POLITECNICO DI MILANO

Master of Science in Computing System Engineering

Dipartimento di Elettronica e Informazione

SOLID STATE DRIVE (SSD)

DIGITAL FORENSICS CONSTRUCTION

Supervisor: Prof. Stefano Zanero

T H E S I S

Fkrezgy Yohannes Matr. N. 750710

Academic Year 2010/2011

ASTRATTO

La tesi descrive una indagine di forense su una nuova tecnologia che recentemente sta diventando molto usata basata sulla memoria flash NAND per l'archiviazione dei dati chiamata Solid State Drive (SSD). Si parte con la spiegazione della teoria esistente nella forense digitale (evidenza digitali, la valutazione della loro affidabilità insieme con la sfida della forensic digitale) e anche un analisi dettagliata della memoria basata su NAND-flash che aiuta nel recupero dei dati (breve spiegazione delle cellule NAND-flash, l'architettura di SSD e le tecniche di gestione flash che si incontrano nel trattamento dei NAND-flash). L'obiettivo principale di questa tesi è l'indagine ed il recupero dei dati dalle unità solid state mediante l'applicazione del metodo di forensi ed altre techniche che sono stati usati finora nella tecnologia di memoria flash. I metodi che vengono utilizzati nel processo di forense per SSD non sono molto diverse dagli altri supporti digitali tranne l'eccezione nella tecnica applicata per il recupero dati dovuta dal fatto dell'unicità del SSD. Principalmente tre metodi di forensi sono stati tratati in questo elaborato: L'acquisizione (fare una copia digitale delle evidenza originale), autentificazione della copia evidenza (con funzione di hash) e alla fine l'analisi delle copie digitali. In questa tesi si stato cercato di far vedere una visione generale di analisi di forense per il recupero dei dati dalle unità solid state e si sono determinati che alcune prestazioni di SSD (come il comandoTRIM) possono avere un efetto negativo come quello di eliminazione dei dati.

ABSTRACT

This thesis describes a forensic investigation of an emerging technology NAND-based flash memory data storage called Solid State Drive (SSD). It starts with explaining the existing theory in computer forensics (digital evidence and their reliability along with the challenges in computer forensics) and a detailed background of NAND based flash memory, which helps in conducting data recovery (brief explanation of NAND-flash cell, structure of SSD, architecture of SSD and flash management techniques to handle NAND flash challenges). The main focus of this thesis is to investigate and successfully recover data from solid state drives by applying forensic methods and techniques which have been used so far in the flash memory technology. These methods used in forensic process of the SSD are not quite different from other digital devices except some of the techniques applied for data recovery due to the uniqueness of the SSD. Mainly three forensic methods are covered in this paper: Acquisition (making a digital copy of the original evidence), authenticating the copy of the evidence (using hash function) and finally analyzing the digital copy. In this thesis has been attempted to show a general overview of forensic data recovery analysis of solid state drives and determining if some of SSD performance improving features (like TRIM command) can purge data.

ACKNOWLEDGMENTS

First of all I would like to thank my supervisor Prof. Stefano Zanero for his support, encouragement, guidance and prolific suggestion from the initial to the final level enabled me to complete the thesis. I am also grateful to another research staff Luigi Sportiello for his contribution, constructive guidance and offering me helpful materials.

Above all, personally I wish to thank my parents, my uncle as well as the rest of my family for their advice and encouragement throughout my career. Mom and Daddy, you have been my teacher and role model, and thanks for teaching me everything in my life. I also owe my deepest gratitude to my uncle Gebru Abreha, who is one of the most thoughtful, gracious, kind and generous to me.

TABLE OF CONTENT

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Computer Forensics

Year by year, the number of computers and other digital devices being used is increasing. Today, computers are widely used all over the world and acts as people's right hand and close friend in almost all fields. It is hard and even terrible to imagine how people's life would be if they did not have computers. It's very clear that computers have an immense benefit. On the other hand, computers and other digital devices can also be used for illegal actions by providing avenues for misuse and opportunities for committing crimes. Few examples of criminal activities committed with the help of computers are: fraud, theft, extortion and vandalism. Computers can help criminals to commit crimes in two different ways. The first category of crimes can only be committed by using a computer system. These crimes never existed before the advent of the computer, and a computer is absolutely essential for committing such a crime. The second category of computer crime is much wider, and involves crimes that have existed for centuries, but are now committed by using a computer system.

The increase in computer related crime has created a new branch of forensic science known as computer forensics, which deals with reconstruction of electronic evidence (e-evidence) from digital devices in a manner that is legally acceptable by the court. Computer forensics can be defined as the collection, preserving, analysis and court presentation of computer-related evidence [25]. The forensics process can often involve the creation of bit stream copies of digital storage to both ensure the integrity of the data and to capture data which would otherwise be lost in a logical copy.

1.2 Digital Evidence

The introduction and penetration of computers and other electronics in the society daily life increasingly influenced laws and jurisdiction in the last few years. These days' traditional evidences are no longer the only evidences used in court. Digital evidence like files, photos, videos and others can also provide huge evidence against criminals. Digital evidence comes from a variety of sources including computing devices (e.g., desktop and laptop computers, digital cameras, music players, personal digital assistants [PDAs], and cellular telephones); network servers (e.g., supporting applications such as Web sites, electronic mail [e-mail], and social networks); and network hardware (e.g., routers found in businesses, homes, and the backbone of the Internet) [13, 14, 15].

When people attempt to steal electronic information or commit crime, they leave behind traces of their activities. Properly extracted evidence can be used against the criminals to convict them in a court.

The term *digital evidence* means "any probative information stored or transmitted in digital form that a party to a court case may use at trial [27]". Digital evidence encompasses any and all digital data that can establish that a crime has been committed or can provide a link between a crime and its victim or a crime and its perpetrator.

1.3 Reliability of Digital Evidence

All evidence must meet certain legal requirements before being produced in court. Braid [19] has defined five properties that evidence must have in order to be useful. The first property is that digital evidence has to be *admissible*. That is, evidence must be able to be used in court. Failure to comply with this rule is equivalent to not collecting the evidence in the first place, except the cost is higher. Another property is *authenticity*. Evidence must be tied to the incident in order to prove something. Moreover, the evidence must be shown to relate to the incident in a relevant way. *Completeness* is also another property that dictates the admissibility of digital evidence. It's not enough to collect evidence that just shows one perspective of the incident. Not only should evidence be collected that can prove an attacker's actions, but also evidence that could prove their innocence. Another

property is *reliability*. More specifically, the evidence collection and analysis procedures must not cast doubt on the evidence's authenticity and veracity. One more rule is that of *believability*. The evidence that is presented should be clearly understandable and believable by a jury. After all, there's no point in presenting a binary dump of process memory if the jury has no idea what it all means.

1.4 Challenges of computer forensics

Electronic crime is difficult to investigate and prosecute, as criminals use different tools and techniques to thwart computer forensic experts from extracting digital evidence. In general we can call these challenges as anti-forensics that frustrates forensic tools, investigation and investigators. Some of the primary goals of anti-forensics are:

- Avoiding detection that some kind of event has taken place.
- Disrupting the collection of information.
- Increasing the time that an examiner needs to spend on a case.

Some of the major obstacles that cause much trouble for the forensic examiner and the developers of digital forensic software are described in detail in chapter 4.

Chapter 2

Background

2.1 Flash Memory

Flash memory is a type of non-volatile memory that can be electrically erased and reprogrammed. Two major forms of flash memory, NOR Flash and NAND Flash, have emerged as the dominant varieties of non-volatile semiconductor memories utilized in portable electronic devices. NOR flash was first introduced by Intel in 1988, and support high read performance at a smaller capacity range. However, the new technology NAND Flash memory was introduced by Toshiba in 1989, and supports higher capacities with significantly higher read and writes operations.

NOR Flash has typically been used for code storage and direct execution in portable electronic devices, such as cellular phones and PDAs. NAND Flash, which was designed with a very small cell size to enable a low cost-per-bit of stored data has been used primarily as a high-density data storage medium for consumer devices such as digital still cameras and USB solid-state disk drives. NAND Flash can retrieve or write data as single pages, but cannot retrieve individual bytes like NOR Flash.

Table 2.1 shows the major differences between NOR and NAND technologies. It shows why NAND solution is ideal for high capacity data storage, while NOR is best used for code storage and execution, usually in small capacities.

	NOR	**NAND**
Capacity	1MB-32MB	16MB-512MB
Performance	Very Slow erase (5 Sec) Slow write Fast read	Fast erase (3 Sec) Fast write Fast read
Reliability	Standard	Low

Erase Cycle	10,000-100,00	100,000-1,000,000
Life span	Less than 10% the life span of NAND.	Over 10 times more than NOR
Access Method	Random	Sequential

Table 2.1: Basic difference between NAND and NOR

2.2 NAND Flash basics

Since NAND flash is by far the most commonly used non-volatile solid-state media for SSDs, it is quite helpful to understand some basic fundamentals of the structure of NAND flash cell. The basic NAND flash cell is a floating gate transistor with the bit value determined by the amount of charge trapped in the floating gate. NAND flash uses *tunnel injection* for writing/programming and *tunnel release* for erasing the cell [7]:

- Writing (i.e. programming) to a cell causes the accumulation of negative charge in the floating gate, resulting in a "0" bit value for that cell.
- Erasing a cell removes the negative charge in the floating gate, resulting in a "1" bit value for that cell. To change the bit content of a cell from "0" to "1", the cell must be erased. Due to the NAND architecture of sharing bit control lines across multiple storage transistors, erasing a cell requires erasing the entire Erase Block which contains that cell.

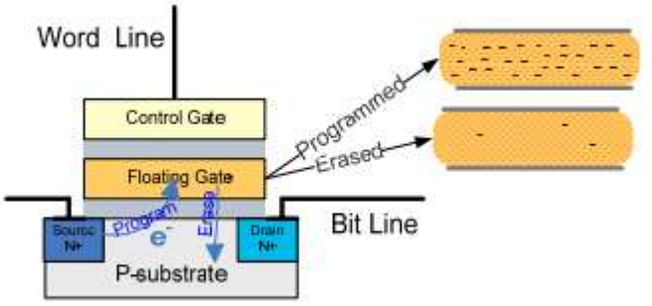


Figure 2.1: NAND Flash Cell Architecture [7]

Today, two NAND flash technologies, SLC (Single-Level Cell) and MLC (Multi-Level Cell), service different applications. MLC NAND and SLC NAND offer capabilities that

serve two very different classes of applications – those requiring the lowest cost-per-bit, and those demanding higher performance and reliability.

MLC NAND flash allows each memory cell to store multiple bits of information, compared to the one bit per cell for SLC NAND flash. When checking for data, an SLC drive only needs to check if the bit is a 1 or a 0. On the other hand, each cell in an MLC drive has four states: 11, 10, 01 or 00. This process takes around 3 times longer to perform. As a result, MLC NAND offers a larger capacity, twice the density of SLC, and at a cost and reliability point targeted for consumer products such as cell phones, digital cameras, USB drives and memory cards [7, 9].

2.3 NAND Flash Challenges

Challenges intrinsic to using NAND flash in a solid state drive (SSD) include:

- Need to erase before writing
- Wear out mechanism that limits service life
- Data errors caused by write and read disturb
- Data retention errors
- Management of initial and runtime bad blocks

With proper flash management techniques, these characteristics of NAND flash can be managed to provide a highly reliable data storage device. Some of the significant factors that resolve these problems in SSD are:

2.3.1 Wear-Leveling

Like all flash memory devices, NAND flash can sustain only a limited number of write and erase cycles before failure. These finite numbers of times information can be erased and written to the memory stick are about 100,000 times [7]. However, this number can be increased by implementing a technique called *Wear-Leveling*. FTL usually employs

some wear-leveling algorithm to ‘shuffle’ cold blocks with hot blocks to even out writes over flash memory blocks.

To understand Wear Leveling, one needs to understand the different addressing schemes in a system. The operating system (OS) uses Logical Block Addressing (LBA) to read and write a block of data from the drive; the flash controller uses physical addresses on the flash to read and write data. Wear Leveling is based upon two mechanisms [7]:

- The controller has the ability to map an LBA address to different physical locations on the flash. The controller uses a mapping table to keep track of the relationship between the logical block and the physical address.
- The presence of spare blocks on the flash for replacement of blocks that contain invalid data.

There are two methods of wear-leveling: dynamic and static. Dynamic wear leveling, as the name says, only wear levels over dynamic or “free” areas. Another method that can be implemented is called static wear leveling. Static wear leveling uses the entire NAND flash.

2.3.2 Error Correction Code (ECC)

One of the key factors to increase flash reliability and write endurance is the implementation of an Error Detection and Correction mechanism. A page can be programmed, erased and read; after each operation it is necessary to verify the status of the page. To perform this verification, flash devices use a verification algorithm that produces a sort of hash/CRC value for each accessed page: the value is then stored in the spare area. This algorithm is generally referred as the Error Correction Code [4]. If a bit error is detected after the read phase, it can be recovered by ECC, if the error is detected after programming or erasing cycle then a block replacement policy is activated. The three most popular Error Correction algorithms that are used with NAND flash technology today are: Reed-Solomon, Hamming and BCH (Bose, Ray-Chaudhuri, Hocquenghem).

2.3.3 Bad Block Management (BBM)

To improve yield and lower cost, all NAND devices are shipped from the factory with some bad blocks which are identified and marked accordingly by the manufacturer. The first physical block (block 0) is always guaranteed to be readable and free from errors.

If ECC reports a non recoverable error, it is required that area be marked as bad. Since the smallest erasable area unit is the block, for any unrecoverable error arising in any page, the whole block to which the page belongs will be invalidated requiring the replacement of such block, so it will not accessed again. Bad blocks identified during NAND lifecycle will be added to the list of bad blocks. Flash manufacturers guarantee that no more than 2% of SLC flash will become bad throughout the 100,000 write/erase cycle lifespan of the flash device [16].

The controller's firmware uses a Bad Block Table to map both initial and accumulated bad blocks and to make sure they are not used in any reading or writing operation. This not only ensures data integrity, but also enhances performance by eliminating the need for repeated write operations resulting from data being repeatedly mapped to the same Bad Block.

To manage invalid blocks, manufacturers do not share a unique rule, but refer to two replacement strategies: Skip Bad Block (SBB) and Reserve Block Area (RBA). In the SBB, when a bad block is detected the flash file system simply skips ahead to the next good block. In the RBA strategy, a predetermined area devoted as reservoir, is used to supply good blocks as a replacement for the bad [5].

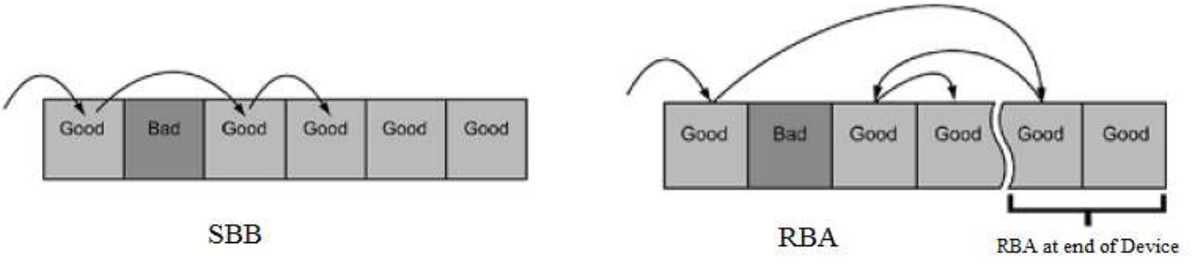


Figure 2.2: SBR vs. RBA (BPMicrosystems, 2008)

2.3.4 Garbage Collection

NAND Flash memory has relatively long erase times, as ERASE operations are done one block at a time. With the FTL this long erase time becomes transparent because instead of erasing a block to be able to rewrite it the FTL simply writes the data to another physical page and marks the data contained in the previous physical page as invalid [6]. This process is called Garbage Collection. The garbage collection is performed when a virtual block is full or the number of free pages in the whole device is lower than a specified threshold value.

2.4 Solid State Drive (SSD)

A Solid State Drive (SSD) is a data storage device that uses Flash NAND memory as its basic component to store data for a long period of time. SSDs offer much faster I/O performance than traditional Hard Disk Drive (HDDs), with no mechanical latency because there are no moving parts. This emerging technology is becoming a more common storage device in the IT environment and is expected to take over the HDD soon.

Figure 2.3: SSD

2.5 HDD Architecture and Operation

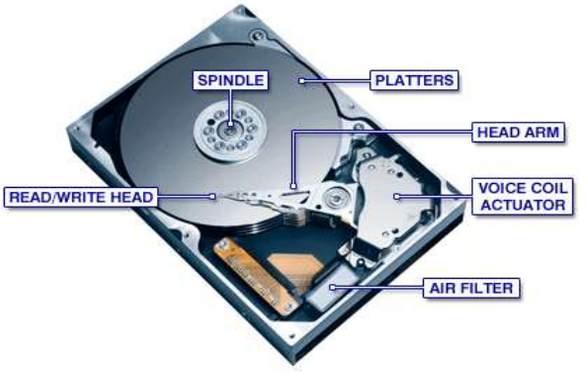


Figure 2.4: HDD

In a standard computer environment, HDDs are the main storage solution for all system and application software, as well as personal data (e.g., files, folders, pictures, etc.). HDD uses rotating magnetic media, in the form of a disk, or a so-called *platter*.

The platter which is shown in Figure 2.4 rotates around several hundred times per second, and it contains the magnetic domains where data is written to. A closer look reveals the sliced division of the platter; each slice is called a *sector* which represents the minimum addressable area of an HDD, typically 512 B. Unfortunately, these moving parts make the HDD susceptible to common HDD setbacks.

An elementary understanding of the inner workings of these storage devices can be helpful in preserving your data. Platters consist of a hard substrate covered with a thin coating of magnetic material. The data are stored on the magnetic surface in structures called *tracks*, which are concentric circles on the disc surface. The drive contains multiple platters that are stacked on top of one another with just enough room between the platters for the read/write heads. Typically, data is stored on both sides of a platter. The platters are connected with a spindle that is attached to a motor.

The head actuator assembly moves the read/write heads to specific locations on the platters. Each side of the platter has its own read/write head and all of the heads move in tandem. The actuator seeks locations on the platters called *cylinders*. A cylinder consists of all the tracks stacked on top of on another at a specific location. Only one platter surface (head) can be read or written at any particular moment.

A Block is the intersection of a track and a sector which is the minimum addressable size of an HDD. This is done by specifying three things: The Cylinder, The Head number and The Sector number.

2.6 SSD Architecture and Operation

Since an individual flash memory package only provides limited bandwidth, flash memory based SSDs are normally built on an array of flash memory packages. As logical pages can be striped over flash memory chips, similar to a typical RAID-0 storage, high bandwidth can be achieved through parallel access. A serial I/O bus connects the flash memory package to a controller. The controller receives and processes requests from the host through a connection interface, such as SATA, and issues commands and transfers data from/to the flash memory array. When reading a page, the data is first read from flash memory into the register of the plane, then shifted via the serial bus to the controller.

A write is performed in the reverse direction. Some SSDs are also equipped with an external RAM buffer to cache data or metadata.

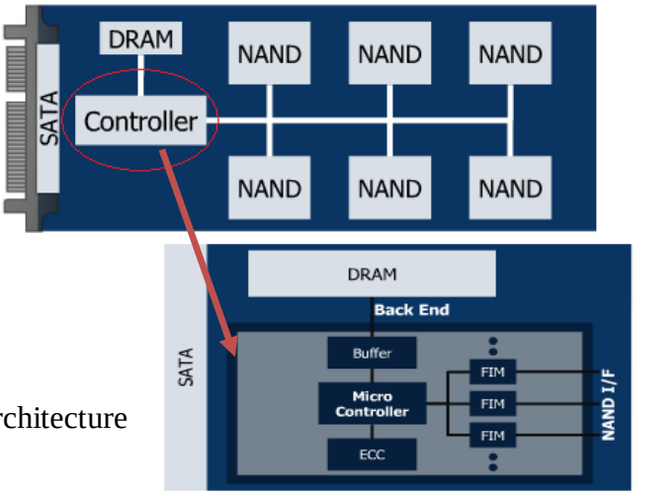
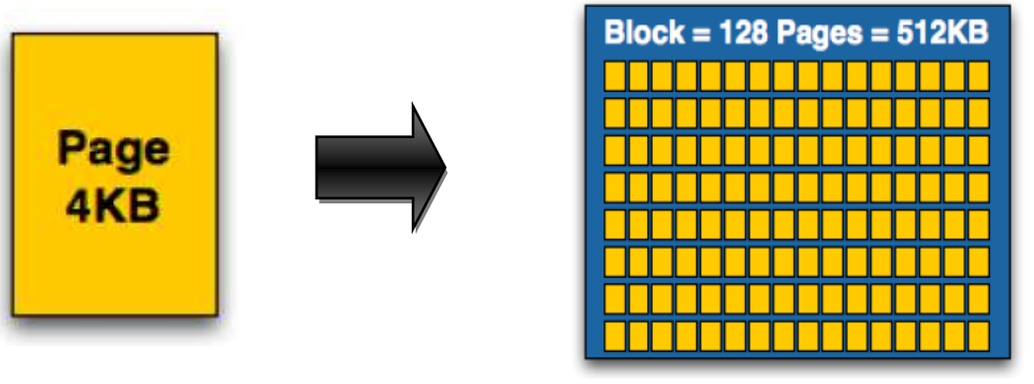


Figure 2.5: SSD Architecture

The NAND flash memory package is composed of one or more *dies* (chips). Each die is segmented into multiple *planes*. A typical plane contains thousands (e.g. 2048) of *blocks* and one or two registers of the page size as an I/O buffer. A block usually contains 64 to 128 *pages*. Each page has a 2KB or 4KB data part and a metadata area (e.g. 128 bytes) for storing Error Correcting Code (ECC) and other information. Exact specification data vary across different flash memory packages. [7]

Arrays of cells are grouped into a page, arrays of pages are grouped into blocks.



Blocks are then grouped into planes, and you will find multiple planes on a single NAND slash die.

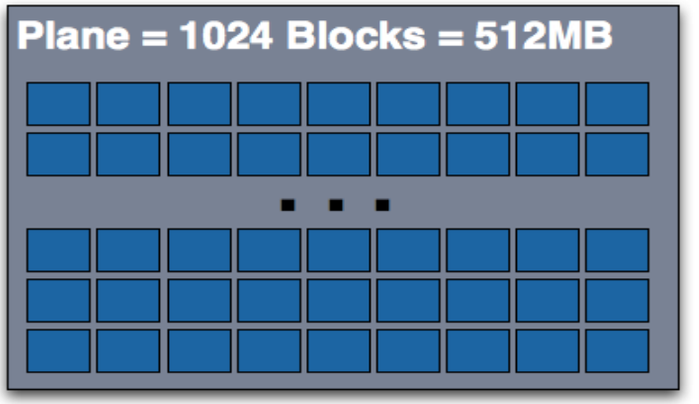


Figure 2.6: Organization of NAND memory

Flash memory supports three major operations, *read*, *write*, and *erase*. Read is performed in units of pages. Each read operation may take 25µs (SLC) to 60µs (MLC). A page is the smallest area of the flash memory that supports a *write* operation and consists of all the memory cells on the same word line. A Block is the smallest area of the flash memory that can be *erased* in a single operation. [7]

A critical component, called the *Flash Translation Layer* (FTL), is implemented in the SSD controller to emulate a hard disk and exposes an array of logical blocks to the upper level components. The FTL plays a key role in SSD and many sophisticated mechanisms are adopted in the FTL to optimize SSD performance.

The controller is regarded the brain of the SSD. It contains several elements, Flash Interface Modules (FIMs), Microcontroller, Buffer and an Error Correcting Code (ECC) memory. The FIMs physically and logically connect the controller to the individual NAND Flash devices. Each one of those FIMs is capable of talking to a number of NAND Flash components, and to the extent of adding more FIMs; the performance of the SSD will increase. There are no real specifications of the internal life of a solid state drive. One can list structures and mechanisms that are really needed to do the work. But solid state drives are a relatively new technology. Every vendor tries to keep their knowledge top secret and hide the techniques that make their own drive better and faster than others. Therefore a solid state drive can also be seen as a black box. It just does the

right thing, but it is not possible to clearly see all the internal details and intelligence. It is hidden from users and developers.

2.7 SSD Vs HDD

Figure 2.7: HDD and SSD

Most important parts of both storage devices are already mentioned earlier in the previous section. Here we are clearly interested in discussing the main areas that can make a difference. As we can see from the picture above, the main difference between a hard drive and a solid-state drive is that the HDD have a lot of moving parts like platters and read/write heads which makes it very prone to vibrations or shocks. Whereas SSD is a lot more well solid. The lack of moving parts in the SSD is what gives it so many of its advantages, such as its speed and durability. There is a lot less that can go wrong when you don't have to worry about things wearing out or breaking.

Some of the motivations for "Why NAND flash is added to a PC?" From different manufactures are dealing with issues in PC architecture today. Normally computers with conventional HDD take long boot times for OS and applications. In addition HDD latency time is high and makes the computer slow. The only bottleneck in computer system performance is the traditional hard drive. So introducing a NAND flash to computers easily solves such problems by enhancing the speed and performance of computer, in which the effect can easily spot during boot.

Of course, conventional hard drives are still a lot more widespread than SSDs, and it will be a while before they are rendered obsolete. While SSDs are much faster it is much easier to get your hands on a very high capacity HDD.

2.5" SATA 3.8Gbps SSD		**2.5" SATA 3.8Gbps HDD**
Solid NAND Flash based	**Mechanism type**	Magnetic rotating platters
64 GB	**Density**	80 GB
73 g	**Weight**	365 g
Read: 100 MB/s Write: 80 MB/s	**Performance**	Read: 59 MB/s Write: 60 MB/s
1W	**Active Power Consumption**	3.86 W
20G (10~2000HZ)	**Operating Vibration**	0.5 G (22~350 Hz)
1500G for 0.5 ms	**Shock Resistance**	170 G for 0.5 ms
0 °C -70 °C	**Operating Temperature**	5 °C -55 °C
None	**Acoustic Noise**	0.3 dB
MTBF > 2M hours	**Endurance**	MTBF < 0.7 M hours

Table 2.2: SSD Vs HDD

Chapter 3

Problem Definition/Goals

3.1 Motivation

Much research work has been done on digital forensics and file recovering from magnetic media (HDDs). To the contrary the number of research works on Solid State Drives (SSDs) is very small. The reason is due to the fact that SSDs are new emerging technologies. The solid state drive is one of the most common storage systems that may be used to contain the computer criminal evidence. Recently few researchers have released some papers on the challenges of SSD during forensics, however the results are highly affected by some features of SSD. Every SSD device is quite unique in its storage architecture and data distribution pattern, which is managed by the unique controller chip inside the device. This controller and algorithms that operate inside are secretly kept by the vendors. Indeed it is one hindrance not to achieve a better result from the forensics works.

3.2 Goals

The goal of this thesis is to get a better understanding of Flash memory in general and of Solid state drives in particular. Then this knowledge can be used to build forensics techniques for recovering data. To successfully recover data from solid state drives requires an in-depth knowledge of how data is stored and addressed throughout NAND flash memory for that vendor's specific implementation. The recovery process performs different phases including imaging disk and verifying.

Solid state drives are relatively new storage devices, but due to their advantages over the traditional hard disks they are becoming more popular and eagerly expected by users to replace hard drives as the main storage medium.

3.3 Limitations

Solid state drives do have some limitations. The most widely known disadvantages of SSDs are price, storage capacity and data recovery.

The most important downside of the solid state drive is its excessive price. As of January 2011, you will pay about £0.03 ($0.05) per GB for a large-capacity hard drive, Whereas even the cheapest 128GB SSD will set you back £1.40 ($2.20) per GB. Yes, they do aim to do different jobs, but it just goes to show that it will be a while before SSDs are ready to fully replace HDDs. [9]

The second drawback of SSDs is they have limited storage capacity when compared to conventional hard disk. Obviously it is important to get a drive that is big enough to hold your operating system, programs and data. Where normal HDD of 500 GB and more are no longer a rarity, you will be hard-pressed to find a solid state drive with a storage capacity much higher than 128 GB.

Another major downside to Solid state drives is the difficulty of data recovery. Usually it takes a long time for a hard drive to fail, giving you plenty of warning and the opportunity to back up the data to another source. You do not get that with a solid-state drive; when a drive fails, it fails completely and instantly. When that happens, it is virtually impossible to retrieve the data. [9]

Chapter 4

Related works

4.1 Introduction

Previous research into flash memory forensics has focused mainly on portable devices, such as thumb drives, phones, and PDAs. Flash chips have been present in those forms for years, so the existing research is much more comprehensive regarding specific single-chip implementation rather than on larger, complex flash arrangements such as an SSD. In addition the documents introduced for non-volatile memories present in nowadays, explain in detail about the devices on how they really work and which challenges they pose to the forensic investigations.

These flash forensic papers generally can be seen in two different ways. Papers that describe methods for recovering data from flash memories, and papers that deals with the acquisition (Logical or Physical images).

4.2 Flash data recovery

In “An Integrated Approach to Recovering Deleted Files from NAND Flash Data,” James Luck & Mark Stokes (2008) have showed the techniques on how to recover deleted files from NAND flash memories, particularly focusing on recovering media files from mobile devices. In “Data Remanence in Flash memory Devices,” Sergei Skorobogato proposes a method to extract remnant data from flash cells that have been erased. Remnant data is information that can be recovered from a storage media after new information has been written over old, in attempts to delete or overwrite the old information.

Breeuwsma et al, (2007) in "Forensic Data Recovery from Flash Memory," introduce flash memory and show how it is possible to acquire data from USB memory sticks. Moreover the paper suggests a low-level approach for forensic examination of flash memories and describes three low-level data acquisition methods for making full memory copies of flash memory devices. Phillips et al. (2008) experimentally tested data recovery of damaged flash drives. Phillips' findings show that the physical destruction of the data on a flash chip is extremely difficult to do. Over-voltages, smashing, water, and incineration all proved ineffective in destroying all the data.

More recently a detailed findings contained in "Solid State Drives: The Beginning of the End for Current Practice in Digital Forensics Discovery?", Graeme B. Bell and Richard Boddington (2010) of Murdoch University in Perth, Australia, explored the effects of SSD garbage collection on data retention. After conducting a series of experiments comparing a sample Corsair 64GB SSD with a conventional Hitachi 80GB magnetic hard drive (HDD), the team found a data recovery problem caused by the 'garbage collection' or purging algorithms used in SSDs to keep them at peak performance [8]. Comparing SSD with the equivalent HDD (all data are recoverable), the team concluded that "Even in the absence of computer instructions, a modern solid-state storage device can permanently destroy evidence to quite a remarkable degree, during a short space of time, in a manner that a magnetic hard drive would not [8]."

4.3 Flash Memories Acquisition

The second category of papers deals with the acquisition process on flash memories. More clearly this further can be categorized in two separate parts as forensics tools acquire data from a device in one of these two ways: physical acquisition or logical acquisition. Physical acquisition implies a bit-by-bit copy of an entire physical store (e.g., a memory chip), while logical acquisition implies a bit-by-bit copy of logical storage objects (e.g., directories and files) that reside on a logical store (e.g., a file system partition). Physical acquisition has advantages over logical acquisition since it allows

deleted files and any data remnants present (e.g., in unallocated memory or file system space) to be examined, which otherwise would go unaccounted [20].

4.3.1 Physical acquisition

In the paper "Forensic imaging of embedded systems using JTAG (boundary-scan)," Marcel Brueeuwsma proposes using the Joint Test Action Group (JTAG) as a physical means to produce an image of stored data in flash memory on an embedded device. The approach described in this paper uses a JTAG test access port to access memory chips without removing these chips from the device. Its advantage is it minimizes the risk of damaging chips during desoldering and allows to access memory chips like SDRAM that cannot be removed from the device [21].

4.3.2 Logical acquisition

In "Data Acquisition from Cell Phone using Logical Approach," Keonwoo Kim et al. described a forensic tool that logically acquires and analyzes data stored in the NAND flash memory and NOR flash memory of the cell phones.

In the paper "An overall assessment of Mobile Internal Acquisition Tool," Alessandro Distefano and Gianluigi Me, proposed a mobile forensic tool called MIAT (Mobile Internal Acquisition Tool) designed to acquire Symbian (and windows Mobile) Smartphones internal memory data without cables, directly from the internal memory slot. The application uses OS APIs to scan and copy the entire internal memory file system to a removable memory card. As per the author's motivation, the adoption of this methodology forces saving hardware tools like USB cables specific for each device or additional equipment like a notebook PC to perform the acquisition. A further benefit of using the MIAT is represented by the parallelism: MIAT can be used to seize n smartphones simultaneously, using n memory cards [22].

In "Analysis of USB Flash Drives in a Virtual Environment," Derek Bem and Ewa Huebner examine the application of the virtual environment in the analysis phase of a computer forensics investigation of USB flash drives. A dd based tool is used for acquiring a logical level image of a USB storage device and stores the image in the dd format or a proprietary format for further investigation [23].

Chapter 5

Forensic Data Recovery

This chapter mainly focuses on the digital forensics data recovery process from solid state drives. Particularly we will see all the phases performed in data recovery and some challenges of SSD during the process. However it is very important to start with how data are stored before and after deletion.

5.1 Data Remanence in SSD

Data Remanence is a term used to describe the residual data remaining after a certain kind of deletion has been performed [28]. After storage media is erased there may be some physical characteristics that allow data to be reconstructed. Understanding how SSDs (specifically NAND cells) retain data after deletion and their characteristics may play a big role in forensics data recovery.

Modern Operating Systems (OS) talk to hard drives using Logical Block Addressing (LBA). While hard drives are rotational media, logical block addressing organizes sectors on a hard drive linearly. When you try to save a file, the OS simply issues a write command for your file at a specific logical block address, for example LBA 15.

The OS knows what LBAs are available and which ones are occupied. When you delete a file, the LBAs that point to that file on your hard disk are listed as available. The OS simply removes the pointer from the file system directory which actually points to the Physical Address on the hard drive. The data you have deleted has not actually been removed and it does not get wiped until those sectors on the drive are actually overwritten.

Believe it or not, SSDs actually work the same way. The Flash Translation Layer (FTL) in a SSD controller maps LBAs to pages on the drives. The table below explains what happens to the data on the SSD depending on the action in the OS: [17]

Action in the OS	Reaction on a HDD	Reaction on an SSD
File Create	Write to a sector	Write to a page
File Overwrite	Write new data to the same Sector.	Write to a Different Page if possible, else Erase Block and Write to the same Page.
File Delete	Nothing	Nothing

Table 5.1: OS actions with possible reactions from HDD and SDD

When you delete a file in your OS, there is no reaction from either a HDD or SDD. It is not until you overwrite the sector (on a hard drive) or a page (on a SSD) that you actually lose the data. File recovery programs use this property to their advantage and that's how they help us recover deleted files.

The key distinction between HDDs and SSDs however is what happens when you overwrite a file. While a HDD can simply write the new data to the same sector, a SSD will allocate a new (or previously used) page for the overwritten data. The page that contains the now invalid data will simply be marked as invalid and at some point it'll get erased.

5.2 How Data is deleted in SSD

Flash-based solid-state drives (SSDs) differ from hard drives in both the technology they use to store data (flash chips vs. magnetic disks) and the algorithms they use to manage and access that data. So assuming that the erasure techniques that work for hard drives will also work for SSDs is dangerous [10].

SSDs use flash memory to store data. Flash memory is divided into pages and blocks. Program operations apply to pages and can only change 1s to 0s. Erase operations apply to blocks and set all the bits in a block to 1. As a result, in-place update is not possible.

Unlike standard hard drives that store the file in a single location, flash drives can make multiple copies of the file on the flash derive and just points to the latest version [10].

SSDs can retain content of a file even after erasing. The difficulty of reliably wiping SSDs stems from their radically different internal design. Traditional ATA and SCSI hard drives employ magnetizing materials to write content to a physical location that's known as the LBA, or logical block address. SSDs, by contrast, use computer chips to store data digitally and employ an FTL, or flash translation later, to manage the contents. When data is modified, the FTL frequently writes new files to a different location and updates its map to reflect the change.

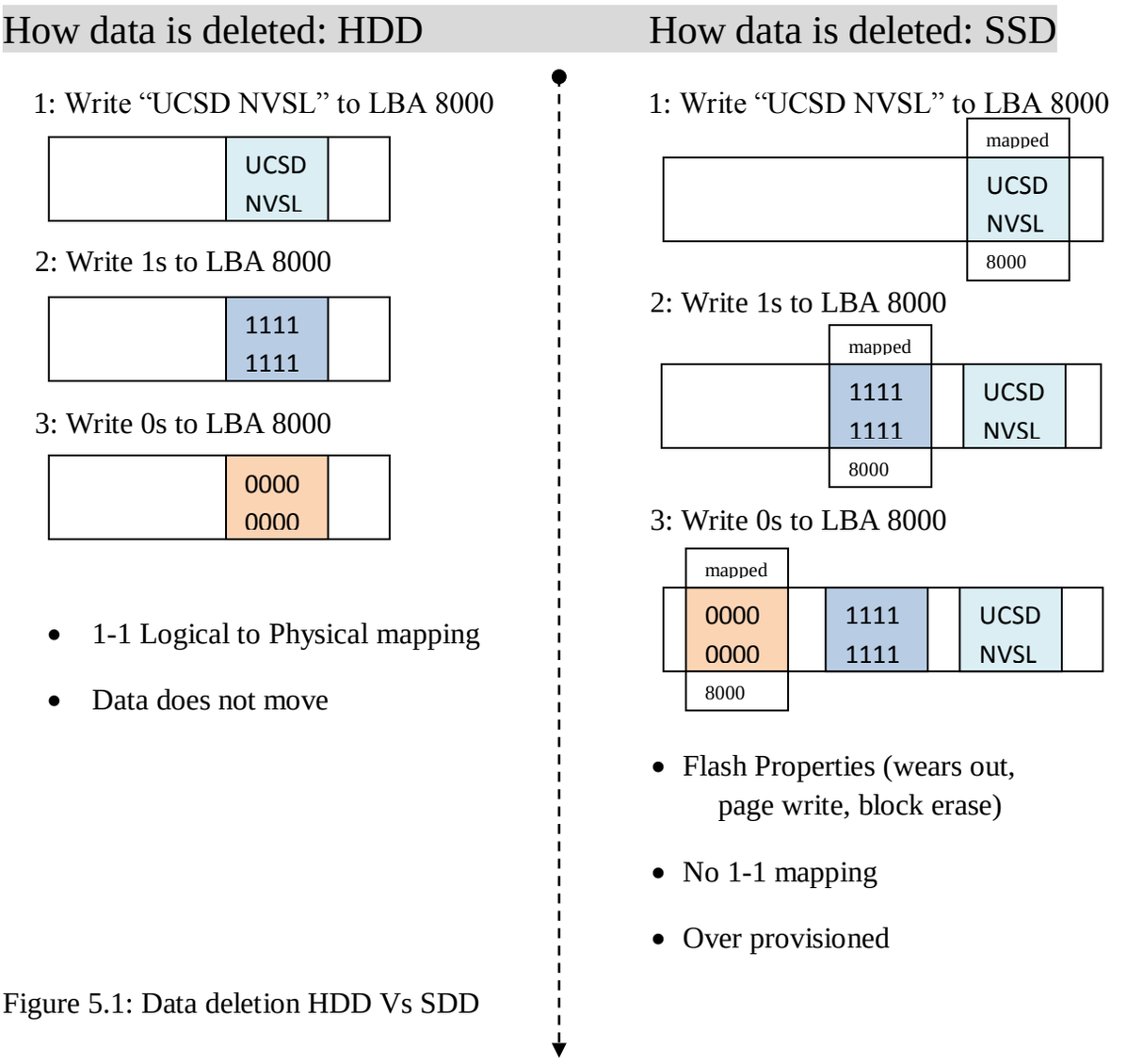


Figure 5.1: Data deletion HDD Vs SDD

5.3 Permanent Destruction of Data

This process might be considered an anti-forensics activity, as different techniques are taken into action for permanently wiping the contents of the solid state drive (SSD), which challenges the computer forensics experts to extract the evidence during the investigation. For erasing data permanently there are different techniques that can be applied on the SSD depending on the user choice and advantages provided by each of the techniques. Some of the most common methods that are frequently used are listed below.

5.3.1 Secure deletion via software

Software erasers work by overwriting a specific data on the drive. If a drive is overwritten enough times, the underlying data will become indecipherable. The more time a drive is overwritten, the more difficult it becomes to forensically restore any of the original data from the drive. ATA Secure Erase is the most effective method for SSD.

However, recent research works (*University of California San Diego study*) revealed that certain processes to wipe data from SSDs actually left data behind. This means there is still a probability to recover a wiped drive [10].

5.3.2 Data Encryption

Recently SSD with the capability of encrypting user data before recording have been introduced. Such drives provide protection of data should the computer or drive be lost or stolen, and even provide high protection from forensic data recovery. The downside here is the algorithm used to encrypt the data might be broken with an immense effort from anyone.

5.3.3 TRIM command

The TRIM command is introduced to enhance the *write* performance, by preparing empty pages when we want to write a new file. A TRIM command allows an OS to inform a solid-state drive about which data blocks are no longer considered in-use and can be wiped internally. When you delete a file, the OS sends a TRIM command for the LBAs covered by the file to the SSD controller. The controller will then copy the block to cache, wipe the deleted pages, and write the new block with freshly cleaned pages to the drive.

A TRIM command purges both data and the link to it, which diminishes the chance of data recovery almost to zero.

When the operating system informs the controller to delete a certain file using the TRIM command, the controller cleans the cells at a physical level. This means that the electrons that are stored in the NAND flash cell are grounded and emptied from the cell.

5.4 Data Recovery

Solid-state devices share many of the same failure modes exhibited by HDD's. Since SSD's are a direct replacement for HDD's in most applications and are subject to many of the same stresses, some SSD failure modes are similar to those of HDD's. The most significant difference between the two technologies is an SSD's lack of moving. As a result, SSD's have no instances of mechanical failure. Shared failure modes aside, the techniques and processes for recovering data from the two storage technologies differ greatly.

Unlike the traditional HDD, the techniques for performing forensically data recovery from SSD are quite different and difficult. Despite the advantages of SSD, it presents forensics challenges that demand further research. Mitchell (2009) and Antonellis (2008) found that data recovery in SSD is extremely difficult and also impossible in some cases due to the fact that the implementations are non-standardized, controller technology is complicated and algorithms are proprietary and different from vendors to vendors. Highly sophisticated data carving technology are required even when the data recovery is possible.

5.4.1 Reasons for Data Loss from SDD

Storage Medias are responsible for storing valuable data which could be lost due to various reasons. There are two main causes of data loss from solid state Drives (SSD),

Logical and *Physical*. In this thesis, we only discuss software techniques for recovery of data with a focus on digital forensics.

Logical – The solid state Drive is functional, but data is lost or inaccessible because the file system structures that are used by the operating system to locate your data have become corrupted, overwritten or possibly even orphaned from the active file system. This may result from several scenarios such as: deleting, formatting and overwrote files.

Physical – The Solid State Drive is NOT functioning properly. SSD drives are similar to USB flash sticks in architecture, but with more complexity. A modern SSD drive consists of a controller chip, a memory cache and multiple NAND type memory chips. Data may not be accessible if any of these components are damaged. Typical issues with SSDs include: bad cell, read error, ESD damage power spike, and Memory chip bad.

5.4.2 How we recover data from SSD?

If the problem has been determined to be a *Logical* issue, standard software's can be used to recover the data from SSD by directly reading the NAND memory blocks. But before starting to recover the data it's very important to create an exact image of every readable byte of data from the SSD. Once an image is secured, we never access the original device again, preferring to work on a copy thus preserving the original media. Logical issues are resolved by careful analysis to determine exactly what went wrong. However, damaged and overwritten file system structures are rebuilt manually or using special utilities.

The main focus of this thesis is recovering data for logical failures.

5.5 Data Recovery Process

Depending on the type of data recovery, you will have five phases. This is the same regardless of Hard Drives or Solid State Drives. The difference is what you have to do to fix the solid State Drives compared to the repair process for mechanical hard drives.

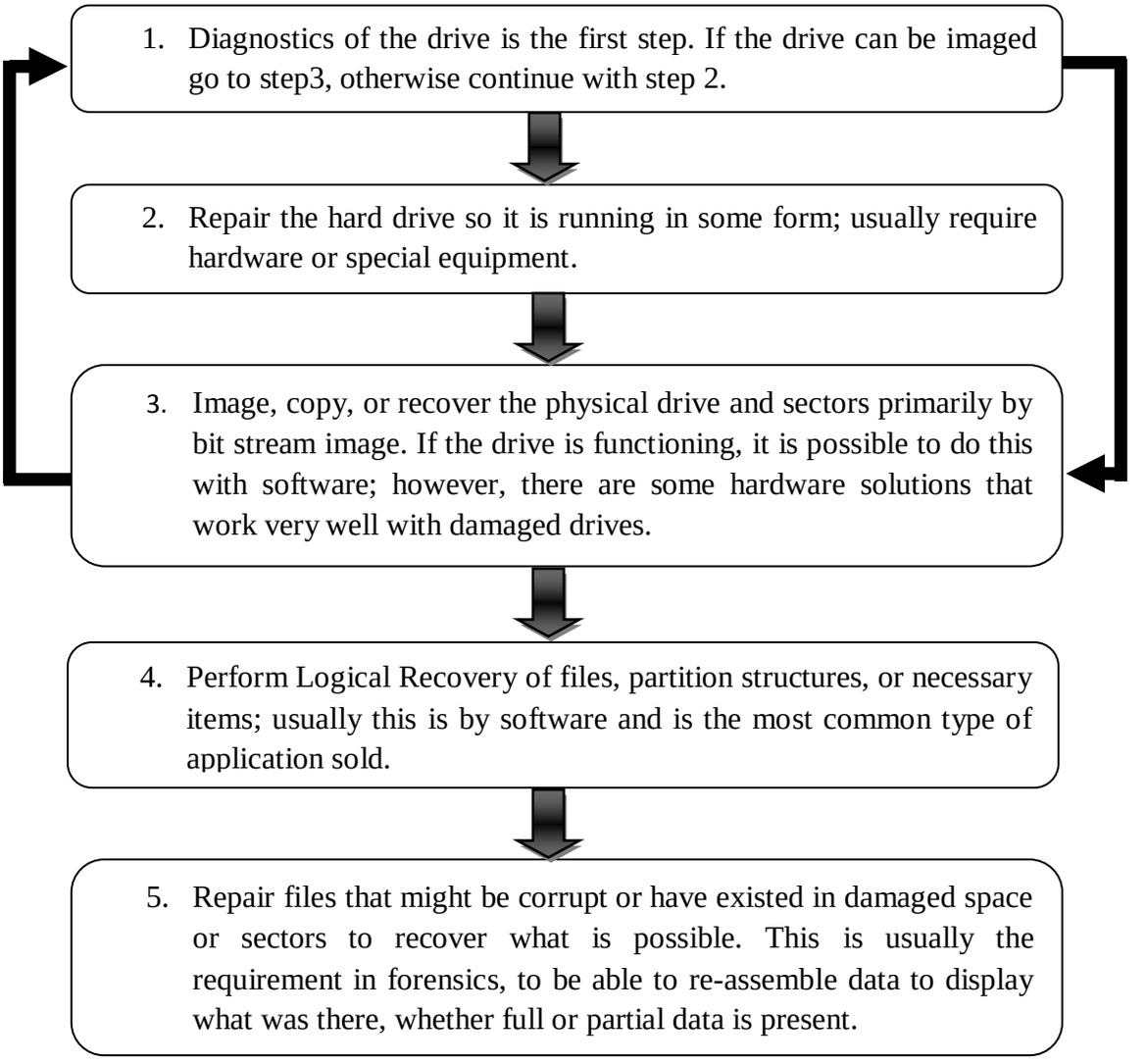


Figure 5.2: Data recovery phases [30]

In the first step we can diagnose the problem in many ways, but one of the easiest is to attempt to image the drive using hardware or software. There are some pieces of software that can talk to the drive and help you diagnose the type of problem before continuing.

The second step is totally about repairing the damaged hardware part if the problem is physical failure.

5.5.1 Acquisition/Imaging

The point of the acquisition is to copy and preserve the state of data that could be evidence. The forensic acquisition (physical) of media refers to the process of making a bit-for-bit copy (e.g., a memory chip), or image file, of a piece of media, which image files frequently used in civil or criminal court proceeding [18]. Therefore, completeness and accuracy of the acquisition process are required. In addition, the source of evidence must remain not altered by attackers or by normal processes innocently. Imaging is very important in data recovery. Even though many data recovery companies did not image their drives, but it is very valuable in forensics to image the original drive and work on the working copy. This allows the investigator or computer forensic expert to maintain the state of the original drive without making changes to it.

5.5.1.1 Never work on the original evidence

Although it is easier to do analysis directly on original evidence it is not best practice in computer forensics. Evidence would be exposed to the risk of contamination. One of the cardinal rules in computer forensics is never work on the original evidence. Why? Because evidence is very fragile in nature and can easily be modified, duplicated or damaged. Evidence must be handled properly and very easily destroyed. With only one strike on keyboard evidence could be accidentally destroyed or modified. [11]

Bit-stream image is the exact replica of the original device. As distinct from the normal media backup, the bit-stream image will duplicate deleted files, file slacks, swap files, hidden areas and unallocated spaces.

5.5.1.2 Verify Image File Integrity

The authenticity of the evidence is one criterion for the evidence to be admissible in court. In general, testimony clearly establishes that the exhibits presented as evidence are identical to the original and the content has not been changed by any means.

Hashing is an effective forensic technique for examining information on computers to identify, verify, and authenticate data. Hash functions are often used for matching. This proves especially useful in the context of forensic analysis. Analysts often use hash functions in court to prove that the storage drive image remained unchanged under forensic analysis.

The accuracy of the bit-stream image must be validated. A mathematical algorithm, such as MD5 or SHA1, is used to calculate a hash value for the original SSD and compute another hash value for the bit-stream image. Both hash values must be the same to verify that both images are identical.

A hash value is computed by a hash function, which is a well-known, openly published algorithm that takes a stream of bytes (such as an electronic file) as input and calculates a fixed-size binary data item as the output. The two most popular hash functions are MD5 and SHA-1. MD5 will take a file and produce a 128-bit binary data while SHA-1 will produce a 160-bit binary data.

5.5.1.3 SSD peculiarities in the acquisition process

One immense hindrance to computer forensics from flash memory technologies is that, nobody seems to be able to set a definitive point on how others can use or implement flash technologies. What really happens inside a flash memory is beyond the knowledge of the computer forensic experts, as flash manufacturers kept the algorithms from being revealed to others.

Garbage collection:

> If garbage collection were to take place before or during forensic extraction of the drive image, it would result in irreversible deletion of potentially large amounts of

valuable data that would ordinarily be gathered as evidence during the forensic process - we call this 'corrosion of evidence'[8].

Bad blocks:

It is unclear who really manages bad blocks and how, wear leveling can be host dependent (that is, managed by the OS) as well as implemented in the flash itself (like the embedded FTL). If it doesn't clear how it works, then it is not possible to decide how to manage it. If the FTL used is embedded in the flash memory, then it will be difficult to access and manage bad blocks because they will be hid to the host file system. Otherwise, if the FTL is supplied from the host, then we can have a chance to manage them properly and have direct access to bad blocks [3, 6].

5.5.1.4 Hardware-based Vs. Software-based imaging tools

There are two types of disk imaging tools in the market, namely hardware-based and software-based. Hardware-based disk imaging tools usually have much better performance over software-based disk imaging tools. Corresponding to the performance, the cost is much higher than the software-based disk imaging tools. Hardware disk imaging tools usually come in a toolkit style with plenty of accessories such as different types of physical interfaces, adapters and cables to acquire different type of devices. Hashing verification, write blocking and read multiple devices simultaneously is the most common function hardware-based disk imaging tools will provide. Logicube Talon, HardCopy 3 from Voom Technologies, Data Copy King from SalvationDATA and TableauTD1 from Guidance Software are some commonly used hardware disk imaging tools.

The second type of disk imaging is software-based. In some case it requires hardware for performing efficient image. For example if we are imaging Windows-based application to image, then we must use a write blocker to ensure that no data is written back to the solid state drive. If we are using Linux, then a write blocker is not required because we can manually mount the drive as 'read only.' Forensic imaging under Windows therefore

requires the use of a special 'write-blocker,' a hardware mechanism that allows reading from, but not writing to, the drive. FTK Imager, EnCase and DD are common and widely used imaging tools. DD was first released as a utility of UNIX. DD is one of the oldest imaging tools and it produces raw image format.

During forensic acquisition and analysis, it is possible to write to the evidence drive accidentally. Since this lead to the immediate dismissal of the evidence, the investigator should take care of it and ensure that using a write blocker. There are two types of write blockers: software write blocker and hardware write blockers. A *software write blocker* replaces a drive access interface on a computer with external drives. It blocks any commands that could modify a storage drive [29]. *A hardware write blocker* is a hardware device that physically attaches to a computer system. Its main purpose is to intercept and block any modifying commands from reaching the storage device [29].

5.5.2 Logical Recovery

A storage device may be split up into multiple partitions (logical entities where each partition appears as a separate storage device). Each partition would then have its own directory and file system [2]. Criminals with good knowledge of computers can hide data quite easily by partitioning a hard drive and encrypting the resulting partition, which makes it a tough task for the investigators.

This step mainly falls on recovering files and partitioned structures from SSD, when you come up against a drive error (not hardware failure).

The partitioning on seized solid state drives should also be evaluated as it is possible that hidden partitions and/or partitions that have been formatted with an operating system other than a DOS compatible operating system. When hidden partitions are uncovered, they should be analyzed for evidence and their existence should be documented.

Criminals usually try to thwart detection by deleting partitions using tools such as FDISK. After using the tool, the partition itself and all of its data are untouched by the process.

However, by recovering these deleted partitions we can evaluate and extract deleted partitions. The partition recovery process is important in case of data recovery.

Before dealing with recovering deleted or lost partitions it's pretty important to briefly describe the basic terms which help in understanding partitions and partition recovery. Master Boot Record (MBR), Partition Table and Volume Boot Record (VBR) are among the very important concepts we are going to look next.

A Master Boot Record is the 512 byte boot sector, which is the first sector (LBA sector 0) of a partitioned storage device, such as a hard drive or in our case an SSD. One of the MBR's tasks is to hold the partition table and that is the only thing we care about. The partition table, a 64-byte data structure used to identify the type and location of partitions on a storage device, conforms to a standard layout independent of the operating system. Each partition table entry is 16 bytes long, with a maximum of four entries. The location of the first partition depends on the type of operating system, but most of them share the same starting point which is sector #63. For example the first Windows XP partition starts at sector #63, the middle of a SSD page.

The following picture shows a partial printout of an MBR revealing the partition table from a computer with three partitions. When there are fewer than four partitions on a disk, the remaining partition table fields are set to the value 0: [12]

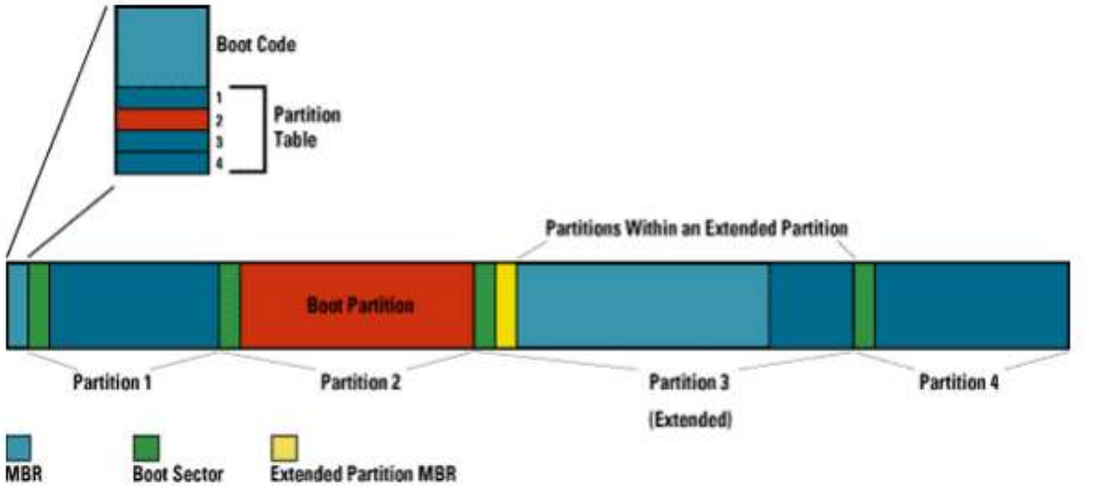


Figure 5.3: MBR and Partition Information

5.5.3 Recovering Partitions

Normally file recovery is limited to a file or a hundred files even. But when a computer partition has gone missing it requires the retrieval of tens of thousands of files. In computer forensics recovering data from the damaged, formatted, deleted or corrupted partitions of different operating systems is very essential for undergoing investigation of the evidence. The data recovery process can take place by identifying the device and locating all partition(s) on the inaccessible drive. Following this the file structure of selected partition and the data area of the inaccessible drive will be scanned in two different ways:

Case 1: The first case and easiest way is MBR recovery. This is performed by a simple quick scan. The scan finds all the partitions when there is some corruption with the MBR of the operating system or it is missing or deleted. This scan easily locates all the lost partitions in no time and allows the data recovery components to recover partition data.

This quick scan also works out to repair or rebuild partition table and recover lost or deleted partitions except for physical damage. Through modifying the partition table or boot sector we can recover the partitions, which means without any contact (write data) with the drive we want to recover.

Case 2: The second type of scan is advanced one performed for lost partitions. This disk scanning method is extremely helpful when a drive is re-partitioned or re-sized. A repartitioned disk not only loses all of your data, it also loses the original file system structures. It is not possible to 'un-partition' the repartitioned disk in a way you can deal with a formatted drive. Finding older or reformatted partition is extremely difficult, this is the reason why this method scans the drive heuristically and tries to locate and validate lost partitions.

A number of tools are available for partition recovery, each of which has various features that can make it easy to restore data that may have been lost from accidental deletion or damage to the partition.

5.5.4 Bad Sector/Block

The last phase of forensic data recovery takes place right after we have recovered the files. Most of the time, we might encounter some specific problems with the media, bad sector and damage to the drive. Due to such setbacks there may be corrupted files after we have recovered them. Usually these problems can be solved with the help of up-to-date tools designed to handle such problems. In computer forensics dealing with these problems after data recovery is suggested and very useful. Particularly repairing files from bad sectors are the focus of this phase. A bad sector is an unusable part or subdivision within a track on a magnetic or optical disc located on a computer's hard disk or flash drive. A bad sector is typically formed as a result of physical damage of some sort, or rarely, the operating system's inability to access the information. The physical damage occurs to the disk surface or as a result of flash memory transistor failure. Once the bad sector is identified by disk utility software - such as SCANDISK or CHKDSK it marks the sectors that have failed so that the OS can skip them in the future [24].

From the forensics point of view it's not suggested to recover and analyze data on defective storage device as it may cause further loss and damage on the storage media. As described in the early phase the most efficient and common approach is to image the data (including the bad sectors) to a stable media and then proceed to recover and analyze data from the image. Still some files recovered from bad sectors or damaged ones might require further repairing with the help of other tools. WinHex is a hexadecimal editor that allows you to read sectors and edit them. It's a powerful application that you can recover data or read areas of a disk that contain deleted or damaged data. WinHex can help in repairing bad sectors and even by cutting out the information we want to investigate.

Chapter 6

Forensics Data Recovery Tools

Digital forensics evidence is normally latent by nature; it must be viewed and recovered through the use of tools. Forensic tools are used to analyze digital data often find evidence that someone did or did not commit a crime. As the tool output may be evidence in a court trial, it must meet certain legal requirements. Forensic tools are used in all phases of evidence processing, however this chapter focuses on tools that are used for digital forensics data recovery from flash memory devices (SSD, SD, USB and others).

There is a lack of information on forensic recovery of solid state drives. This is most likely because they are still relatively new to the consumer market. However, NAND-based solid state drives allows you to use traditional forensic tools or slightly modified ones to recover files in slack space as well as normally deleted files [26]. Besides the complete forensic toolkits (like EnCase and FTK) some other tools that can be used to collect electronic evidence from solid state drives are described here. These tools have the ability to extract and recover deleted files and formatted drives for forensic investigation.

Few tools are described below with detailed information.

EnCase

EnCase is a computer forensics tool which is a very popular and widely accepted in the court of law in forensic investigation. EnCase helps examiners to easily analyze large volume of digital evidence and view files, file slack and unallocated data. EnCase contains tools for several areas of the computer forensic process: like for data acquisition, file recovery, indexing and file parsing. The data recovery tool of EnCase has been used successfully in various cases to convict criminals to the court. Investigators can use

EnCase to collect data from various storage sources including: HDD, Flash drives, Floppy disks, CD ROMs, Digital cameras and others.

Publisher	Guidance Software
Latest Version	EnCase 6.11.2 – Forensic Edition
License	Commercial
Platform	Windows 95/98/NT/2000/XP/2003 Server, Linux Kernel 2.4 and above, Solaris 8/9 both 32 & 64 bit, AIX, OSX
Site	www.guidancesoftware.com

Table 6.1: EnCase Software Detail

AccessData Forensic ToolKit (FTK)

FTK is another popular and widely used computer forensic tool which offers a complete suite for performing forensics examinations of computer systems. In addition to the FTK Imager, password recovery, cracking encryption and other services, FTK is also famous for recovering deleted files and file slack analysis.

EaseUS Data Recovery Wizard

Data Recovery Wizard is complex data recovery software developed for non-destructive data recovery from HDD, SSD and other storage devices. It solves all data loss problems - recover files emptied from Recycle Bin, or lost due to software crash, formatted or damaged hard drive, virus attack, lost partition and other unknown reasons. It recovers data from formatted partitions with the original file names and storage paths. Data Recovery wizard consists of four editions, Free, Standard, professional and Professional Unlimited edition.

Publisher	Chengdu Yiwo Tech Development
File Size	5.33MB
Latest Version	5.5.1 Full version
License	Free Edition (Recover 1 GB for FREE)
Platform	Windows 2000, XP, 2003, vista, 2008, windows 7
Site	http://www.easeus.com/datarecoverywizard/download.htm

Table 6.2: EaseUS Data Recovery Wizard Software Detail

PC Inspector Smart Recovery

PC Inspector Smart Recovery is a Freeware data recovery program for Flash Card, Smart Media, SONY Memory Stick, IBM Micro Drive, Multimedia Card, Secure Digital Card (SD) or any other data device for digital cameras. It enables you to recover accidentally deleted or formatted pictures, videos or sound files from the selected media. The program also offers a mode that enables you to check the media for errors.

Publisher	CONVAR DEUTSCHLAND GmbH
File Size	Size 6233 Kb
Latest Version	3.0
License	Freeware
Platform	Windows 98/ME/2000/XP/vista
Site	http://www.snapfiles.com/get/smartrecovery.html

Table 6.3: PC Inspector Smart Recovery Software Detail

EPOS FlashExtractor

EPOS FlashExtractor is a professional solution for recovering data from storage devices (USB Flash, memory cards, SSD) that are based on NAND Flash memory type. The tool

allows you to recover data in all cases when you cannot access the flash drive through its external interface i.e. in case of physical defect of a controller, a drive locked with password, etc.

Flash Doctor

The flash Doctor is professional tool designed for recovering data from damaged devices (both physical and logical problems). It supports all NAND-based flash storage devices (SD, SM, MMC, XD, USB Pen Drive, Memory stick, Compact Flash etc.). The tool is capable of recovering data from accidental file deletion or format, file system corruption and microcontroller firmware corruption.

Chapter 7

Experiment

This section presents experimental results and a discussion of the results, for the problems posed by new technologies of solid state drive. The experimental tests deals with the two most widely known solutions for SSD performance degradation called TRIM command and Garbage Collection (GC). Both help to keep up the high performance of SSDs by erasing invalid or deleted blocks before a *write* operation performed. However, both this elegant solutions draw a huge threat to the digital forensics existence. When they do an erasing process any data resided in the blocks will gone forever. The only difference between these two techniques is the time they start to execute. TRIM command is an operating system dependent triggered by the file system when a *delete* or *format* operations happened. Whereas the integrated internal garbage collection basically runs in the background independent of the operating system and starts wiping data when the operating system is idle and number of free blocks are below some threshold.

On both experimentations three different SSDs and one conventional hard disk drive have tested. Goals, methods and results of each experiment have discussed in detail for each chosen case in this paper.

7.1 Experiment 1

"*Does TRIM* ruins *forensic evidence in Solid State Drives (SSD)*?"

Goal of Experiment

It's quite clear that the data in the invalid blocks or deleted pages are of interest to the forensic analyst. However these evidences can be altered or destroyed permanently by

some of the newly introduced features of SSDs. The goal of this case goes to verify and figure out the truth that a TRIM command purges forensic evidence in SSD.

Method of Experiment

A new recently bought three different SSDs have filled with a JPEG image files and each drive formatted immediately after storing the files. Finally, a data recovery tool used to get back the deleted data.

Experiment Details

The experiment was carried out on three different SSDs each installed windows 7 and containing a partition drive (D) of size 3.90GB, with the following testing environment:

1. A Dell brand PC 4GB RAM, Intel core 2 quad CPU 2.5 GHZ and 64-bit system type.
2. Crucial M4 SSD 64GB size (D: 3.9GB) , 2.5" form factor, SATA interface, windows 7 installed
3. Kingston SSDnow V 100 64GB size (D:3.9GB), 2.5" form factor, SATA interface, windows 7 installed
4. Samsung 470 series, 64GB size (D: 3.9GB) , 2.5" form factor, SATA interface, windows 7 installed

Before the main experiment was conducted, few settings have changed to make the TRIM command functioning properly. By default windows 7 enables a TRIM command, but that does not make it work completely. Measuring periodically the read and write speed score was a must to guarantee that a TRIM is correctly functioning (i.e. if the storage drivers are passing the command on to the storage controller IC in the SSD) using tools like ATTO disk benchmark. System restore was also turned off as it degrades SSD performance and impedes TRIM from working properly.

Each SSD was connected to the PC at different time for testing. During this time all the drives were hooked up with a SATA interface (AHCI mode). After putting things on the right track, the partitioned drive of each SSD filled with 1,522 JPEG image files of 3.90 GB total size (a 2.59 MB JPEG was duplicated). Following this, each filled partition

drive formatted with “quick format” and “NTFS” options chosen. Immediately we measured each drive read and write speed score with ATTO (to make sure if TRIM is working properly), in which both *Crucial M4* and *Samsung 470 series* SSDs have shown the maximum write score with consistency, whereas the *Kingston SSDnow* has failed to show the same consistency of write speed score. Finally, the PC was rebooted and after 5 minute delay from login time a data recovery process performed with “PC Inspector smart recovery” tool.

Experiment Results and Discussion

The result was similar to the theory of TRIM mentioned above in this paper, the TRIM command purged all the files deleted from the partitions during formatting, except for *Kingston SSDnow V 100* which was fluctuating in *writing* speed score (Though the Kingston vendors claimed it supports TRIM command). SSDs featuring TRIM with windows 7 unlike conventional HDD, they actually delete the data completely not even any remnants are left. Besides improving computer performance the TRIM command is obviously anti-forensic which suites and encourages criminals to pursue their e-crime activities. The results in table [7.2] below shows the recovered number of files from TRIM testing experiments in SSDs.

SSD Type	**# of files stored**	**# of files recovered**	**Time consumed for recovery**
Crucial M4 (TRIM)	1522	0	1:15 hr
Samsung 470 series (TRIM)	1522	0	1:16 hr
Kingston SSDnow V 100	1522	1383	1:20 hr

Table 7.1: TRIM command experimentation results

The above results show that TRIM can sanitize the entire drive in seconds and makes forensics data recovery worse. The probability of recovering files is almost 0%, giving no chance at all.

Similar experimentation with DELETE operation:

All the SSDs once again tested to verify if a TRIM command can also be triggered and purged data after "DELETE" operation. The test was only performed on C drive by deleting thousands of JPEG image files mentioned in the above experimentation. Similarly the results bear a striking resemblance with "FORMAT" operation experiment in the above. 0% data recovery chance with both *Crucial M4* and *Samsung 470 series*, whereas *Kingston SSDnow* does not purged data and everything was able to recover, still failed to show the same result like the other two SSDs.

7.2 Experiment 2

"*Does a Garbage Collection ruin forensic evidence in Solid State Drives (SSD)*?"

Goal of Experiment

This experiment determined if garbage collection purges forensic evidence from SSD when it starts running during the idle state of the operating system.

Method of Experiment

The same method as that used in experiment 1 has applied, except the addition of a conventional HDD for result comparison. All SSDs and a wiped HDD filled with a JPEG image files. To realize the effect of garbage collection and if data from unallocated space can be fully recovered, we formatted each SSD and HDD right after storing the files. Finally, a data recovery tool used to recover the files deleted during the formatting process.

Experiment Details

The experiment was carried out on both storage devices SSD and HDD with the following testing environment:

1. A Dell brand PC with Windows 7 OS, 4GB RAM, Intel core 2 quad CPU 2.5 GHZ and 64-bit system type.
2. Crucial M4 SSD 64GB size, 2.5” form factor, SATA interface
3. Kingston SSDnow V 100 64GB size, 2.5” form factor, SATA interface
4. Samsung 470 series, 64GB size, 2.5” form factor, SATA interface
5. Western Digital HDD, 250 GB (only 64.8 GB partition was used), SATA interface

The experimental test for each drive went by connecting them to the PC via SATA interface at different times. We have set up all SSDs with all the free space filled up by 17,267 JPEG image files of 58 GB (a 3.44 MB size JPEG image was duplicated) with 2.72 KB free space left. A TRIM command also disabled in Command Prompt window to stop from being intervening.

A traditional HDD was another part of the experiment to help out in comparing the result with SSD. A 64.8 GB partition of the HDD was first formatted and wiped using Active@ kill disk data wiping tool ahead of storing the targeted JPEG images. The default wiping method “one pass zeros (1 pass)” used during the process. After making the drive ready, a 64.7GB size of 4,739 JPEG image files (a 14 MB size JPEG was duplicated) filled the HDD.

Immediately after each drive has filled up to its capacity a formatting process has taken place with “quick format” and “NTFS” options chosen and PC restarted. Finally, a standard SSD and HDD data recovery procedure was followed to recover the files with the help of “PC inspector smart recovery” data recovery tool.

Experiment Results and Discussion

After full tests have conducted, based on the procedures mentioned in the methods of experiment a summary of the experimental results have discussed here. Unlike to the expectation none of the SSDs was having an internal background garbage collection. Unfortunately understanding and dealing with the effect of garbage collection was unsuccessful. Like the conventional HDD, the recovered files from all three SSDs were

exactly the same. Table [7.1] below shows the results of the experiment: number of files recovered from each type of drive. One thing is true here, any SSD which does not support internal garbage collection is far from purging data and any file can be recovered like HDD.

Drive Type	# of files stored	# of files recovered
HDD	4,739	4,722
Crucial M4 SSD	17,627	17,625
Kingston SSDnow V 100	17,627	17,625
Samsung 470 series	17,627	17,625

Table 7.2: Garbage collection experiment results

The above results clearly do not show anything about the garbage collection effect on deleted or invalid data, and this is due to lack of supporting the GC inside the SSD controllers. Not all SSDs have an internal background/idle garbage collection to keep up high-speed of writing performance, however they could (not) support another similar sustaining performance solution called TRIM.

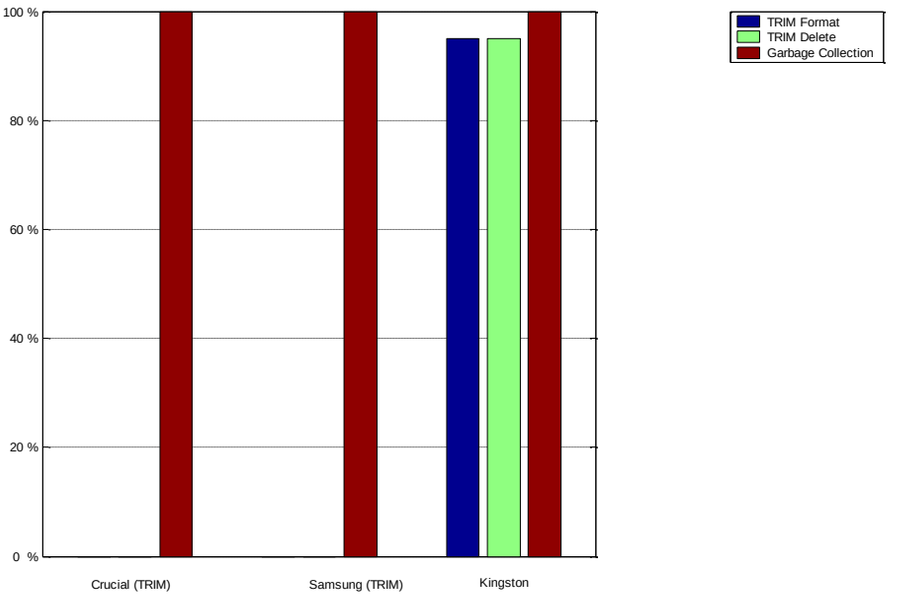


Figure 7.1: Percentage of data recovered

The above chart conveys percentage of data recovered for the different experimental tests conducted on the three SSDs. It simply shows the overall experimental results.

Chapter 8

Conclusion

For a long time the percentage of users having SSD drive were very limited due to the high price, however recently the price of SSD has begun to fall down and ready to take over the conventional HDD which increases the number of SSD users. While TRIM functionality is also becoming more prevalent and widely supported by many operating systems so that computer performance can be improved. The inevitable emergence of the TRIM-enabled Solid state drives with a TRIM supporting operating system (like windows 7) dramatically announces the demise of digital forensics golden age and aggravates the cyber crimes on SSD. As the results of TRIM experimentation in this paper reveal SSD data recovery with existing tools seems almost impossible, vendors should analyze all the effects of TRIM command and make reduce its burden on forensics. Engineers too, should not consider SSD like HDD any more, a new advanced SSD data recovery might be necessary to overcome the problem and to come up with lasting solutions.

BIBLIOGRAPHY

[1] Esther Spanjer. “Flash Management – Why and How? A detailed overview of flash management techniques”

[2] William F. Heybruck, Senior Applications Engineer: An Introduction to FAT 16/FAT 32 File Systems

[3] NUMONYX. TN-29-63 Technical Note. “Wear Leveling in NAND flash memory”.

[4] NUMONYX. TN-29-63 Technical Note. “Error correction code in single level cell NAND flash memory”

[5] NUMONYX. TN-29-59 Technical Note. “Bad block management in NAND flash memory”

[6] NUMONYX. AN1821 Application note. “Garbage collection in NAND flash memory”

[7] Esther Spanjer. “Flash Management – Why and How? A detailed overview of flash management techniques,”

[8] Graeme B. Bell, Richard Boddington. “Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?” Journal of Digital Forensics, Security and Law 5, 2011 Available: http://www.jdfsl.org/subscriptions/JDFSL-V5N3-Bell.pdf

[9] Lachlan Roy. “Hard drive of the future - your guide to solid state drives”

[10] Michael Wei, Laura M. Grupp, Frederick E. Spada, Steven Swanson. “Reliably Erasing Data from Flash-Based Solid State Drives,” University of California, San Diego. Available:http://cseweb.ucsd.edu/users/m3wei/assets/pdf/FMS-2010-Secure-Erase.pdf

[11] Madihah Mohd. “An overview of disk imaging tool in computer forensics,” SANS Institute.

[12] Mark E. Donaldson. “The Master Boot Record, The Partition Table & The boot Sector”

[13] Brown, C. L. T. “Computer evidence: Collection and preservation, ” 2nd ed. Boston, MA: Course Technology.

[14] National Institute of Justice (NIJ). “Digital evidence in the courtroom: A guide for law enforcement and prosecutors (NIJ Special Report NCJ 211314),” (2007, January) Available: http://www.ncjrs.gov/pdffiles1/nij/211314.pdf

[15] Casey, E. “Digital evidence and computer crime: Forensics science, computers and the Internet, 3rd ed.” Amsterdam, The Netherlands: Elsevier Academic Press.

[16] SAMSUNG. 2007. XSR 1.5. “Bad Block Management,” Available: http://www.samsung.com/global/business/semiconductor/products/flash/downloads/xsr_v15_badblockmgmt_application_note.pdf

[17] Anand Lal Shimpi. “The SSD Anthology Understanding SSDs and New Drives”

[18] Kornblum, J. D. “The Linux Kernel and the Forensic Acquisition of Hard Discs with an Odd Number of Sectors,” International Journal of Digital Evidence.

[19] Matthew Braid. “Collecting Electronic Evidence After a System Compromise,” SANS security.

[20] Wayne Jansen. “Rick Ayers Guidelines on Cell Phone Forensics”

[21] M.F. Breeuwsma. “Forensic imaging of imbedded system using JTAG (boundary-scan)”

[22] Alessandro Distefano, Gianluigi Me. “An overall assessment of Mobile Internal Acquisition Tool”

[23] Derek Bem, Ewa Huebner. “Analysis of USB Flash Drives in a Virtual Environment”

[24] http://www.techopedia.com/definition/992/bad-sector

[25] John Patzakis. “New Accounting Reform Laws Push For Technology-Based Document Retention Practices,”

[26] Christopher J. Antonellis. “Solid state Disks and Computer Forensics”

[27] Casey, E. “Digital Evidence and Computer Crime, Second Edition,” 2004

[28] ProtoLogic Privacy Controls, “Uncovering the Ghost in the Machine”, Product White Paper, 2006.

[29] NIST. “Hardware Write Blocker Device (HWB) Specification,” National Institute of Standards and Technology. May 19, 2004

[30] Scott Moulton. “ Solid State Drives & How SSD and USB Flash Work”