3-12-2009

A forensics overview and analysis of USB flash memory devices

Krishnun Sansurooah
Edith Cowan University

DOI: 10.4225/75/57b28b7240cd3
7th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2009.

A forensics overview and analysis of USB flash memory devices

Krishnun Sansurooah
secau – Security Research Centre
School of Computer and Security Science
Edith Cowan University

Abstract

Current forensic tools for examination of embedded systems like mobile phones and PDAs mostly perform data extraction on a logical level and do not consider the type of storage media during data analysis. This report suggests different low level approaches for the forensic examination of flash memories and describes three low-level data acquisition methods for making full memory copies of flash memory devices. Results of a file system study in which USB memory sticks from 45 different make and models were used are presented. For different mobile phones, this paper shows how full memory copies of their flash memories can be made and which steps are needed to translate the extracted data into a format that can be understood by common forensic media analysis tools. Artefacts, caused by flash specific operations like block erasing and wear levelling, are discussed and directions are given for enhanced data recovery and analysis of data originating from flash memory.

Keywords Embedded systems flash memory, USB flash memory, flash translation layer (FTL), forensics acquisition and analysis.

INTRODUCTION

The era of portable digital data has seen an exponential expansion with the evolution in consumer electronics. The possible criminal use of mobile phones, personal digital assistant (PDAs), digital cameras, portable music and data storage devices has grown at an equally rapid rate. Most of these devices make use of memory cards which allow them to maintain portable data storage in a non-volatile way. These handheld devices have the capabilities to store either small or huge amounts of data. (Hu, 2004). This has all been made possible because of the availability of a non-volatile storage medium known as flash memory which has played a key role due to its size, low power consumption and resistance to shock (Douglis et al., 1994).

Flash memory is presently the most controlling non-volatile solid-state technology on the market and is accessible enough to be used for either legal or illegal purposes. From a forensics point of view, the tiny flash devices or drives may make the life of forensics experts very problematic when it becomes necessary to acquire and analyse their content. Current forensic tools for the examination of handheld devices, such as mobile phones and PDAs, do not always permit the successful acquisition and recovery of all the data that have been stored on the devices. Most of the time-deleted data, or other deleted data, which might be useful evidence about the offence perpetrated, cannot be acquired. The only way to be sure of acquiring all the data from a flash memory drive is to acquire the data at the lowest layer where evidence may be expected (Breeuwsma et al., 2007).

Flash memory is gaining popularity, as mentioned earlier, due to being shock resistant, being small enough for transportation of data to be hardly noticeable, its low power consumption, excellent response rates when it comes to random access time, the non-volatility of the medium and its low cost. In certain countries the number of flash memory drives is greater than the number of inhabitants (Breeuwsma et al., 2007).

Nowadays there are more and more systems using flash memory drives, either in conjunction with or as systems embedded into driver applications. Hence it is of the utmost importance that a sound way of data acquisition is developed to sustain and present evidence collected from flash memory drives in a court of law. Unfortunately, when it comes to flash memory storage systems, current forensics tools have great difficulties in acquiring essential data.

While writing to a flash memory storage system, the flash memory management scheme is actuated by the characteristics of the underlying flash media. The access patterns are released by the file systems and user applications (Huang et al., 2008). Despite including some of the evaluation tools for benchmarking storage devices, such as HD bench for hard drives and FD bench for flash drives, these tools do not take the flash memory characteristics into consideration, especially the flash memory management scheme which is more

commonly known as the flash translation layer (FTL) (Huang, 2008). According to Huang et al. (2008), the performance and reliability of any flash memory storage device is highly influenced by the following major factors:

1. the underlying flash media,
2. the management scheme design, and
3. the access patterns generated by the application.

Therefore, flash memory drives present a serious challenge for law enforcement, especially for forensics investigators who are hitting a brick wall when it comes to the acquisition and analysis of evidence gathered from the flash drives. Given that not much attention is being paid to these devices, the lack of understanding of how to acquire and analyse evidential data forensically, especially at the FTL level, is the primary motivation for researching this area.

FLASH TECHNOLOGY

Flash memory is a type of electrically erasable programmable read-only memory (EEPROM), meaning that the flash memory is non-volatile, i.e. it memorises its value without having to induce power, hence it is relatively dense (Gal & Toledo, 2005). Flash drives are commonly used to store files and other objects on different handheld devices such as mobile phones, PDAs, portable music, USB drives, digital cameras, to name just a few. However, flash memory write/read/erase behaviours are very different from other normal memories, such as random access memory (RAM) and magnetic disks. With flash devices, the memory cells can only be written to a limited number of times, typically between 10,000 and 1,000,000 times, whereafter they become unstable as they wear out (Gal & Toledo, 2005).

Flash (EEPROM) is normally available in two types:

1. NOR flash, which allows and supports a fast random access speed, but at a very high cost.
2. NAND flash, which is newer and cheaper, with the advantage that it carries a larger storage capacity and achieves decent, if not high, execution for large read/write operations (Lim & Park, 2006).

These two different flash memory types have a common factor: each bit in a new flash chip will be appointed a logical one where only a WRITE operation can alter its value from a 1 to a 0. However, the only method of performing this change is to go through an ERASE operation (Woodhouse, 2001). NAND flash memory chips are compartmentalized into blocks. Each block has a pre-defined number of pages which are fixed and which, in turn, are scaled down into regions for storing data. There is also a free space region which is responsible for holding the status of the data region.

Woodhouse (2001) stated that the first generation of NAND flash memory had a typical page size of 512 bytes, each carrying a surplus of 16 bytes of "***out of band***" storage space which was designed to be used for metadata and error correction. Normally NAND flash is written by injecting the necessary data into an internal buffer one byte at a time, requesting a WRITE command.

A NOR flash memory device operates differently by allowing bits to be wiped out individually until every bit is cleared. In NAND flash, only a few WRITES cycles are written to each page before the page's content becomes undefined and has to wait for the next ERASE pass by the blocks where the page is located. In other words, each time data is altered, the new data must be written to a different and available page in a different location (Lim & Park, 2006).

Therefore the old page where the data was written initially is considered to be a dead page. When a period of time has passed, the amount of dead pages accumulated is reclaimed by the system, which performs an ERASE operation to make the dead pages available again.

This process is known as "***garbage collection***" (Woodhouse, 2001) and reclaims the invalid pages. However, the flash memory block has a limited number of allowable ERASE cycles; therefore a strategy must be put in place to ensure that all the erased blocks are performed evenly to achieve a longer life span of the flash memory device. This is also known as "wear-levelling" (M-Systems, 1998).

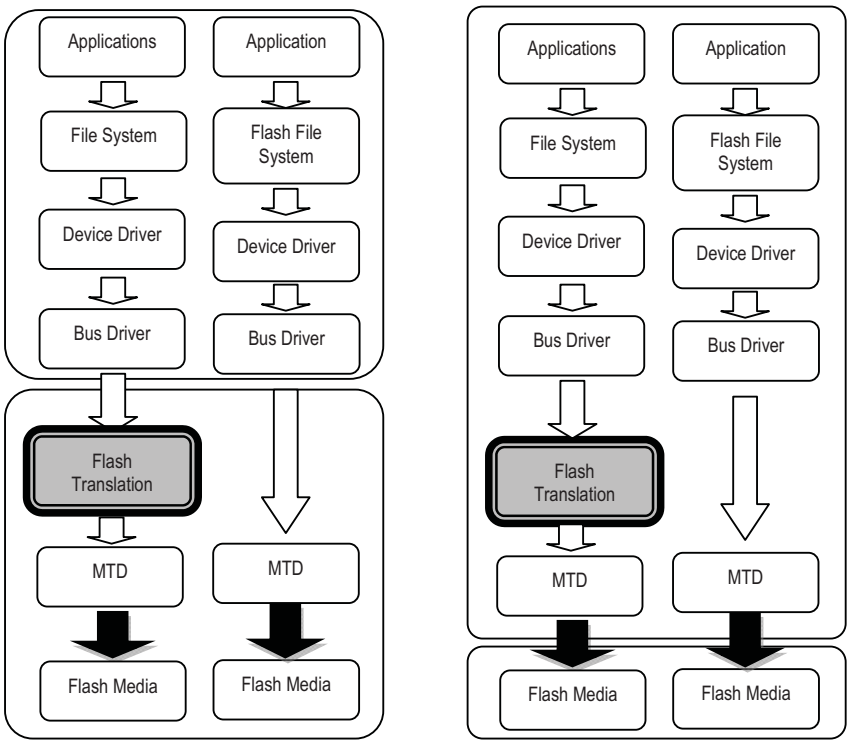

Figure 1 The MTD storage system architecture.

FLASH TRANSLATION LAYER (FTL)

Flash memory has gained a lot of popularity during the past decade, because of its storage capabilities which have already reached gigabytes of data, its fast speed to access data, its non-volatile memory storage, small size, shock resistance, low powered devices and, finally, because it is inexpensive (Intel Corporation, 1998). The FTL driver has been introduced to work between existing file systems, including existing operating systems, or even embedded applications. Flash memory is designed to make linear flash memory like writing onto a disk. According to Huang et al. (2008), the flash translation layer protocol (FTL) and the NAND flash translation layer (NFTL) are very popular. The FTL driver therefore mimics the flash media as block devices so that both the user and the systems may access the flash media transparently. It was therefore concluded that there could be two types of flash memory chips, as shown in figure 1 above.

The first type includes both the MTD and the FTL driver in one package, such as the USB flash drive. The second type does not include the MTD, as illustrated in figure 1. Therefore this block level layer is responsible for redirecting the location of updated data from one page to another. It is also responsible for the management of the actual physical location of the data which is located into a mapping table. This mapping of logical to physical location can only be achieved at the page level (FTL) or at the block level (NFTL) (Ban, 1995). The main differences between these two mapping techniques are the table sizes and the redirecting constraints. Nonetheless, these methods may be used directly on the flash translation layer (Lim & Park, 2006).

Lim and Park (2006) also mentioned that making direct use of an existing file system may impact on performance, due to the fact that file systems are designed and developed for disk-storage systems. Therefore the way of accessing the files, the file sizes and the file metadata on disk storage is not the same as on a flash memory device.

JOURNAL FLASH FILE SYTEM (JFFS)

A journaling flash file system is a log-structured file system where nodes with their content, such as data and metadata, are stored on flash chips in a sequential order advancing further into other free slots or spaces in a linear pattern through the storage space. In JFFS there is only one node type which is known as "*struct jffs-raw-inode*" and which has a single association with the inode. According to Woodhouse (2001), the different constituents of the inode area common header retain information about the current metadata file system of that particular inode and the data. The log contains fixed-sized sections of the disk which are attached, together forming a pointer list. Both metadata and data are placed at the back of the log, thus never overwriting the old

data recorded on the storage space. Therefore the modified data need to be written somewhere else (Gal & Toledo, 2005). Originally developed by Axis Communication AB for embedded Linux. (Axis Communication, 2004), it was remodelled later to create journaling flash file system 2 (JFFS2). JFFS2 was more flexible and thus permitted new type of nodes to be defined whilst retaining the ability to work backwards.

Kawaguchi, Nishioka and Motoda (1995) pointed out that log-structured file systems were appropriate for flash memory management, especially when it came to designing a block-mapping device.

Every JFFS2 node has a common header which contains the node's length, the node's cyclic redundancy checksum and its type. Yet these are not the only data that the common header retains; it uniquely identifies the node's structure and the node's type field which hold a bitmask allowing either an unsupported or a supported format of data to be read. This was not the only obstacle that flash memory drives were facing. Systems will scan every single node and will therefore create two different structures. The first structure will be a list of every inode, and their respective versions, and the second will include all structures that are equivalent to a valid node on the flash. As mentioned previously, those two data structures are then linked, with one containing all the physical addresses to assist in garbage collection and the other a sequential order of all the nodes (Gal & Toledo, 2005).

JFFS2 also uses a simple wear-levelling technique which helps to extend the life span of flash memory drives, in this example the USB flash. Obviously, with flash memory drives, data may be written to an address several times, typically between 10,000 and 100,000 (Corsair, 2007). If you write to the same location over and over, it is more likely that the flash chip will wear out at that address. Therefore wear-levelling is used to make sure that data is distributed evenly across each memory block of the whole USB flash memory.

YET ANOTHER FLASH FILLING SYSTEM (YAFFS)

Written and developed by Aleph One as a NAND file system, YAFFS has a more efficient approach than JFFS and JFFS2 (Aleph One, 2002). The way in which YAFFS addresses the pages in the flash memory drives is totally different. All files are saved in fixed-size chunks of either 512 bytes, one kilobyte (1KB) or two kilobytes (2KB) (Gal & Toledo, 2005). Each page is assigned a file ID and a chunk number. The inode number is associated with the file ID. The file ID, also known as the header, will normally be 16 bytes for 512 bytes, 30 bytes for one kilobyte (1KB) and 42 for bytes for two kilobytes (2KB). Having the same characteristics as JFFS2, the mapping information on the flash is the only content of each and every chunk which lies as part of the header. Consequently, at the time of mounting the flash drive, all the headers must be read from the flash to generate the file ID. File ID are generally stored in RAM at all times as for the JFFS principle. Therefore, to save RAM, a more effective map structure to map file locations to the corresponding physical addresses was required and subsequently addressed by YAFFS. This mapping plan follows a tree-structure where the internal nodes hold 8 pointers to the other nodes and leaf nodes hold 16 pointers to the physical location. With the YAFFS, which is slightly more complex than the previous version, the primary aim was to get YAFFS2 to write in a sequential pattern within the ERASE units so that all the pages could be erased one after the other. (Gal & Toledo, 2005).

Lim and Park (2006) agreed that the flash memory drive normally writes the raw or untransformed data one byte at a time. Each time that data is altered, the new data must be re-written into a different and available page in a different location. The FTL is a technique used to hold some of the direct map embedded within the flash drive, whilst reducing the action of updating the map on the flash drive (Ban, 1995). Using the FTL in turn uses a virtual block map (VBM) of 32 bits to represent each entry point to a logical address of the flash media where the virtual data block resides (Intel Corporation, 1998). Since the untransformed data that has been injected into the medium still exists, any altered or new data is therefore written into another available page at a different location. Only the FTL knows the locations through the mapping table. Since the original data still resides on the flash memory chip and because the FTL and garbage collection happens after a number of writes, recovering the virtual-to-physical list becomes very challenging. This paper intends to determine how to acquire the raw evidential data residing on the medium forensically without tampering with the untransformed data.
The following figures will provide a better understanding of how the flash memory drives are arranged and how data is stored on them.

Figure 3(i) depicts an illustration of the data that are stored on a flash medium after the initial injection of data.

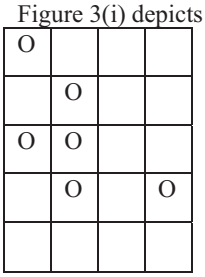

Figure 3(i) illustrates the initial injection of data onto the flash device

When the first data are stored for the first time on the flash memory drive, they are put into an available free slot, according to their sizes, as illustrated in figure 3(i) by the letter 'O'.

When the data has been changed or modified, or new data need to be saved onto the flash memory drive, the FTL determines which is the best way of storing the new data on the medium. Figure 3(ii) shows the pattern of the modified data, denoted by the letter 'X', that will be injected on the medium.

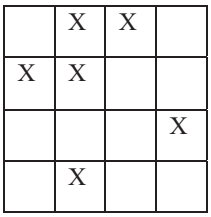

Figure 3(ii) represents the data that needs to be added to the flash medium.

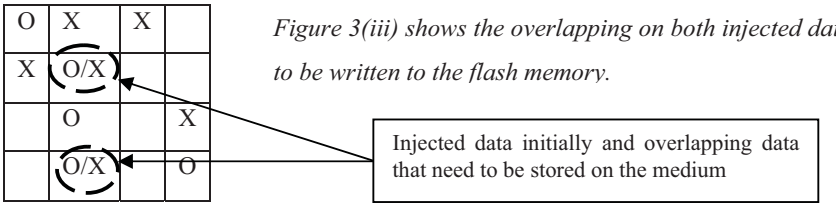

Figure 3(iii) shows the overlapping on both injected data and the altered data that need to be written to the flash memory.

There is an overlap of the initial data already stored on the medium and the newly modified data or newly added data needing to be stored in the memory, thereby causing some difficulties in managing the storage space. The FTL is responsible for re-organizing the arrangement in which the data needs to be stored on the flash memory drive.

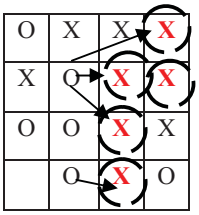

Figure 3(iv) shows when both the initially injected data and the added data have been recorded onto the medium which is indicated by the circled red 'X' and which can be in any temporary space on the flash medium.

According to Lim and Park (2006), each time that data is modified and needs to be written to the flash memory drive, the date for the new data, which may be additions to an existing document or the retouched sections of an image, must be written into a different free page in a separate location within the flash chip in order to denote that the page is a live one. Depicted in figure 3(iv), and marked with a red 'X' and circle, both the initial and modified data is represented on the same chips with some data having to shift to a different location managed by the FTL. Since there is a conflict between the initial data and the new or modified data that need to be stored on the flash memory drive, the only way that this may be resolved is that the FTL is forced to ERASE the initial data already in place on the flash chip and then performs a READ/ERASE/WRITE the modified or new data onto the same location. This forces the flash drive to perform an action that will cause wear-levelling to the memory chip. After a period of time, the memory chip will degrade and the flash drive will become unstable, wearing out sooner than its expected life span.

SIGNIFICANCE OF STUDY

Due to ongoing change and evolution in digital technology, there has been an exponential growth in the number of flash memory drives. The data stored on flash drives reflects a spectrum of human behaviour and may become subject to a forensics investigation. However, forensics tools for acquisition and analysis of flash memory drives are of relatively low quality. There is currently no established framework or methodology to support law enforcement officers who may need to carry out forensic analysis of these devices (Breeuwsma et al., 2007). This contrasts with the basic traditional computer forensics methodologies and standards that are already supported by various government agencies (Ayers, Jansen, Cilleros & Daniellou, 2005).

Brinson et al. (2006) claimed that the tiny and adaptable nature of these devices make forensics investigators' tasks even more complex. Organized crime is using flash memory technologies to perpetrate its illegal activities. Due to their high portability and small sizes, anyone may use USB flash drives to carry valuable information or secrets that have been stolen from a business. Criminals arrested under terrorism charges may reveal useful information about planned bombings or other information that help prevent further catastrophes. People suspected of abusing children and transporting child pornography may be found to have used USB flash drives to share images on a particular network. Flash drives may be attached to key rings or be used as USB wireless dongles for wireless mouses without being detected. A study conducted in the Unites States of America has demonstrated that the highest percentage of Internet users addicted to pornography is aged between 12 and 17 years (Rockwell, 2005).

Methodologies have been designed to acquire data from computer systems and analyse evidence forensically. A forensics investigator will follow a set of standards and procedures before any conclusions are drawn. A complete analysis of the whole system may be necessary: log files may permit the examiner to draw conclusions about the particular source of the attack on the system and why it occurred (Forté, 2005; Jones, Brejtlich & Rose, 2006). However, very little attention has been paid to the forensic analysis of flash memory drives. (Boyd & Foster,2004; Jones & Meyer, 2004; Marcella& Greenfield, 2002).
Flash memory drives do not generally hold connection logs but fortunately they do keep a partial record of their entire virtual block mapping, i.e. virtual-to-logical, making use of virtual block map (VBM) (Intel, 1998). With more and more flash memory drives flooding the market and being accessible to anyone, it is possible that everyone owns at least one. The potential for flash drives to become a major source of digital evidence is illustrated in figure 5 below.

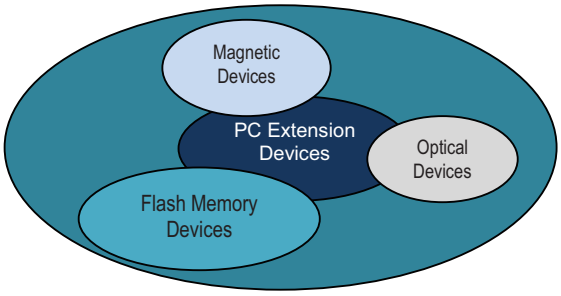

Figure 5 illustrates the interaction of the small scale digital devices in contrast to the level of interaction with the digital evidence world which is quite significant.

The major focus of this research will be the development of a methodology to acquire, analyse and classify untransformed evidential data. This might require the investigator to develop unique methods or steps to associate, verify, tag, secure and preserve useful information from a presumed criminal use of flash memory drives which have the potential of retaining gigabytes of data which may be revealed in the prosecution of criminals.

DIGITAL FORENSICS

Carr, Gunsch and Reith (2002) disagreed that lawbreakers believe that there is a level of obscurity associated with perpetrating electronic crimes. Nolan et al. (2005) agreed that criminals nowadays are aware that evidential data may reside on an electronic device for a long time after the crime has been committed, allowing forensic

investigators to retrieve information later mainly with "persistent data" which would remain on the medium even when it is powered down (Nolan et al., 2005). Despite computer forensics no longer being the only area of interest for forensics investigators, criminals are breaching laws using other small scale digital devices, including cellular phones, digital cameras and PDAs. Some research has already been undertaken and methodologies developed for collecting evidence from small devices with volatile memory (Jansen & Ayers, 2004). There are no detailed guidelines, methodologies, models, frameworks or best practices available to the investigator who wishes to acquire and analyse non-volatile evidential data on USB flash memory drives.

Brown (2006) defined digital forensics as being an in-depth inspection of computer networks and digital devices to collect evidential information in such a way that it would be presentable for admission in a court of law. The Association of Chief Police Officers' (ACPO) *Good Practice Guide for Computer Based Electronic Evidence* (2003) is a set of guidelines commonly followed when electronic data are to be acquired. Among the four rules that the ACPO provides, the *Guide* agrees that not all electronic evidence would fall under the *Guide's* scope. An example of evidence falling outside the *Guide's* scope would be the forensics acquisition and analysis of a USB flash memory drive, as its architecture differs from traditional storage media. Yet the process of evidence acquisition for forensics purposes has to follow a set of guidelines in the collection, preservation and presentation of the elements in a court of law. The gathered evidence may involve threatening letters, child porn photos or videos, illegal pornographic photographs or materials, network log files, details of planned terrorist attacks, information about other terrorist cells, fraud or identity theft. Data may be retrieved from seized equipment (Jones & Meyler, 2004).

Computer storage media such as hard disks and volatile and non-volatile memory drives may be forensically searched with various pre-tested frameworks or methodologies. Despite outlining the different forensic frameworks available for use, Carr, Gunsch and Reith (2002) demonstrated that these models and protocols are not normalized. Both the United States Department of Justice and the US Secret Service (USSS) remodel and recalibrate existing guidelines to suit their requirements and sometimes need to develop their own methodologies to address a particular issue. This depends on the different scenarios that they are assigned, i.e. depending on the device incriminated, the operating systems or embedded applications and the means available to the forensics investigators.

PDAs and mobile phones have embedded software and carry operating systems which constantly keep changing the content stored (Jansen & Ayers,2004). On small devices, this may occur without user interaction, thus being in contradiction to the ACPO Rule 1 discussed previously. Casey (2004) clarified the benefit of using a toll Win HEX to acquire a memory dump which can allow unencrypted passwords to be retrieved. This mechanism to acquire data from sources such as networks might be used in the same way to recover raw evidential data from flash memory drives.

The second rule employed by the ACPO (2003) declared that the original untransformed data should only be accessed under extraordinary circumstances. This does not take into consideration that, with flash memory drives, data is never written to the same location twice. With the garbage collection process happening in the flash memory drives, this ACPO rule will not be applicable. The examiner will be left with no flexibility.
The Best Practices for Seizing Electronic Evidence issued by the USSS referred to flash memory drives under "other electronic storage devices" heading (USSS, 2006), whereas the National Institute of Justice (NIJ), which is a subset of the Unites States Department of Justice, listed the flash memory drive more precisely under "thumb drive" in their *Electronic Crime Scene Investigation: A Guide for First Responders* publication (NIJ, 2008).

According to a study carried out by Carr, Gunsch and Reith (2002) to analyse the methods and techniques which cover the field of computer and digital forensics, the terminology "digital forensics" should be tailored to encompass both current and future digital technologies. Digital forensic procedures are not addressing the essentials when it comes to small scale digital devices such as USB flash memory drives, meaning that evidential information can be retraced and recovered to be analysed from fixed digital storage and from non-volatile storage.
The very first principle when analysing any piece of electronic evidential data is to ensure that the data held on the medium is kept unchanged, hence establishing procedures to preserve, identify and extract useful information. The various digital forensics models available (Carrier & Spafford, 2003; Department of Justice, 2001; O'Ciardhuain, 2004) support a set of defined processes and procedures for acquiring, preserving, analysing, and finally presenting the data recovered from the digital devices. O'Ciardhuain (2004), however, suggested a model which emphasizes the generic procedures of digital evidence collection during an investigation, but did not consider flash drives.

Breeuwsma et al. (2007) revealed that, when it is necessary to deal with non-volatile flash memory, the easiest non-invasive way to read flash data is by using a flasher that make a copy of all flash memory data from the source system to another separate system for further analysis. (Breeuwsma et al., 2007). The report also mentioned that there is no standardized way of performing such operations. However, using such tools can create havoc as these tolls are mainly developed by manufacturers or service centres for testing and debugging functionalities or simply for checking and modifying the intended purpose of the device. Yet, forensics investigators have to be very prudent while using these tools as they have other options that might cause the whole device to lose all its content and beyond forensics recovery. Breeuwsma et al. (2007) refer to another method of accessing the flash memory drives through the Joint Test Action Group (JTAG), also known as boundary-scan, when the flasher is not an option. Van der Kniff (2002) pointed out that using a universal memory chip reader /programmer and de-soldering the chip from the confiscated device could be a very risky option. Modern systems on the market have a special port called the JTAG test access port. As in most embedded systems, as represented in Figure 7, the flash memory is mounted or connected to the other chips like a processor which can be used to access the flash memory of the embedded device as the JTAG test access port is meant to be used for testing and debugging (Breeuwsma et al., 2007). The JTAG option is very safe as it will produce a forensics image of the content of the flash memory drive and dump it onto a different medium for later analysis.

Another feasible way of producing a forensics dump of the flash memory drive is to de-solder the flash chip from the printed circuit board (PCB) and read the memory through a flash reader or programmer. As most chips nowadays are packed in a thin small outline package (TSOP) or on a micro ball grid array (BGA), the chips could be physically extracted from the embedded system and hence imaged for further analysis or examination. Again there is no proper way of handling, or a sound forensic methodology, to ensure this operation of extraction and imaging of the flash memory chip is performed successfully so that it might be admissible as an exhibit in a court of law.

CONCLUSION

Digital forensics has been evolving during the past decades. It is no longer focusing on computers only but instead includes many small scale digital devices. Moreover, there is very little evidence of research being undertaken in the area of flash memory drives where this research will be beneficial to the law enforcement field. More research needs to be done on the flash read mechanisms used by flasher tools in order to adapt these mechanisms for usage in the next generation of forensic data acquisitions tools. Steps have been illustrated for translating acquired flash data to a level that can be understood by existing forensic tools targeted towards commonly used file systems.

FURTHER RESEARCH

More research is needed for flash data that cannot be directly translated to file system level. More research is also needed on the relation between flash specific operations like block erasing and wears levelling on one side and the resulting artefacts and potentials for data recovery and analysis on the other side. With the results of this research, future forensic tools might be able to improve the power and efficiency of embedded systems examinations for reasonably skilled IT professionals. Further research will attempt to extract the flash chips and the FTL micro-controller chips to have a better in-depth analysis of how evidential data could be recovered from the different USB flash devices.

REFERENCES

ACPO. (2003). Good Practice Guide for Computer based Electronic Evidence. Retrieved May, 17, 2008, from http://www.acpo.police.uk/asp/policies/Data/gpg_computer_based_evidence_v3.pdf

Aleph One Ltd, (2002). Yaffs: A NAND-Flash Filesystem. Retrieved May 9, 2008 from http://www.aleph1.co.uk/yaffs/, 2002.

Ayers, R., Jansen, W., Cilleros, N., & Daniellou, R. (2005) Cell Phone Forensics Tools: An Overview and Analysis. Retrieved May 29, 2008, from http://csrc.nist.gov/publications/nistir/nistir-7250.pdf

Ban, A. (1995). Flash File System. US Patent, no. 5,404,485,

Boyd, C., & Forester, P. (2004). Time and a date issues in forensics computing - a case study. *Digital Investigations, 1*(1), 18-23.

Breeuwsma, M.F. (2006). Forensic imaging of embedded systems using JTAG (boundary-scan). Retrieved April 24, 2008 from http://0-www.sciencedirect.com.library.ecu.edu.au/science?_ob=MImg&_imagekey=B7CW4-4JG5FJG-1-4&_cdi=18096&_user=1385697&_orig=search&_coverDate=03%2F31%2F2006&_sk=999969998&view=c&wchp=dGLzVtz-zSkzk&md5=0209b5e13b2504060991e3837387de08&ie=/sdarticle.pdf

Breeuwsma, M.F. et al. (2007) Forensic Data Recovery from Flash Memory. Retrieved April 23, 2008 from http://www.ssddfj.org/papers/SSDDFJ_V1_1_Breeuwsma_et_al.pdf

Brown , C. (2006) Computer Evidence Collection and Preservation. Hingham, MA: Charles River Media.

Brinson et al. (2006). A Cyber forensics ontology: Creating a new approach to studying cyber forensics. Retrieved September 21, 2008, from http://www.dfrws.org/2006/proceedings/5-Brinson.pdf

Carr, C., Gunsch, G., & Reith, M. (2002). An Examination of Digital Forensic Models. *International Journal of Digital Evidence 1*(3), 1-12.

Carrier, B., & Spafford, E.H (2003). Getting Physical with the Digital Investigation Process. *International Journal of Digital Evidence, 2*(2), 1-20.

Carrier, B. & Grand, J. (2004). A hardware-based memory acquisition procedure for digital investigations. *Digital Investigation,1*(1), 50-60.

Corsair. (2007). USB Flash Wear-Leveling and Life Span. Retrieved June 2, 2008, from http://www.corsairmemory.com/_faq/FAQ_flash_drive_wear_leveling.pdf

Douglis, et al. (1994). Storage Alternatives for Mobile Computers. In Proceedings of the First USENIX Symposium on Operating Systems Design and Implementation (OSDI), Monterey, California. 25–37.

Forte, D. (2005) Log management for effective incident response. *Network Security, 2005*(9), 4-7.

Gal, E. & Toledo, S. (2005). Algorithms and Data Structures for Flash Memories. Retrieved April 11, 2008, from http://www.tau.ac.il/~stoledo/Pubs/flash-survey.pdf

Gal, E. & Toledo, S. (2005). Mapping Structures for Flash Memories: Techniques and Open Problems. Retrieved April 15, 2008, from http://www.tau.ac.il/~stoledo/Pubs/swste2005.pdf

Hu, C. (2004). A Preliminary Examination of Tool Markings on Flash Memory Cards. Retrieved March 3, 2008, from http://scissec.scis.ecu.edu.au/publications/forensics04/Hu.pdf

Huang, P. et al. (2008). The Behavior Analysis of Flash-Memory Storage. Retrieved April 7, 2008 from http://newslab.csie.ntu.edu.tw/~johnson/public_files/2008%20ISORC%20-%20The%20Behavior%20Analysis%20of%20Flash-Memory%20Storage%20Systems.pdf

Intel Corp. (1998). Understanding the Flash Translation Layer (FTL) Specification. Retrieved May 8, 2008, from http://developer.intel.com

Intel Corp. (1998). Understanding the Flash Translation Layer (FTL) Specification. Retrieved May 8, 2008, from http://www.embeddedfreebsd.org/Documents/Intel-FTL.pdf

Intel Corporation. (1998). Flash file system selection guide. Application Note 686, Intel Corporation.

Jansen, W. (2005). *Mobile Device Forensic Software Tools.* Paper presented at the Techno Forensics 2005, Gaithersburg, MD, USA.

Jansen, W., & Ayers, R. (2004). Guidelines on PDA Forensics. Retrieved March 1, 2008, from http://csrc.nist.gov/publications/nistpubs/800-72/sp800-72.pdf

Jones, K. J., Bejtlich, R., & Rose, C. W. (2005). *Real Digital Forensics : Computer Security and Incident Response*. Upper Saddle River, NJ, USA: Addison-Wesley Professional.

Jones, A., & Meyler, C. 92004). What evidence is left after disk cleaners? *Digital Investigations, 1*(3), 183-188.

Jones, K. J., Bejtlich, R., & Rose, C. W. (2006). *Real Digital Forensics: Computer Security and Incident Response.* Upper Saddle River, NJ: Addison Wesley.

Kawaguchi, A., Nishioka, S., and Motoda, H. (1995). A flash-memory based file system. In Proceedings of the USENIX 1995 Technical Conference, New Orleans, Louisiana. 155–164.

Lim, S. & Park, K. (2006). An Efficient NAND Flash File System for Flash Memory Storage. May 15, 2008 http://ieeexplore.ieee.org/iel5/12/34313/01637405.pdf?tp=&isnumber=&arnumber=1637405

Lin, et al. (2007). A NOR Emulation Strategy over NAND Flash Memory. Retrieved September 13, 2008, from http://ieeexplore.ieee.org/iel5/4296820/4296821/04296841.pdf

Marcella, A. J., & Greenfield, R. S (2002). *Cyber Forensics: A Field Manual for Collecting, Examining and Preserving Evidence of Computer Crime.* Boca Raton, FL: Auerbach.

NIJ. (2008). Electronic Crime Scene Investigation: A Guide for First Responders. Retrieved September 22, 2008, from http://www.ncjrs.gov/pdffiles1/nij/187736.pdf

Nolan, R., O'Sullivan, C., & Waits, C. (2005). First Responders Guide to Computer Forensics. Retrieved June 9, 2008, from http://www.cert.org/archive/pdf/FRGCF v 1.3.pdf

Rockwell, M. (2005). Study: Wireless Offering New Temptations. Retrieved July 31, 2008, from http://www.wirelessweek.com/article/CA630117.html

Samsung Electronics. (1999). APPLICATION NOTE for NAND Flash Memory. Retrieved July 17, 2008, from http://www.samsung.com/Products/Semiconductor/Memory/appnote/app nand.pdf

Samsung Electronics Co. (2002). NAND Flash Memory & SmartMedia Data Book. Retrieved March 15, 2008 from http://www.samsung.com/

USSS. (2006). Best Practices for Seizing Electronic Evidence. Retrieved April 22, 2008, from http://www.ustreas.gov/usss/electronic_evidence.shtml

Van der Kniff, R. M. (2002). Embedded Systems Analysis. Chapter 11 of Handbook of Computer Crime Investigations - Forensic Tools and Technology. Academic press.

Woodhouse, D. (2001). JFFS: The Journaling Flash File System. Retrieved March 21, 2008, from http://sources.redhat.com/jffs2/jffs2.pdf

Woodhouse, D. (2004). Memory Technology Device (MTD) Subsystem for Linux. Retrieved June 18, 2008, from http://www.linux-mtd.infradead.org

COPYRIGHT