mobile phones. This information can help investigators retrieve deleted call records and assist them in reconstructing the sequence of user activities.

Keywords: Chinese Shanzhai phones, forensic analysis, phonebook, deleted data

1. Introduction

The use of mobile phones around the world has increased dramatically. According to the ITU, the number of global mobile subscribers reached 5.3 billion in 2011. During the first quarter of 2011 alone, vendors shipped 371.8 million units, an increase of 19.8 percent over the previous year [11].

Because of their portability and constant use, mobile phones hold information about user activities, contacts and whereabouts. This can be a treasure trove of evidence in criminal investigations. Traditionally, data that can be recovered from a mobile phone includes the phonebook, call logs, short message service (SMS) messages [8], and possibly even deleted items. Smartphones also provide e-mail, multimedia files and web browsing data. The data is stored on various storage media, includ-

Figure 1. Chinese Shanzhai phone (fake version of iPhone 4).

ing the SIM card, internal flash memory and external memory cards [12]. A variety of forensic tools are available for extracting evidence from SIM cards and external memory cards. However, extracting evidence from internal flash memory is more challenging.

Chinese Shanzhai mobile phones developed by MediaTek (MTK) [6] and Spreadtrum [9] have gained a massive market share due to their low cost, high performance and ease of purchase. For example, the fake iPhone 4 shown in Figure 1 costs just $130 and low-end versions can be purchased for as little as $60.

As expected, more and more Shanzhai phones are being used by criminals. However, there has been little published research related to Shanzhai phone forensics. One reason is that the huge popularity of Shanzhai phones was entirely unexpected. Another is that there is almost no official documentation about the phones, especially regarding flash memory, file systems and other details about how the phones store data.

About 90 percent of Shanzhai phones are based on "turn-key" solutions developed by MTK and Spreadtrum. The turn-key solutions are development platforms that cover mobile phone hardware, operating system and software, including the core processing chipset, peripheral hardware prototype, operating system, software platform and software development kit. As with other smartphones, internal flash memory is the major data storage component in Shanzhai phones, but extracting evidence from the flash memory of Shanzhai phones is very challenging because of the variety of phones and the absence of documentation.

This paper is, to our knowledge, the first formal attempt to address Shanzhai phone forensics. It provides key details about the storage of the phonebook and call records in MediaTek Shanzhai mobile phones. This information can help investigators retrieve deleted call records and assist them in reconstructing the sequence of user activities.

2. Related Work

Research in mobile phone forensics was initiated in the early 2000s. Since then, a wide range of mobile forensic tools have been developed to acquire data from the flash memory of mobile phones [5]. However, most of the tools use commands and protocols that indirectly access the memory. In particular, they rely on the operating system, which means that the phone must be operational and only data that is visible to the operating system can be recovered – deleted data is not recoverable. Also, because of the manner in which they operate, the tools may change the memory contents.

Flasher tools provide the easiest non-invasive means of reading flash memory data [2]. They have been used in a number of mobile forensic examinations [3]. However, like traditional mobile forensic tools, flasher tools cannot ensure a complete dump of the memory and rely on the operating system. Also, if the data connector of the mobile phone is not supported by the flasher tool or the pins of the data connector are not connected directly to the pins of the main processor, then the pins on the printed circuit board must be manually connected to the flasher tool, a task that involves considerable electronic work and one that could possibly damage the device.

Physical extraction involves desoldering the internal flash memory chip from the mobile phone and acquiring the data using a chip memory reader. The extraction procedure requires considerable expertise and skill because the memory chip is easily damaged during desoldering.

Another physical extraction technique uses Joint Test Action Group (JTAG), a standard developed to automatically test the functionality and quality of integrated components on printed circuit boards. In this technique, JTAG test access ports are used to put the microprocessor in the debug mode and communicate with the memory chip, enabling the

Table 1. Comparison of internal memory acquisition techniques.

	Desoldering	**JTAG**	**Flasher Tools**
Chip Damage Risk	High	Low	Low
Data Modification Risk	Low	Low	Medium
Complexity of Use	High	Medium	Low
Electronic Soldering	High	Medium	–
Data Completeness	High	Medium	Medium

memory to be dumped bit-by-bit [12, 13]. The advantages of physical extraction are that it does not rely on the operating system and ensures that the entire binary image is extracted.

Table 1 compares the three internal memory acquisition techniques discussed above. In our experiments, we opted for the easier technique involving the use of a flasher tool because the focus was on understanding the memory storage organization. Note, however, that the JTAG technique provides a better low-level picture of mobile phone memory.

Several mobile phone forensic tools have been developed for specific operating systems such as Symbian [7], Windows Mobile [4] and Android [10]. However, since these tools are operating system dependent, they cannot be used to acquire data from Shanzhai phones.

With regard to MTK mobile phones, Zhang [13] has written a paper about recovering data from a flash file system. However, very few technical details are provided in the paper.

3. Chinese Shanzhai Phones

The minimal architecture of an MTK-based Shanzhai phone includes the MTK core chipset and an internal flash memory chip. The core chipset integrates a microprocessor unit, radio frequency transceiver module, GSM/GPRS module, multimedia modules, power management module, etc. The flash memory chip is usually integrated with random access memory (RAM) to conserve space on the printed circuit board. Note, however, that depending on the type of the flash memory, the memory allocation architecture for an MTK-based Shanzhai phone can be very different.

There are two main types of flash memory architectures: NOR flash and NAND flash. NOR flash memory can be read byte-by-byte in constant time. NOR flash memory is typically used for direct code execution, but some of the memory may be reserved for data storage (Figure 2).

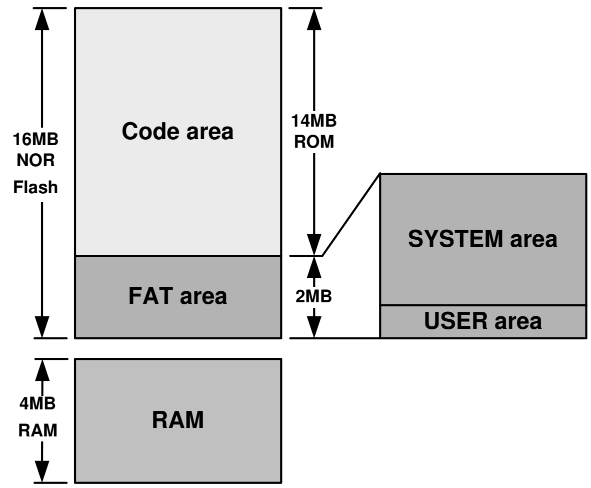

Figure 2. NOR flash memory architecture.

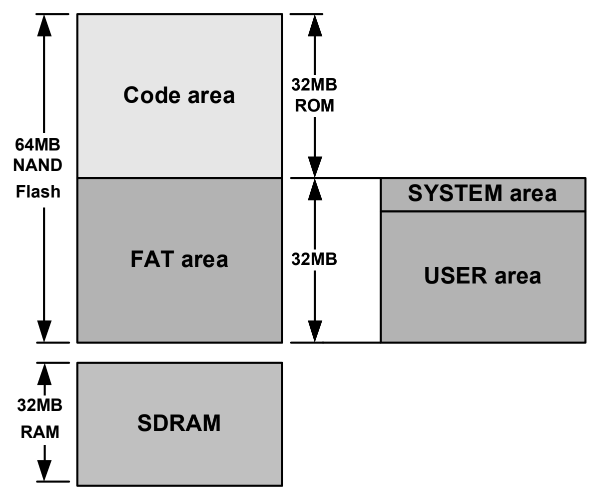

Figure 3. NAND flash memory architecture.

NAND flash memory is primarily used for general storage and data transfer. Code stored in NAND flash memory must be copied to RAM before it can be executed (Figure 3).

Shanzhai phones use both types of flash memory. NOR flash memory is usually deployed in low-end MTK chips (e.g., MT6225, MT6223 and MT6253) while NAND flash memory is usually deployed in high-end MTK chips (e.g., MT6235, MT6238, MT6228 and MT6230). Since NOR flash memory has higher a cost-performance ratio than its NAND

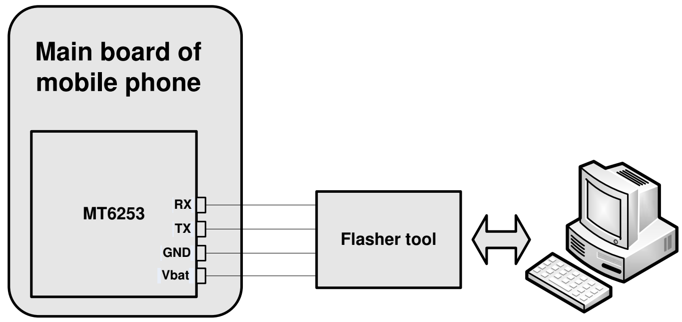

Figure 4. Connecting a flasher tool to a Shanzhai phone.

counterpart, it is used in a large number of Shanzhai phone models. Therefore, the rest of this paper focuses on NOR flash memory and the associated Shanzhai mobile phone models.

4. Experimental Results

Experiments were conducted on a low-end Shanzhai mobile phone model, a fake version of the Apple iPhone 4. Upon disassembling the device, we observed that the model was equipped with an MTK MT6253 chip and a 16 MB NOR flash memory chip (Toshiba TC58FYM7T8C). Our first task was to retrieve a binary image of the flash memory chip. Next, the flash memory dump was reverse engineered to extract forensically-relevant information.

4.1 Internal Memory Acquisition

A flasher tool was used to extract the memory contents of the MT6253 chip (Figure 4). Upon connecting the flasher tool to the MT6253 chip via the UART interface (RX and TX pins) on the chip, the Shanzhai phone was set to run in the factory programming mode. Next, a bootloader with a flash downloader program was transferred to the system RAM. This program dumped the flash memory byte-by-byte to the host computer.

4.2 Memory Dump Analysis

Before a hex editor such as HexWorkshop [1] could be used to analyze the binary image, it was necessary to understand the memory allocation scheme in the Shanzhai phone. The Toshiba TC58FYM7T8C chip integrates 16 MB NOR flash memory and 4 MB RAM. The 16 MB of NOR flash memory in the Shanzhai phone under test was divided into 14 MB

Figure 5. Directory of the user file area.

of code area and 2 MB of FAT area, the default configuration in MTK's turn-key solution.

The 2 MB of FAT area comprised a user file area and a system file area. The user file area is directly accessible by normal users via the mobile phone operating system; it stores photographs taken by the phone camera, downloaded files, etc. The user file area serves as removable storage media when the mobile phone is connected to a personal computer running Windows. As shown in Figure 5, this area is about 256 bytes. The remaining portion of the FAT area is used as non-volatile random-access memory (NVRAM) for storing system files such as the phonebook, SMS messages and call history.

The NVRAM is the most important portion of the flash memory image because it contains the majority of user data. Some of the data can be read or modified using the Shanzhai phone display and user interface, but the raw format of the data is unknown and the data is inaccessible to mobile phone users.

```
001D1400  50 48 4F 4E 45 30 30 30 31 FF FF FF FF FF FF FF  PHONE0001._.....
001D1410  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1420  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1430  FF FF FF FF FF FF FF FF FF FF FF FF FF FF 07 81  ................
001D1440  31 75 40 [redacted] F2 FF FF FF FF FF FF FF FF FF FF  1u@#............
```

Figure 6. Phonebook entry in the binary image.

Data in the NVRAM is organized as data items and stored as files. The Data Item Management System deployed to manage NVRAM data maintains an internal lookup table to read and write data items.

Our experiments focused on understanding how phonebook entries and call records are stored. We inserted one phonebook entry with the name "PHONE0001" and the number "135704****2" (four digits are hidden to protect the privacy of the user). We then attempted to extract the information from the binary image using HexWorkshop.

As shown in Figure 6, nine bytes of ASCII characters beginning at address `0x001D1400` are used to record the name of the phonebook entry (`0x50` is the ASCII code for "P" in PHONE0001). For simplicity, the hex address is the offset address shift from the absolute address `0x00E00000` (corresponding to 14 MB) in the flash image. One byte at address `0x001D143E` is used to indicate the length in bytes of the stored phone number. The next byte contains `81` corresponding to an international phone indicator. The phone number is stored at address `0x001D1440` (corresponding to `0x001D1445` in the BCD scheme). Thus, the length of one phonebook entry is 86 bytes.

```
000A6970  00 00 01 07 07 13 10 01 01 05 0A 02 81 31 75 40  _.............1u@
000A6980  [redacted] F2 00 F8 E9 01 08 00 00 00 00 00 00 00 00  #...............
000A6990  00 00 F2 F2 F2 F2 00 00 00 00 00 00 00 00 6C 08  ..............l.
000A69A0  00 00 00 00 00 01 01 00 00 00 00 00 00 00 00 00  ................
```

Figure 7. Call log entry (missed call).

Next, we searched the binary for the number "135704****2" in the call log. The BCD code corresponding to this number is `317504****F2`. Figure 7 shows a match for the number. The time of the call is "2010-01-01, 16:19:07, Friday," which corresponds to the hex values in the upper rectangle in the figure (`07 13 10` refer to the time "16:19:07" in reverse; note that the values are in hex, so that `13` corresponds to "19," `01 01` represents "January 1," `05` represents Friday and `0A` refers to the "2010"). The duration for the call is "00:00:00" corresponding to a missed call; its value is indicated by the hex value in the lower rectangle in the figure.

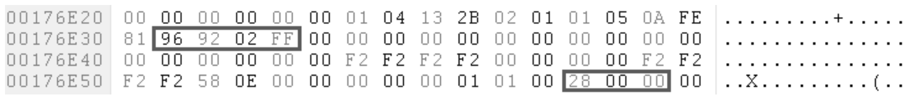

Figure 8. Call log entry (call duration of "00:00:40").

Figure 8 shows another example. In this case, the call log entry is for a call with a duration of 40 seconds.

The important observation is that data in the binary image matches the data obtained by manual acquisition via the user interface of the mobile phone. The mobile phone only records time information for the last call corresponding to a phone number. Time information for all calls cannot be retrieved via the user interface. However, this information can be located in the binary as long as the data is not yet overwritten.

4.3 Time Sequence Reconstruction

As discussed above, only the most recent call data is obtained via the mobile phone operating system. This is a problem in criminal investigations, especially when a comprehensive call history is required. However, the binary image often contains multiple copies of data that are created and stored at different points in time. This is an outcome of the flash memory erasure and allocation mechanisms. Since flash memory can only be erased one block at a time and arbitrary random-access rewrites and erases are not permitted, we observed that when a data item in NVRAM is updated, the mobile phone may modify the old data items in RAM and copy the entire block to a new address and mark the memory address of the old data item as invalid. This produces multiple copies of data items, which persist until the memory space is overwritten by other data. While the creation times of the copies are not stored, it is still possible to construct the time sequence corresponding to when the copies were created. Note that data that has been deleted by the mobile phone operating system can also be retrieved and used to construct the time sequence.

To understand the situation, we performed the operations listed in Table 2 sequentially on the Shanzhai phone. The operations modified the data items contained in the phonebook.

After all the operations were performed, a flash memory image was dumped for analysis. Using the hex editor, we found multiple copies of various entries. The results are presented in Table 3.

Five memory segments (short for *MEM_SEG_PHB*) are shown in Figures 9 through 13. With the help of the user interface of the mobile

Table 2. Number of copies of various phonebook entries.

Sequence	Operation	Contact	Phone Number
1	Add one entry	PHONE0001	135704****2
2	Add one entry	PHONE0002	13800138000
3	Add one entry	SZSMSC (in Chinese)	13800000755
4	Add one entry	PHONE0044444	110101
5	Delete one entry	SZSMSC (in Chinese)	13800000755

Table 3. Copies of various phonebook entries.

Contact	Phone Number	Copies
PHONE0001	135704****2	5
PHONE0002	13800138000	4
SZSMSC (in Chinese)	13800000755	3
PHONE44444	110101	2

phone, we discovered that the current phonebook contains three entries, where MEM_SEG_PHB-5 in Figure 13 shows the storage of the current data item. Since each modification operation performed on the data item generates one copy, we can deduce that MEM_SEG_PHB-4 in Figure 12 is the previous version of MEM_SEG_PHB-5 because the memory space for entry "SZSMSC" in MEM_SEG_PHB-5 is filled with `0xFF` values.

Upon comparing MEM_SEG_PHB-4 and MEM_SEG_PHB-3 (Figure 11), we see that the difference between the two segments is the entry "PHONE44444." Thus, it can be deduced that MEM_SEG_PHB-4 is the updated version of MEM_SEG_PHB-3.

MEM_SEG_PHB-1 shown in Figure 9 only contains one entry of "PHONE0001," which has the smallest number of entries but the largest number of copies. Thus, MEM_SEG_PHB-1 should correspond to the earliest version of phonebook.

MEM_SEG_PHB-2 in Figure 10 has a new entry of "PHONE0002" appended to "PHONE0001." Thus, MEM_SEG_PHB-2 is the newer version of MEM_SEG_PHB-1. This line of reasoning can be used with the data in the five memory segments to reconstruct the time sequence of user activities.

Thus, historical versions of phonebook data and call entries are retained until they are overwritten. However, even when some data items in memory are overwritten, it may still be possible to use the duplicated copies of phonebook data and call entries to establish the time sequence

```
00175C00  50 48 4F 4E 45 30 30 30 31 FF FF FF FF FF FF FF  PHONE0001.......
00175C10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00175C20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00175C30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF 07 81  ................
00175C40  31 75 40 [redacted] F2 FF FF FF FF FF FF FF FF FF FF  1u@#............
00175C50  FF FF FF FF FF FF 9B DE FF FF FF FF FF FF FF FF  ................
```

Figure 9. Memory segment of phonebook (1).

```
001D1E00  50 48 4F 4E 45 30 30 30 31 FF FF FF FF FF FF FF  PHONE0001.......
001D1E10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1E20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1E30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF 07 81  ................
001D1E40  31 75 40 [redacted] F2 FF FF FF FF FF FF FF FF FF FF  1u@#............
001D1E50  FF FF FF FF FF FF 9B DE 50 48 4F 4E 45 30 30 30  ........PHONE000
001D1E60  32 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  2...............
001D1E70  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1E80  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1E90  FF FF FF FF FF FF 07 81 31 08 10 83 00 F0 FF FF  ........1.......
001D1EA0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF 6C CF  ..............l.
```

Figure 10. Memory segment of phonebook (2).

```
001D1400  50 48 4F 4E 45 30 30 30 31 FF FF FF FF FF FF FF  PHONE0001.......
001D1410  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1420  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1430  FF FF FF FF FF FF FF FF FF FF FF FF FF FF 07 81  ................
001D1440  31 75 40 [redacted] F2 FF FF FF FF FF FF FF FF FF FF  1u@#............
001D1450  FF FF FF FF FF FF 9B DE 50 48 4F 4E 45 30 30 30  ........PHONE000
001D1460  32 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  2...............
001D1470  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1480  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D1490  FF FF FF FF FF FF 07 81 31 08 10 83 00 F0 FF FF  ........1.......
001D14A0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF 6C CF  ..............l.
001D14B0  80 6D F1 57 33 77 ED 4F E1 4E 2D 5F C3 FF FF FF  .m.W3w.O.N-_....
001D14C0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D14D0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
001D14E0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF 07 81  ................
001D14F0  31 08 00 00 57 F5 FF FF FF FF FF FF FF FF FF FF  1...W...........
001D1500  FF FF FF FF FF FF D1 94 FF FF FF FF FF FF FF FF  ................
```

Figure 11. Memory segment of phonebook (3).

Figure 12. Memory segment of phonebook (4).

Figure 13. Memory segment of phonebook (5).

of phone calls. Note that, because this feature is a direct result of the flash memory read/write mechanisms, the approach can also be used to establish the time sequence of other types of content such as normal data files.

5. Conclusions

The analysis of memory dumps from an MTK-based Shanzhai mobile phone reveals important details about how the system handles the addition and deletion of phonebook data and call entries. Although only the most recent phonebook items and call entries are obtained when querying the memory using the mobile phone operating system, valuable historical data pertaining to duplicated copies of phonebook data and call entries can be obtained from a memory dump as long as the associated memory locations have not been overwritten. This information can be used to establish the time sequence of phone calls as well as other content.

Our future research will focus on using the JTAG interface to obtain a complete image of internal memory, including the spare area. We will also attempt to apply reverse engineering techniques to obtain a detailed understanding of the allocation architecture for phone calls, phonebook entries, SMS messages and other related data. Finally, we will conduct research on Spreadtrum-based Shanzhai phones, another popular variety of Chinese-made pirated phones.

References

[1] BreakPoint Software, Hex Workshop Hex Editor (`www.hexworkshop.com`).

[2] M. Breeuwsma, M. de Jongh, C. Klaver, R. van der Knijff and M. Roeloffs, Forensic data recovery from flash memory, *Small Scale Digital Device Forensics Journal*, vol. 1(1), pp. 1–17, 2007.

[3] V. Gratzer and D. Naccache, Cryptography, law enforcement and mobile communications, *IEEE Security and Privacy*, vol. 4(6), pp. 67–70, 2006.

[4] C. Klaver, Windows Mobile advanced forensics, *Digital Investigation*, vol. 6(3/4), pp. 147–167, 2010.

[5] P. McCarthy, Forensic Analysis of Mobile Phones, Bachelor's Thesis, School of Computer and Information Science, University of South Australia, Mawson Lakes, Australia, 2005.

[6] MediaTek, Product lines, Hsinchu City, Taiwan (`www.mediatek.com/en/index.php`).

[7] P. Mokhonoana and M. Olivier, Acquisition of a Symbian smartphone's content with an on-phone forensic tool, *Proceedings of Southern African Telecommunication Networks and Applications Conference*, 2007.

[8] S. Punja and R. Mislan, Mobile device analysis, *Small Scale Digital Device Forensics Journal*, vol. 2(1), pp. 1–16, 2008.

[9] Spreadtrum Communications, Overview, Shanghai, China (`www.spreadtrum.com/en/products/products_overview`).

[10] T. Vidas, C. Zhang and N. Christin, Toward a general collection methodology for Android devices, *Digital Investigation*, vol. 8(S), pp. S14–S24, 2011.

[11] R. Wauters, Worldwide mobile phone market grew 20% in Q1 2011, fueled by smartphone boom, *techcrunch.com*, April 28, 2011.

[12] S. Willassen, Forensic analysis of mobile phone internal memory, in *Advances in Digital Forensics*, M. Pollitt and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 191–204, 2005.

[13] Z. Zhang, The research of MTK mobile phones flash file system recovery, *Netinfo Security*, issue 11, pp. 34–36, 2010.