

Sentient Storage:

Do SSDs have a mind of their own?

Tom Kopchak

Table of Contents

Abstract	8
Introduction	9
Purpose	10
Literature Review	12
Challenges	20
Overview of Storage Architecture	22
Magnetic Hard Drives	22
Solid State Drives	24
Flash Memory	24
Controllers	27
Evidence Collection	30
Methodology	31
Materials	34
Control: Seagate ST98013ASG, 80GB, 7200 RPM	34
SSD #1: Intel SSD 320, 120GB	35
SSD #2: Crucial M4, 128GB	36

SSD #3: Patriot Pyro SE, 128GB	36
SSD #4: Samsung 830, 128GB.....	37
SSD #5: OCZ Agility 3, 60GB.....	38
SSD #6: SuperTalent MasterDrive, 64GB.....	39
Forensics Lab Configuration	40
Experiments	42
Test 1, Part 1 – Text File Deletion.....	42
Procedure	42
Results	42
Test 1, Part 2 – Text File Deletion – Quick Format	43
Procedure.....	43
Results	43
Test 2	43
Test 3, Part 1 – Single JPG File	44
Procedure.....	44
Results	44
Test 3, Part 2 – Single JPG File, Quick Format	45
Procedure.....	45

Results	45
Test 4, Part 1 – Single JPG File, TRIM disabled	46
Procedure	46
Results	46
Test 4, Part 2 – Single JPG File, TRIM Disabled, Quick Format	47
Procedure	47
Results	47
Test 5, Part 1 – Single JPG File, TRIM enabled, SSD connected via USB	48
Procedure	48
Results	48
Test 5, Part 2 – Single JPG File, TRIM enabled, SSD connected via USB, Quick Format	49
Procedure	49
Results	49
Test 6, Part 1 – Single JPG File, Windows XP	50
Procedure	50
Results	50
Test 6, Part 2 – Single JPG File, Windows XP, quick format	51
Procedure	51

Results	51
Test 7, Part 1 – Two JPG Files, one deleted	52
Procedure	52
Results	52
Test 7, Part 2 – Two JPG Files, one deleted, quick format	53
Procedure	53
Results	53
Test 8, Part 1 – Two JPG Files, 8 MB partition, one deleted	54
Procedure	54
Results	54
Test 8, Part 2 – Two JPG Files, 8 MB partition, one deleted, quick format	55
Procedure	55
Results	55
Test 9, Part 1 – Two identical JPG Files, one deleted	56
Procedure	56
Results	56
Test 9, Part 2 – Two identical JPG Files, one deleted, quick format	57
Procedure	57

Results	57
Test 10, Part 1 – 60 numbered JPG Files, one deleted per minute over an hour.....	58
Procedure.....	58
Results	58
Test 10, Part 2 – 60 numbered JPG Files, one deleted per minute over an hour, quick format	60
Procedure.....	60
Results	60
Test 11, Part 1 – 60 numbered JPG Files, one deleted per minute over an hour, TRIM Disabled ..	61
Procedure.....	61
Results	61
Test 11, Part 2 – 60 numbered JPG Files, one deleted per minute over an hour, TRIM Disabled, quick format	62
Procedure.....	62
Results	62
Test 12, Part 1 – 60 numbered JPG Files, one deleted per minute over an hour, USB connection	63
Procedure.....	63
Results	63

Test 12, Part 2 – 60 numbered JPG Files, one deleted per minute over an hour, USB connection,
quick format 64

Procedure..... 64

Results 64

Observations and Analysis 65

Forensic Implications 70

Conclusions 71

Future Work 72

Works Cited..... 73

Abstract

This research represents the most comprehensive study of the implications of solid state drives on the recoverability of deleted files to date. These drives have the potential to fundamentally change the digital forensics industry due to the differences in how they respond to the deletion of files. The goal of this study was to demonstrate and quantify these differences. A pool of six solid state drives was subjected to eleven two-part tests, each specifically designed to incrementally demonstrate variations in the performance of SSDs when compared to a control hard drive. Each experiment began with a deleted file recovery test, followed by a quick format file recovery test. A wide variety of differences between the solid state drives were observed, including some drives that behaved nearly identically to the control drive and others where the likelihood of recovering deleted data was significantly reduced. By far, the most influential factor in determining the recoverability of data during a given test was the state of the TRIM command, which is responsible for notifying the drive controller of a deletion event and expediting the garbage collection process for erasing the freed flash cells and returning them to the pool of available storage. By better understanding the controller and firmware functionality of a solid state drive as well as the presence or absence of the TRIM command, forensics investigators can apply the results of this research to determine the likelihood of successful deleted file recovery from an evidence bearing solid state drive.

Introduction

The proliferation of computer technology and the Internet has significantly impacted society. As computers became more commonplace in homes and businesses, they assumed an ever-expanding role in the lives of their users. Everyday activities such as banking, shopping, and communication, previously conducted between individuals in a public area, could now be completed without leaving one's home or office. While this increased convenience for end users, criminals also adopted these new technologies and developed strategies to perpetrate their crimes. This shift in technology required investigators to design new approaches for interrogating computers and their associated storage devices to obtain evidence.

Currently, there exists a wide variety of methods employed for storing user information and data. Traditional computer hard drives use magnetic media to store information. Forensic investigators have well-defined protocols and procedures for managing the capture of evidence from these types of drives (Bell, 3). Many of these processes take advantage of the physical characteristics of the storage media, as well as the methods by which various operating systems utilize the media. For example, since many popular operating systems do not purge data once the user issues a command to delete a file, investigators are able to either partially or fully recover the contents of a deleted file. This data may be available since the file is often simply marked as deleted, and the space associated with the file is returned to the pool of available storage. The slow speed and mechanical latency of magnetic storage is a technological limitation that makes purging a file's contents upon deletion impractical (Gutmann). By knowing and understanding this characteristic, it is possible, and generally a fairly straightforward process, to recover deleted files from a magnetic hard drive. This basic property of magnetic storage has become a cornerstone of many forensic investigations (Bell, 4).

Flash-based solid state storage is rapidly becoming a popular replacement option for traditional hard drives, especially in high-performance servers for caching, as well as in mobile devices such as laptops. These drives offer many clear advantages over their mechanical counterparts, such as significantly quicker seek times and throughput rates, better vibration and shock tolerance, and increased reliability, which have led to their increased deployment (Leventhal, 26). New technologies typically offer new challenges, especially in the area of data security, and solid state drives are no exception (Bell, 4-5). With the proliferation of this type of media, concerns have mounted regarding the impact of this technology on the data storage lifecycle. Minimal research has been conducted regarding the impact of this new storage technology on existing forensic practices (King, S111). Can simply creating a disk image of a solid state drive irrevocably destroy crucial evidence? Can deleted files be recovered employing the same techniques used on traditional hard drives? These are just a few of the many questions that must be answered.

Forensic investigators often rely on trace amounts of evidence to make or break a case. A single mistake during the data acquisition or recovery portions of an investigation can be the difference between a successful conviction or a failed case. Investigative procedures should never be developed ad-hoc, due to the risk of tampering with or destroying evidence. When working with any new storage technology, it is critical to understand the characteristics of the underlying physical storage, and how data is managed and deleted by the drive or device (King, S112). By studying solid state drive forensics procedures now, forensics professionals will be better prepared to work with these drives in the future.

Purpose

This project will seek to comprehensively study the impact of solid state drives on the forensics data acquisition and investigation process. Specifically, this research will focus on the impact of these drives on current forensics practices involving deleted file recovery. Ultimately, this research will seek to

answer the question of how these drives behave when compared to traditional hard drives, and if traditional forensics approaches are sufficient for recovering deleted files from a solid state drive or if they must be modified. The goal of this research is to contribute to the forensics community a more coherent understanding of the implications of working with these storage devices that may be referenced when collecting and investigating evidence stored on a solid state drive.

Literature Review

Overall, minimal research has been conducted regarding the impact of solid state drives on existing forensics procedures, but this research is certainly an area of keen interest within the digital forensics community. Preliminary work has shown that this type of storage behaves in a significantly different manner than existing magnetic storage technologies, and presents unique challenges with respect to data retention and forensic investigation (Bell, 11). This certainly appears to be an area where there is much more to learn, to research, and to better understand. This literature review will focus on the general understandings and concepts related to flash memory forensics, including the research that has been done previously in this area. This review will also serve as a delineation of the basic hardware and fundamental operation of solid state disks, and why the architecture of these drives pose such a challenge to traditional forensics techniques. This information will provide a basis for a better understanding of the physical properties of flash media, and will also provide a snapshot of current research and issues to help guide the direction and focus of this research.

The key component of a solid state disk is its flash memory. This flash-based form of storage exhibits different properties when compared to traditional magnetic storage mediums. To understand solid state drives, one must first understand the underlying flash memory. The article “Characterizing Flash Memory: Anomalies, Observations, and Applications”, written by researchers at the Center for Magnetic Recording Research at the University of California, detailed these parameters (Grupp, et al.). The article attempted to look past the typical manufacturer specifications for flash memory and reveal true facts about the media – including the performance, power, and reliability of flash devices. This article provided information about the physical structure of flash memory, and how data is stored on the physical medium. One of the more interesting aspects detailed was the analysis of flash memory life. Flash memory is made up of a variety of cells, with a finite amount of information stored in each cell (25). After extended periods of repeated use and reuse, individual cells wear out and fail. When

failures occur, the entire affected block of storage is taken out of service. This creates the potential for residual data to remain within a flash based solid state disk. In addition to describing this process, actual tests were performed using a variety of flash memory chips, to observe the behavior of the flash when stressed and repeatedly used (28-29). The study indicated that certain types of use patterns are more likely to cause errors to occur in flash memory devices, and those errors increase dramatically when approaching the manufacturer specified lifecycle for the chip (33). By better understanding the physics and lifecycle of flash memory, it is possible to look for areas where sensitive data may be left on a solid state disk. This article provides insight into the operation of these drives, which in turn can be used towards better understanding the forensics characteristics of these types of devices.

For the purpose of this research, it is important to explore the current condition of the solid state drive industry, and examine how these devices are used in enterprise environments. Leventhal explores the applications of flash technology in the computer storage industry. As the cost per gigabyte of flash storage has declined nearly exponentially over the past several years, the popularity of this format has increased substantially (Leventhal, 25). Benefits of SSDs include significantly reduced latencies when compared to even the fastest rotating magnetic disks, while consuming less power and dissipating less heat (30). Flash memory is still expensive in terms of cost per gigabyte, so currently, it is still not competitive when compared to magnetic disks for bulk data storage, although this cost disparity continues to diminish as time goes on. For caching and logging, however, flash based storage offers many advantages due to its faster performance and lower read and write latencies (28). One of the facts mentioned in this article was that flash memory devices typically contain a substantial amount of “unadvertised” storage capacity, held in reserve, which is invisible to the operating system (27). This is done to increase the lifecycle of the drive as individual flash memory cells fail and become unusable. Since a fair amount of storage is unavailable at any given point in time, it seems that this could provide a source of residual data on the drive, which might prove to be a useful component of a forensic

investigation. This may provide a productive avenue to investigate further as part of an experiment or additional research.

To gain a further understanding of the technical limitations of solid state drives, an article written by Mark Moshayedi and Patrick Wilkison was referenced. This piece discussed some of the advantages and limitations of current SSD technology. One surprising restriction is that the cells in many types of flash memory must be erased before being written, and this erase process is very slow (Moshayedi, 36). To further complicate matters, only an entire block can be erased at a time (35). In other words, data is marked for deletion, but a block cannot be reclaimed until all the data on that block is either marked for deletion or moved to another block (36). This would appear to be an area that might be a substantial risk for leaving residual data behind. SSDs also need to manage bad blocks that develop over time, and replace them with good blocks (37-38). These operations require a substantial amount of background processing, which can become a bottleneck. Additionally, it is possible that this background processing might manipulate data or delete potential evidence, perhaps without receiving any commands from the operating system at all. Although this article did not cite any experiment, it provided an excellent overview of the challenges and benefits on the SSD landscape. Understanding the issues surrounding erasing and reclaiming blocks of storage will be an important area for this research.

Several of the aforementioned articles discussed the use of additional storage within a solid state disk for the purposes of wear leveling. To better understand this concept, the article “Design and Implementation of an Efficient Wear-Leveling Algorithm for Solid-State-Disk Microcontrollers” was referenced. This article explored several of the issues mentioned in the previous article, such as the management of block erasures and movement of data from one area on the drive to another (Chang, 6:2-6:5). Because SSDs do not have the seek time associated with mechanical drives, there is minimal performance penalty for random versus sequential reads. Consequently, this results in data being

scattered widely across a drive, along with many blocks of data that may be marked “invalid” but still contain information needing or waiting to be erased (6:6, 6:8). This erasure procedure is handled differently, depending on the controller chip and drive firmware. With so many variables, including drive autonomy, investigating the wear-leveling algorithms used in solid-state drives could provide an interesting pathway for forensics investigations.

To better understand how a representation of an SSD can be presented to the operating system, the article “Modeling and Simulating Flash based Solid-State Disks for Operating Systems” was chosen. Currently, virtually all research concerning SSDs is directly performed on the physical hardware itself. There has been little research into how these drives might be simulated, to allow them to be better tested, dissected, and studied. Currently, the majority of the logic performed within an SSD drive is done by the drive’s controller itself (Maghraoui, 17). In many cases, the operating system is not aware of any major differences between a magnetic hard drive and a flash based solid state disk. If an operating system was more disk type aware, there potentially could be many benefits, especially with respect to SSDs, their block management, and data security. However, this could have potentially disastrous impacts on the viability of current forensics recovery methods and investigation procedures. Being able to simulate the behavior of an SSD and tailoring this ability to develop improvements to operating system management routines for these types of devices would be a major breakthrough. Unfortunately, the simulator developed as part of this research was limited to a kernel extension for the AIX operating system, which makes it less accessible to many users. At the present time, physical hardware will still need to be the primary test bed.

Perhaps one of the most commonly cited papers with respect to data sanitization and computer forensics was published by Peter Gutmann and Colin Plumb in 1996: “Secure Deletion of Data from Magnetic and Solid-State Memory”. This paper described a secure-deletion algorithm utilizing a series

of 35 patterns to be written over the data to be erased. The majority of the patterns written using this method were not random, but instead targeted various encoding methods used by different storage technologies prevalent when the research was conducted (Gutmann, np). Since patterns are targeted towards certain encoding techniques, if the specific encoding is known, only the patterns designed for a certain method are necessary (np). Using a pattern designed for a different encoding method is akin to writing random data on the drive. The implementation of the ATA secure erase command has reduced the need for the Gutmann method, especially when an entire drive is being sanitized. This paper is notable from a historical standpoint, since it is commonly referenced when considering forensic data recovery. It is interesting to note that the research in this paper is often misunderstood, to the point where Gutmann has added an appendix to the research to address these misunderstandings and misconceptions, and emphasize that the erasure patterns described are not applicable to most modern storage technologies (np).

In 2010, a research report involving solid state drives and forensic recovery was published. The article, "Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?" detailed several experiments where the forensic properties of a traditional magnetic hard drive were compared to a solid state drive. This study demonstrated that a solid state drive is fundamentally different from a magnetic storage drive, and it is possible for the drive controller in one of these devices to manipulate data on its own (Bell, 7-10). Furthermore, the presence of a write-blocker did not seem to prevent irrecoverable data loss from occurring (10-11). This article is groundbreaking, since it is the first to call into question traditional forensic practices, and alert investigators to take caution when working with these drives. However, the scope of the research conducted, as well as the sample size used, was quite minimal. Additionally, the study only focused on drives that had been quick-formatted, as opposed to considering other use patterns or deletion procedures. As a result, the reader of this article is left with many questions, and few definitive answers.

In 2011, researchers at Carnegie Mellon University performed an empirical analysis of solid state disk data retention. This study sought to determine the impact of various operating systems on the operation of the garbage collection algorithms of different solid state drives, with particular attention paid to the TRIM command. This ATA command serves to manage blocks of storage that contain data no longer used by the operating system (King, S112-S113). Since flash-based memory cells must be completely erased to be re-used, the cleanup of these cells can cause a major loss of forensically-interesting data. The researchers concluded that there is a significant amount of variability concerning combinations of operating systems, format types, and solid state drive controllers (S116). In some cases, deleted data is able to be recovered or nearly completely recovered. However, in other cases, most notably those involving operating systems and drives supporting TRIM, data is completely sanitized within minutes of deletion (S116-S117). Once again, there are a substantial number of variables involved in determining cases where data is recoverable and when it is lost. However, a fundamental fact remains – solid state drives demand different approaches for forensic investigation, and the technology and challenges involved are not yet well understood.

Wei, et al, presented an article involving the reliable erasure of data from flash-based solid state drives. This article explored the effectiveness of traditional hard-drive sanitization procedures, including the implementation of the ATA secure-erase command, as well as the use of data wiping applications. This study found that the entire SSD can generally be successfully sanitized using existing techniques (Wei, 12). However, single-file sanitization techniques were consistently ineffective at permanently removing data from the drive (7). This research also uncovered a notable variability in the implementation of ATA commands for secure erase. In several cases, these commands were found to be completely ineffective at actually erasing the drive, while others worked just as effectively as those implemented on magnetic hard drive counterparts (4-5). This article also looked beyond the drive controller and examined the flash memory chips themselves. When bypassing the flash translation layer

(FTL), it is possible to examine the contents of individual memory cells. Data may exist multiple times within different flash chips at any given point in time, and this data can be recovered by physically exploring the flash memory itself (10). This fact could prove extremely helpful to forensics investigators, who might be searching for fragments of evidence. Since forensics and data sanitization (anti-forensics) go hand-in-hand, this article provided very useful background information regarding the possibility of recovering data from solid state drives and individual flash memory chips.

A 2012 paper by Yuri Gubanov and Oleg Afonin entitled “Why SSD Drives Destroy Court Evidence, and What Can Be Done About It” addresses many early concerns about the impact of solid state drives on the digital forensics process. The central theme of this paper is the contamination of evidence, or self-corrosion of evidence by the solid state drive’s controller. Unlike traditional solid state drives, there are many factors that influence the likelihood of data recoverability. The presence and support of the TRIM command along with the drive usage and operating system configuration will often have a significant impact on forensics efforts. This paper also addressed the impact of full-disk encryption on the recoverability of data. Interestingly, a fully-encrypted SSD is more likely to be able to have data recovered than a non-encrypted one, provided the investigator has access to the decryption key or decryption password for the drive. This is due to the fact that solid state drive controllers are not able to optimize data and TRIM is typically non-functional on a fully encrypted disk.

Finally, a 2013 paper by Gabriele Bonetti, Marco Viglione, Alessandro Frossi, Federico Maggi, Stefano Zanero, and Politecnico di Milano entitled “A Comprehensive Black-box Methodology for Testing the Forensic Characteristics of Solid-state Drives” builds on much of the previous work conducted by researchers such as Bell. These researchers recognize the practicality and potential value of performing preliminary forensic investigation techniques through the SSD’s controller (termed black-box analysis) versus a more complex and failure-prone approach of bypassing the flash translation layer

and accessing the flash memory chips directly (termed white-box analysis). This research utilized similar methodologies as Bell and in some case did not successfully replicate the results previously seen in this research, even using identical drives, controllers, and firmware. Based on an analysis of three popular SSD models and the presence of TRIM, garbage collection, wear leveling, and compression, these researchers seek to establish a framework for gauging the likelihood for successful file recovery on a given SSD. Based on the results of their experimentation, researchers assigned the pool of test SSDs a ranking from Complete Wiping (no data can be recovered) to Platter Disk equivalent (the SSD behavior matches that of a traditional hard drive). The research conducted in this study will support many of the conclusions first seen in this paper and can use this framework to communicate the results.

Solid state drive technology is still in its infancy. There is much more that needs to be researched, developed, and understood regarding these devices. As with the implementation of any new equipment, there are inherent changes that must be evaluated in light of existing technologies and processes. Solid state drives are a very different technology than magnetic storage media. These drives use dissimilar algorithms for storing data than traditional disks, contain surplus memory capacity for accounting and replacing failed memory blocks, and must manage deletion operations, which can only be performed on an entire block at a time. Consequently, this revolutionary change in storage architecture will demand a corresponding revolution in the computer forensics field. Understanding the current research in this area will help direct the scope of testing and the issues that must be explored and addressed, and guide the focus of this study to modify and improve existing forensics techniques to successfully manage this new technology.

Challenges

This research will focus on and attempt to reconcile several of the challenges faced by forensics investigators when working with solid state drives. One area of particular concern is the autonomy of solid state drives, and the potential for these devices to modify or delete evidence without any external input by the investigator. Wear leveling and garbage-collection algorithms may result in data being moved between or removed from physical flash memory locations. Since these operations are managed by the onboard SSD controller, deletions may occur independently of any commands issued by a host device controller of the motherboard or operating system, regardless of whether or not a write blocker is in place (Bell, 10-11). It is entirely possible that modification could occur while the device is merely powered on, yet disconnected from any computer or data connection. This research will investigate the autonomy of SSDs upon the deletion of a file, and attempt to isolate the garbage collection operations in order to determine which events are more likely to trigger evidence loss.

The management of writes and rewrites within a solid state drive presents a new challenge to forensic investigators. The physical requirement that flash cells be completely erased before being rewritten is a significant change from the operation of magnetic hard drives, where data could be rewritten without any preparation or erasure of the media. Since the process of refreshing a flash cell is time-intensive, it is generally best performed when the drive is idle as opposed to when data is being overwritten. The performance of a solid state drive can be heavily dependent on a pool of available flash which is pre-refreshed and ready for use (Leventhal, 27). Because these drives are marketed as performance boosting devices, manufacturers are under constant pressure to increase the speed of their drives. Current forensics work relies on recovering data artifacts produced during the magnetic recording process. These processes are not designed with the architecture of an SSD in mind. Current forensic practices must be adapted in order to yield results with this new technology.

Solid state drives also present a marked increase in the number of variables that must be considered when conducting an investigation. When working with traditional hardware, the data recovery process is mainly dependent on the operating system and the file system of the evidence drive. For example, evidence from a Windows machine using a magnetic hard drive can be captured and processed using similar procedures in nearly all investigations, whether or not the drive is a desktop or laptop form factor, contains a certain storage capacity, or is made by a specific manufacturer. However, SSDs can behave differently depending on the operating system in use (King, S114). The properties of individual drives are significantly influenced by the firmware of the onboard SSD controller. Different manufacturers produce drives with different controllers or controller firmware versions, and competing models of SSDs contain appropriate quantities of spare flash memory capacity in accordance with the manufacturer's expected use model for the drive. Each of these differences poses a variable and potential challenge to forensic investigators. Although it will be impossible to test every possible combination that might appear in an actual forensic investigation, this research will attempt to demystify at least some of these variables, in order to provide insight for future work and investigations.

Overview of Storage Architecture

Solid state drives and magnetic hard drives are fundamentally different hardware. Both of these storage platforms utilize different material properties to store data on their respective physical media. Due to these distinct differences, magnetic hard drives and solid state drives cannot be considered to be the same type of hardware, even though both of these devices accomplish the same goal.

Both of these drives share a common property – they are designed to be presented as storage to an operating system. Although solid state drives represent a newer, updated storage technology, they are intended to serve as drop-in replacements for traditional hard drives. In general, the operating system should not require any special modification or drivers to utilize either form of storage. Drives are ultimately designed to abstract the physical medium of data storage from the user. However, from an information forensics perspective, understanding the operations that occur behind this layer of abstraction is of critical importance. Although both the operating system and the user may be naïve of the distinction between different types of drives, a forensics investigator must be acutely aware of the unique characteristics of each type of storage device they must interrogate for evidence.

Magnetic Hard Drives

The operation and behavior of traditional hard drives is well understood and documented. The form and function of these drives has been more or less unchanged over the past fifteen to twenty years. Although drive capacities have increased exponentially, fundamentally, the process by which the data itself is stored, encoded, and accessed is essentially unchanged. Because of this, forensics investigators are familiar with handling these types of drives.

The inside of a hard drive contains circular metal discs, which are called platters. The platters are typically made from aluminum, which is coated with a magnetic recording layer. Platters have also been made from glass and ceramic substrates as well. Platters have a fixed capacity, and drives may use

multiple stacked platters in order to increase the overall capacity of the drive. When the drive is in operation, these platters spin rapidly at a constant speed that is governed by a precise motor. Data is recorded in a series of circular tracks, which are written and accessed using a recording head. This drive component is physically analogous to the needle on a phonograph, but it does not operate in a purely sequential manner. Additionally, hard drive tracks consist of concentric circles, as opposed to a spiral groove.

The drive electronics are responsible for moving the head to (seeking) the proper position to align the drive heads over the track containing the data that is to be written or read. The data itself is stored in sectors, which represent the smallest quantity of storage that is usable at a time. Generally, drives use a sector size of 512 bytes, although larger capacity drives are moving to larger sector sizes in response to ever-increasing platter densities. The actual data is read and recorded using a magnetic coil. The drive electronics are able to discern minute differences in the magnetic field of the drive platters, and decode this into a binary representation of the encoded and stored data.

In order to accurately read and store data, a great deal of precision between all these mechanical and physical components is required. Normal drive operation can be negatively impacted by shock, vibration, or dust coming in contact with the head or platters. Any disassembly of a hard drive places the stored data at extreme risk for corruption, due to the tight mechanical tolerances required. Generally, any work involving the platters themselves must be done in a clean-room environment by an engineer familiar with the dangers and processes involved. Depending on the condition of a drive, it is possible to rebuild a damaged or non-functional drive to recover critical data. This is an inherently perilous and expensive process which is only employed when the value of the data involved exceeds the costs and risks associated with such data recovery methods.

Solid State Drives

Solid state drives are designed to serve as drop-in replacements for traditional hard drives.

Although the physical storage media and operation of the device is completely different when compared to a magnetic hard drive, the drive electronics are designed to abstract these properties from the operating system and user as much as possible. From a consumer and end-user perspective, this is a major benefit, since these drives will work with their existing operating systems and hardware without any additional inconvenience or effort. However, since these drives are fundamentally different technologies, techniques for obtaining and managing evidence acquisition from these drives may need to change.

Evaluating solid state drives is a challenging proposition: although the basic data storage properties are constant across product lines, there is a significant degree of variance between the drives themselves. Individual drive components, including the type of flash memory, the controller, the firmware, and the compression and data management algorithms can differ greatly between individual drive models, and even more substantially between manufacturers.

Flash Memory

Flash memory is the physical storage medium in a solid state drive. The structure and physical properties of flash memory cells are responsible for many of the operating characteristics of solid state drives. In many respects, significant engineering accommodations are required for raw flash memory to come together as a single storage device.

Different types of flash memory cells are also available – SLC and MLC. Single-level cell (SLC) flash memory cells store one bit, represented in binary as either a one or a zero. Multi-level cell (MLC) flash memory, on the other hand, offers the ability to store two bits in a single cell. This is accomplished by using multiple levels of charge to simulate the binary values of 00, 01, 10, and 11. Different voltage

levels correspond to different values. Due to the minute differences between the electrical characteristics of these binary values, MLC flash is significantly more sensitive (and potentially less reliable) than SLC flash.

At the physical layer, flash memory is divided into pages. Pages are the smallest addressable unit within a flash memory cell. Pages cannot be overwritten, due to the fact that erasing pages could potentially modify adjacent cells within a block. This is a physical limitation of the storage medium itself. Due to this restriction, only entire blocks are erased at a time.

A completely overwritten flash cell is represented in binary as being filled entirely with 1s. To store data, these 1s are overwritten with 0s. However, once a 1 is changed to a 0, it cannot be changed back without completely erasing the block (resetting it back to 1s) and starting over from scratch. At present, this is an absolute requirement, since the process of erasing a block requires significant quantities of both electrical current and time.

When data is deleted on a solid state drive, the blocks containing the data are marked as invalid. However, since they still contain data, they cannot be immediately reused without first being erased to a clean state and returned to the pool of free blocks. These operating procedures ensure that there is some period of delay from the time data is deleted at the operating system level to when it is actually deleted from the flash memory itself.

Flash memory cells also have a finite usable amount of write cycles. It is not uncommon for individual blocks to fail with age as they are used. Due to the nature of computer I/O operations, certain files are more frequently modified than others. This results in uneven utilization patterns, where some blocks are frequently rewritten and others (such as the locations of core operating system files) sit

stagnant for extended periods of time. The drive controller is responsible for ensuring even wear across the entire solid state drive.

Controllers

The heart of a solid state drive is its controller. This device is responsible for all operational aspects of a solid state drive. The choice of a controller and the implementation of its firmware can have a significant consequence on the overall impact and performance of the drive itself. There are several varieties of competing controller designs which implement features differently. Furthermore, manufacturers frequently customize controller firmware to help distinguish their products. These differences complicate the evaluation of solid state drives, since the characteristics of the hardware has a tendency to vary across manufacturers and even individual models of drives.

By far, the slowest operation in a current-generation solid state drive is the erasure of invalid flash cells to restore them to a refreshed and usable state. Since any valid data must be moved out of a block before it is erased, this becomes an even more time-consuming process. Early generation SSDs showed noticeable degradation of performance as they aged, due to the lack of available unused pages for storing data as it was written. When no free pages are available, the SSD controller must manipulate the existing data, shifting around valid data to other areas of the drive so that entire blocks may be erased. A design goal of more recent SSDs is to prevent this degradation of performance whenever possible. To prevent SSDs from incrementally getting slower as they are used, manufacturers implement a variety of garbage collection techniques to ensure a pool of available pages are continually ready for use.

The time periods where any drive is actively reading or writing data is generally a small component of the overall duration of a drive's lifecycle. This means that the drive spends a significant portion of its lifetime sitting idle, waiting for the next I/O operation to occur. To prevent performance degradation, SSD controllers frequently use idle time for performing garbage collection and wear leveling operations. By keeping an internal record of invalid pages waiting for erasure, a controller can free up entire blocks

so that they can be erased. By doing this work when the drive is idle, the controller ensures that free blocks are readily available.

The time period from when data is erased by the user and ultimately purged from the drive by the garbage collection mechanism varies from drive to drive. Some operating systems speed up the process by issuing the TRIM command, which notifies the SSD of deleted pages. On other solid state drives, this data will be permanently deleted very nearly immediately. On other drives, this data may remain on the drive in a stagnant but undeleted state for a longer duration of time.

Wear leveling procedures are also managed by a solid state drive's controller. As previously discussed, flash memory cells have a finite number of write cycles that they can sustain before they degrade and become unusable. To compensate for the uneven wear patterns caused by general operating system I/O patterns, controllers will move data to different flash cells. Data that is frequently rewritten will be moved to cells that are fresher, whereas more constant data will be moved to cells that have seen a higher degree of wear. The ultimate goal of this process is to ensure a fairly uniform distribution of write cycles across the entire set of flash memory in a solid state drive. The wear leveling process is ongoing and constant, allowing for the longest drive lifetime possible.

SSD controllers also frequently perform a significant amount of data processing to optimize it for storage on the flash media. Although these processes vary according to the type of controller and manufacturer's firmware implementation, common manipulations include compression, deduplication, and striping of data across flash chips for redundancy.

At any given point in time, there is a significant amount of work being performed by a drive controller. One of the most disastrous side effects of read and write cycles is the concept of write amplification. Frequently, writing data to flash requires much more effort within the drive than simply

recording the data to flash and moving on to the next task. Wear leveling and garbage collection processes also require write and erase cycles to accomplish these tasks. Although write amplification is impossible to avoid, manufacturers seek to develop methods to reduce write amplification factors as much as possible. To prevent redundant writes of information that already is stored, some SSD controllers may analyze incoming data and temporarily store it on an internal cache, for comparison with existing data. In the case of a small modification of a large file, it is likely that a majority of the file will remain constant. Some controllers will recognize this, and write out only the changes to the file as opposed to the entire file. This feature greatly reduces the amount of file manipulation that occurs within the flash memory itself.

Deduplication seeks to avoid writing identical blocks of data to multiple locations on the drive. A controller that supports deduplication will analyze incoming data and compare its contents to cells already written. If there is an identical match, the controller will use a single block to represent both blocks. In a perfect scenario, a drive containing multiple copies of the same file will only store a single copy of data within the flash memory. Although data deduplicated by a solid state controller will actually result in more free flash space being available within the SSD, this space is reserved by the drive and not revealed to the user as additional storage space.

In many respects, the SSD is much less related to a standard hard drive than is apparent to the user. The underlying hardware and software come closer to emulating an enterprise level SAN or RAID array of flash chips than a simple hard drive.

Evidence Collection

Perhaps the most critical stage of any forensics case is the evidence collection process. Any mistakes that are made during this stage can have significant and devastating negative ramifications on the success of the entire case. In order for evidence to be considered admissible in a court of law, a forensics investigator must be able to demonstrate that sound forensics techniques were utilized throughout the course of the investigation. The investigator must also be able to demonstrate the chain of custody for the evidence, and also be able to ascertain that the evidence was never modified, altered, or corrupted as a result of the investigation.

One of the standard tools used to prevent contamination of evidence during a forensics investigation is a write blocker. This is a device that allows for a source drive to be imaged, while prohibiting, or blocking, any write operations from being performed on the evidence drive. In order for evidence to be considered to be admissible in court, the use of a write blocker is expected.

Methodology

This work is primarily a quantitative study, but there were also some qualitative methods employed as well. To begin with, a pool of solid state drives was acquired. Initially, it was intended to accomplish this by contacting drive manufacturers and requesting loaner or demo models for this research, in order to obtain a pool of various manufacturer's devices and controller chips to achieve a sufficient sample size. The goal was to obtain a large sample size, with a contingency for conducting research on any drives that became available. Ultimately, it proved difficult to obtain participation from the majority of manufacturers that were contacted, either due to the lack of response or intellectual property concerns. To prevent this setback from diluting the sample size, several commercially available drives from various manufacturers were obtained for testing. The drives used for these tests represent a sample of drives available during the 2012 model year from various manufacturers. For comparison, a standard magnetic hard drive was used as a control for all tests.

All tests were conducted to simulate a forensics evidence capture process. This primarily included the imaging and recovery of deleted files from the pool of drives. Various factors were manipulated, one at a time, to determine which activities might result in a successful or unsuccessful recovery of files on the drive.

In order to understand the exact differences between solid state drives and physical hard drives with respect to the management of deleted evidence files, a series of experiments were conducted to explore the operation of the test pool of drives. A variety of test scenarios were designed to track the results of minor changes on the contents of a drive. These tests included the deletion of an individual file (using various operating systems and interfaces) and the formatting of the drive (both quick and full). Since solid state drives can behave differently depending on the amount of capacity in use on the drive, experiments were conducted with a nearly blank drive to simplify the analysis of the data and to

reduce the number of potential variables. In all cases, tests were designed to build upon and verify the results of previous experiments.

Throughout the research process, a standardized test environment was used across all tests. Previous work has shown that the operating system in use can have a significant impact on the operation of a solid state drive, especially if SSD-specific ATA commands such as TRIM are implemented. To ensure that this property was investigated, operating systems that support and allow TRIM to be turned on and off (such as Windows 7) were used, as well as legacy non-TRIM supporting operating systems (such as Windows XP). All analysis was performed using open source or freely-available tools. The majority of the investigation was performed using data acquisition and file carving capabilities built in to the Caine Forensics Linux distribution. When determining which tools to use for each experiment, simplified approaches were preferred in order to minimize the number of potential variables.

A dedicated laptop was used for creating all of the “evidence” drives used for all of the tests. All of these drives were used as secondary drives in this machine; the operating system was run on a different drive. This was done to allow maximum control of any data or file operation on the drives being researched, and also to reduce the likelihood of any background operating system behavior resulting in corrupted data.

A dedicated desktop was used for all of the data acquisition. All drive images were made using a commercial write blocker device, ensuring that all data collection was conducted in a forensically sound manner. After initial analysis, drive images were compressed and stored on an external drive for any follow up research that would be necessary. To prepare each drive for subsequent tests, an ATA Secure Erase command was issued to the drive. When properly implemented, this command resets the drive to factory defaults and completely wipes clean all data-bearing storage regions of the drive.

Quantitative methods were generally used to analyze the hard data generated as part of the research experiments described above. Tests were designed to generate information about the viability of forensically recovering deleted files on an SSD. The primary contribution of this research is the generation of detailed discussions regarding the results and their underlying forensics implications.

When evaluating the overall practicality of forensics approaches using SSDs, the outcomes of the tests were synthesized with the results of the research. Some of these conclusions tended to be qualitative in nature, and will be presented as a discussion of the major areas of concern with respect to SSD security. A compilation of best practices for handling and working with the evidence contained and stored on the drives will also be included at the conclusion of this paper.

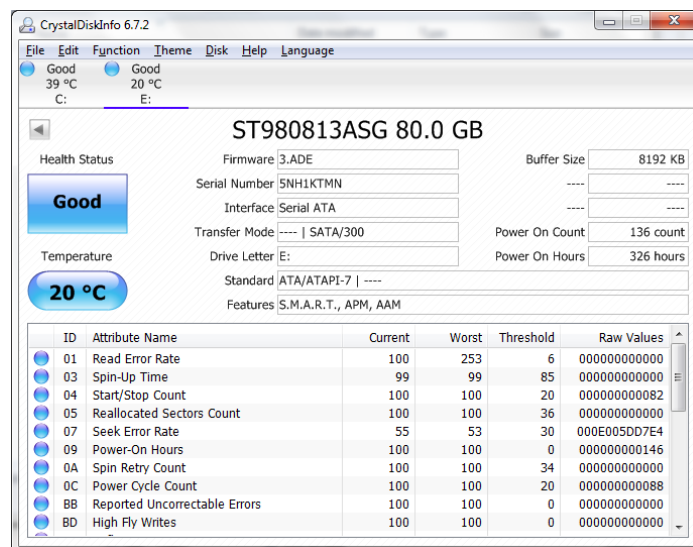
Materials

The experiments associated with this research were performed across a pool of six solid state drives. A seventh drive, which is a traditional magnetic device, was used as a control for all tests. Due to a failure of one of the solid state drives in the test pool, some of the later tests were conducted across five drives, with the sixth being the control. The CrystalDiskInfo utility was used to capture the firmware revision and version information for each of these drives.

Control: Seagate ST98013ASG, 80GB, 7200 RPM

The control drive is a standard Seagate laptop hard drive, with a capacity of 80 GB and a rotational speed of 7200 RPM. This drive represents a typical laptop hard drive that might be replaced with a similarly sized solid state drive. The behavior of this drive in a forensics investigation should model the expected behavior of the overwhelming majority of traditional hard drives.

The firmware on this drive was not upgraded and represents the code originally shipped when the drive was new.



ID	Attribute Name	Current	Worst	Threshold	Raw Values
01	Read Error Rate	100	253	6	000000000000
03	Spin-Up Time	99	99	85	000000000000
04	Start/Stop Count	100	100	20	000000000082
05	Reallocated Sectors Count	100	100	36	000000000000
07	Seek Error Rate	55	53	30	000E005007E4
09	Power-On Hours	100	100	0	000000000146
0A	Spin Retry Count	100	100	34	000000000000
0C	Power Cycle Count	100	100	20	000000000088
BB	Reported Uncorrectable Errors	100	100	0	000000000000
BD	High Fly Writes	100	100	0	000000000000

Figure 1: Seagate Drive Disk Info

SSD #1: Intel SSD 320, 120GB

The Intel SSD 320 contains the same controller as previous Intel SSDs, but is running improved firmware. One of the unique features of this controller is native encryption, where all of the data is committed to flash encrypted using AES-128. According to the manufacturer, this allows for secure erase operations to be performed simply by changing the encryption key. The controller in the Intel SSD 320 offers TRIM support.

The firmware revision in this SSD (4PC10362) was released to address a significant bug that presented the drive capacity as 8 megabytes (MB) and prevented any data on the drive from being accessed or modified (Frosty).

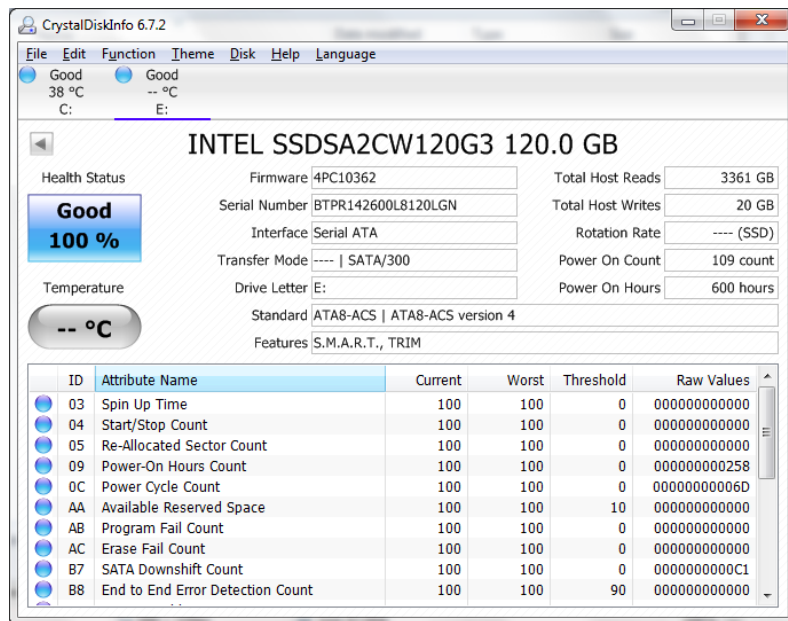


Figure 2: Intel SSD Info

SSD #2: Crucial M4, 128GB

The Crucial M4 SSD uses a Marvell 88SS9174-BLD2 controller, running firmware written by Crucial. It is considered an evolution of the controller used in previous generation Crucial SSDs, with an improved firmware revision. The controller in the Crucial M4 SSD offers TRIM support.

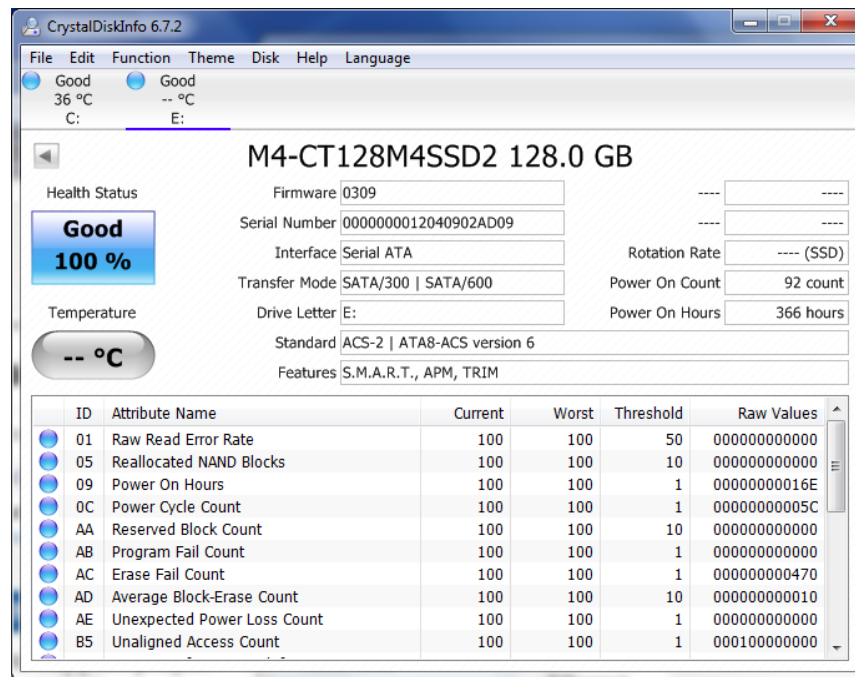


Figure 3: Crucial SSD Info

SSD #3: Patriot Pyro SE, 128GB

The Patriot SSD uses a SandForce SF-2281 controller. This SandForce controller implements some data management techniques that are unique, including the analysis and deduplication of incoming data prior to it being written to flash. This controller is used by a number of solid state drive manufacturers in various SSD models. The drive tested was running 3.3.2 firmware, which was the latest available when the test drive was produced. Subsequent firmware versions have been released since the tests were performed that include release notes indicating that the TRIM behavior for this controller has been modified and improved (Vättö).

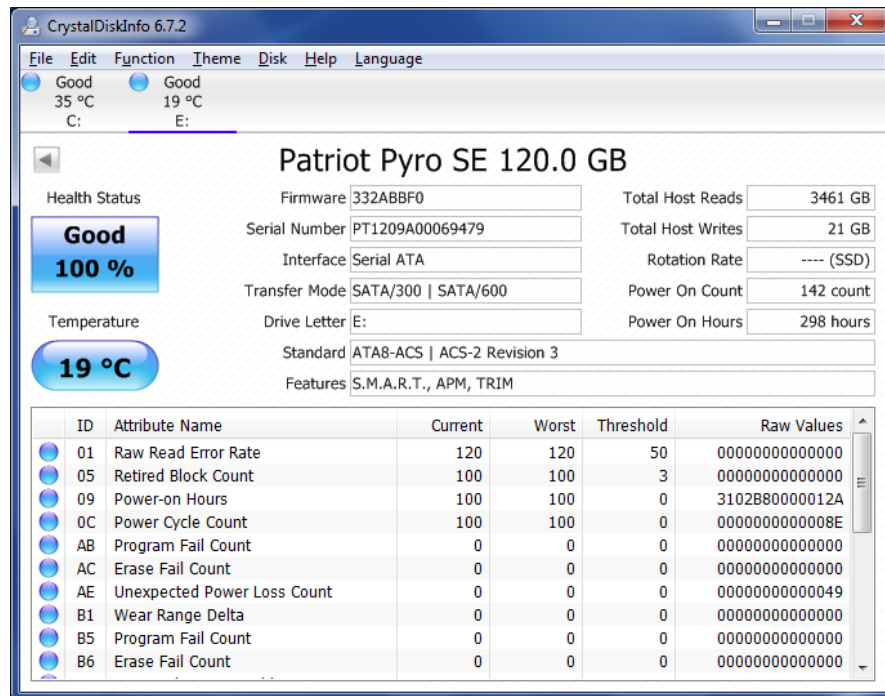


Figure 4: Patriot SSD Info

SSD #4: Samsung 830, 128GB

The Samsung 830 SSD uses a multi-core ARM based controller. There is not much information publicly available about its operation other than it is a multi-core design (a total of three cores) and based on the ARM architecture (Shimpi). However, the type of processing done by each core in this controller is unknown. The Samsung SSD supports the TRIM command.

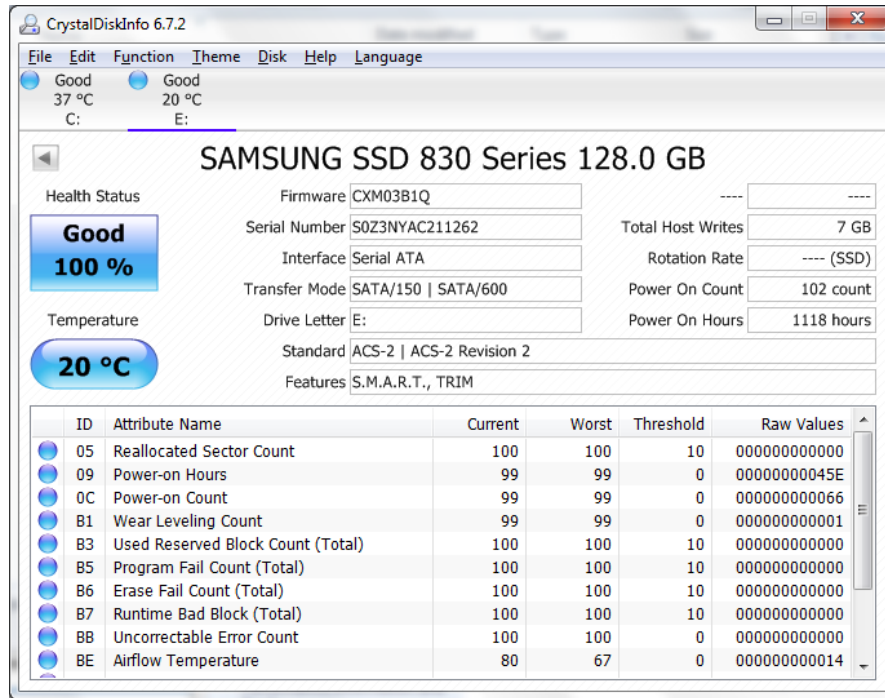


Figure 5: Samsung SSD Info

SSD #5: OCZ Agility 3, 60GB

The OCZ Agility 3 SSD is similar to the Patriot Pyro SE SSD in that it also uses the SandForce 2281 controller, but contains half the usable flash memory as the Patriot drive. This drive was included in the pool of sample drives to determine if consistent behavior is seen across different manufacturers' drives using the same controllers. This drive suffered a failure during the timed deletion tests and consequently was not used for several of the final experiments.

Note: no info screenshot is available for this SSD due to drive failure

SSD #6: SuperTalent MasterDrive, 64GB

This SSD is unique that it consists of a parallel ATA (PATA) flash drive which interfaces to the system using a SATA to PATA bridge chip. This drive represents one of the earliest generations of solid state drives and was manufactured well before the other drives in the test pool. The presence of the SATA to PATA bridge chip precludes the sending of any native SATA control commands, such as the TRIM command, to the drive, which also does not support the TRIM command.

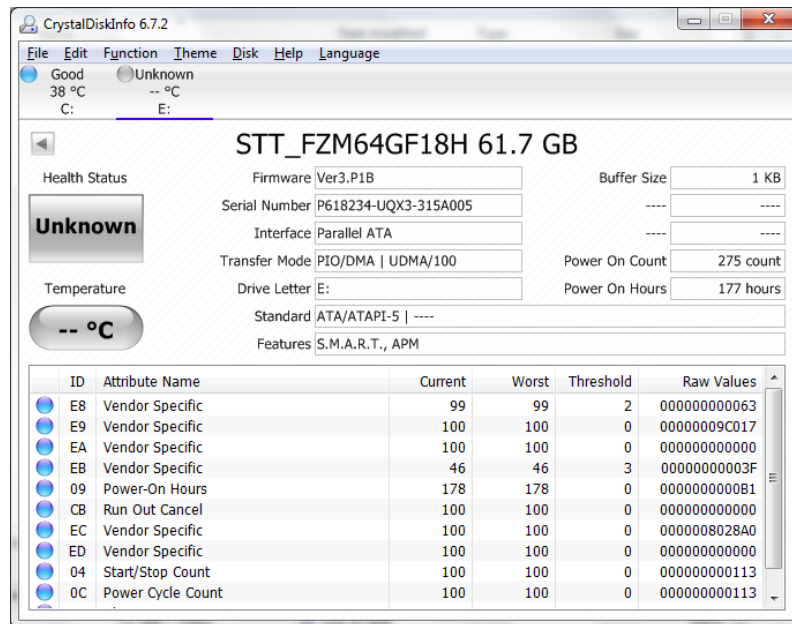


Figure 6: SuperTalent SSD Info

Forensics Lab Configuration

The forensics lab used for these tests consisted of two separate computers – one for evidence creation and the second for evidence acquisition and analysis. This allowed for the complete isolation of these tasks.

The evidence generation machine was a Dell Latitude E6500 laptop. For the majority of the tests, this machine was configured to run Windows 7 Professional 64-bit. A small sampling of the tests was performed using a Windows XP 32-bit installation to represent an operating system without TRIM support. This machine has both eSATA and USB interfaces, which allowed for the operating system and the SATA bus of the laptop to interact directly with the controller on the drives tested. Direct SATA access is required for native ATA commands such as TRIM to function as designed. USB connections were used for some tests to block TRIM functionality by means of the drive's interface.

For all tests, the drive being analyzed was independent of the boot drive containing the evidence generation machine's operating system. This was done to limit the write and access activity to the drive being investigated to only the tests being performed if at all possible. This research, which consists of executing and analyzing a series of repeated, minute changes, would be significantly more difficult to analyze and more likely to produce ambiguous results had the test drives also been running the operating system.

The data acquisition machine was a HP DC7700 desktop running the Caine 2.0 forensics distribution, which is based on Ubuntu 12.04. Any acquisition was conducted using a Diskology Disk Jockey Pro Forensic write-blocker, which requires the use of a USB 2.0 connection for write-blocking functionality. All images were stored on an external 3TB USB hard drive for analysis and long-term storage. All drive images were compressed after analysis for future reference.

The Caine distribution provides several built-in forensics tools which were used for the data acquisition and analysis portions of this research. The primary image creation/acquisition tool was Guymager, which is included in the Caine interface. Recovery of deleted files and file carving was performed using Selective File Dumper (SFDUMPER). Disk images were compressed after analysis using the Linux tar binary.

Experiments

The following sections will provide an overview of each test that was performed, the intent of each test, and a summary of the results of these tests. All tests consisted of two parts – a file deletion test and a subsequent quick format test.

Test 1, Part 1 – Text File Deletion

Procedure

All drives were formatted to their full capacity and connected to the evidence creation machine via eSATA. A single text file was created on the drive with one line of text: “Test File”. Once this file was copied to each drive, the drive was unmounted to ensure that the file was written to disk. Then, the disk was re-mounted and the file was deleted. The drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine if the text was recoverable.

Results

The text file was recoverable on all drives.

Test 1, Part 2 – Text File Deletion – Quick Format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. The Unix “strings” command was run across the image to determine if the text from the text file was still recoverable.

Results

The text string was recoverable from the control hard drive and the Patriot SSD. All other solid-state drives did not have the text string.

Test 2

Due to a numbering error, test 2 was omitted. To avoid confusion, all tests will be referenced using their originally assigned numbers.

Test 3, Part 1 – Single JPG File

Procedure

A single JPG file was saved to each drive using the evidence creation machine via eSATA. The evidence machine was running Windows 7 with the TRIM command enabled in the operating system (this is the Windows 7 default). Once this file was copied to each drive, the drive was unmounted to ensure that the file was written to disk. Then, the disk was re-mounted and the file was deleted. The drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine if the image file was recoverable.

Results

The file was recoverable on the control hard drive. It was not recoverable on all of the SSDs. A list of the per-drive results follows:

- Seagate HD – recoverable
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – recoverable
- Patriot SSD – recoverable
- Samsung SSD – partially recoverable (parts of file missing)
- SuperTalent SSD – recoverable (with file carving)

Test 3, Part 2 – Single JPG File, Quick Format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if the image was still recoverable.

Results

The quick format did not change the results of the part 1 of this test. All drives with recoverable or partially recoverable image files remained in the same state after the quick format. Drives with irrecoverable image files continued to be irrecoverable. A list of the per-drive results follows:

- Seagate HD – recoverable
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – recoverable
- Patriot SSD – recoverable
- Samsung SSD – partially recoverable (parts of file missing)
- SuperTalent SSD – recoverable (with file carving)

Test 4, Part 1 – Single JPG File, TRIM disabled

Procedure

A single JPG file was saved to each drive using the evidence creation machine via eSATA. The evidence machine was running Windows 7 with the TRIM command disabled in the operating system. Once this file was copied to each drive, the drive was unmounted to ensure that the file was written to disk. Then, the disk was re-mounted and the file was deleted. The drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine if the image file was recoverable.

Results

The file was ultimately recoverable on all of the drives. Only the Intel SSD required additional file carving to recover the image file. A list of the per-drive results follows:

- Seagate HD – recoverable
- Crucial SSD – recoverable
- Intel SSD – recoverable (with file carving)
- OCZ SSD – recoverable
- Patriot SSD – recoverable
- Samsung SSD – recoverable
- SuperTalent SSD – recoverable

Test 4, Part 2 – Single JPG File, TRIM Disabled, Quick Format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if the image was still recoverable.

Results

The quick format resulted in the image becoming irrecoverable on several of the SSDs. The image was recoverable from two of the SSDs and the control hard drive using file carving. A list of the per-drive results follows:

- Seagate HD – recoverable (with file carving)
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – recoverable (with file carving)
- Patriot SSD – recoverable (with file carving)
- Samsung SSD – not recoverable
- SuperTalent SSD – recoverable

Test 5, Part 1 – Single JPG File, TRIM enabled, SSD connected via USB

Procedure

A single JPG file was saved to each drive using the evidence creation machine via USB. The evidence machine was running Windows 7 with the TRIM command enabled in the operating system. Since TRIM is a native ATA/SATA command, this test was designed to explore the behavior of these drives when using an interface that does not support these native commands. Once this file was copied to each drive, the drive was unmounted to ensure that the file was written to disk. Then, the disk was re-mounted and the file was deleted. The drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine if the image file was recoverable.

Results

The file was ultimately recoverable on all of the drives. Only the SuperTalent SSD required additional file carving to recover the image file. A list of the per-drive results follows:

- Seagate HD – recoverable
- Crucial SSD – recoverable
- Intel SSD – recoverable
- OCZ SSD – recoverable
- Patriot SSD – recoverable
- Samsung SSD – recoverable
- SuperTalent SSD – recoverable (with file carving)

Test 5, Part 2 – Single JPG File, TRIM enabled, SSD connected via USB, Quick Format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via USB. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if the image was still recoverable.

Results

The image was recoverable with file carving on all drives with the exception of the Crucial SSD. A list of the per-drive results follows:

- Seagate HD – recoverable (with file carving)
- Crucial SSD – not recoverable
- Intel SSD – recoverable (with file carving)
- OCZ SSD – recoverable (with file carving)
- Patriot SSD – recoverable (with file carving)
- Samsung SSD – recoverable (with file carving)
- SuperTalent SSD – recoverable (with file carving)

Test 6, Part 1 – Single JPG File, Windows XP

Procedure

A single JPG file was saved to each drive using the evidence creation machine via eSATA. The evidence machine was running Windows XP, which does not offer TRIM support in the operating system. Once this file was copied to each drive, the drive was unmounted to ensure that the file was written to disk. Then, the disk was re-mounted and the file was deleted. The drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine if the image file was recoverable.

Results

The results were identical to the previous test on Windows 7 with TRIM disabled: the file was ultimately recoverable on all of the drives. A list of the per-drive results follows:

- Seagate HD – recoverable
- Crucial SSD – recoverable
- Intel SSD – recoverable
- OCZ SSD – recoverable
- Patriot SSD – recoverable
- Samsung SSD – recoverable
- SuperTalent SSD – recoverable

Test 6, Part 2 – Single JPG File, Windows XP, quick format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if the image was still recoverable.

Results

The image was recoverable with file carving on all drives. A list of the per-drive results follows:

- Seagate HD – recoverable (with file carving)
- Crucial SSD – recoverable (with file carving)
- Intel SSD – recoverable (with file carving)
- OCZ SSD – recoverable (with file carving)
- Patriot SSD – recoverable (with file carving)
- Samsung SSD – recoverable (with file carving)
- SuperTalent SSD – recoverable (with file carving)

Test 7, Part 1 – Two JPG Files, one deleted

Procedure

Two different JPG files were saved to each drive using the evidence creation machine via eSATA. The evidence machine was running Windows 7, with TRIM support enabled. Once these files were copied to each drive, the drive was unmounted to ensure that the files were written to disk. Then, the disk was re-mounted and a single file was deleted, leaving the second file untouched. The drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine if the deleted image file was recoverable.

Results

The image was completely recoverable on only half of the SSDs. Both the Crucial and Intel SSD performed garbage collection, rendering the deleted file unrecoverable. The file on the Samsung SSD was partially recoverable using file carving but damaged so that a significant portion of the file was corrupt/missing.

- Seagate HD – recoverable
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – recoverable
- Patriot SSD – recoverable
- Samsung SSD – partially recoverable – file is damaged
- SuperTalent SSD – recoverable

Test 7, Part 2 – Two JPG Files, one deleted, quick format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if either of the image files were still recoverable.

Results

After the quick format, both files were unrecoverable on the SSDs where the deleted file was unrecoverable or damaged during the previous test (Crucial, Intel, Samsung). Both files were recoverable from the other SSDs and the control hard drive.

- Seagate HD – both recoverable
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – both recoverable
- Patriot SSD – both recoverable
- Samsung SSD – not recoverable
- SuperTalent SSD – both recoverable

Test 8, Part 1 – Two JPG Files, 8 MB partition, one deleted

Procedure

A single 8 MB partition was created on each drive, which was the smallest possible partition that Windows 7 would create on these drives. Two JPG files were saved to each drive using the evidence creation machine via eSATA. The evidence machine was running Windows 7, with TRIM support enabled. Once these files were copied to each drive, the drive was unmounted to ensure that the files were written to disk. Then, the disk was re-mounted and a single file was deleted, leaving the second file untouched. The drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine if the deleted image file was recoverable.

Results

The results were identical to the previous two file test. The image was completely recoverable on only half of the SSDs. Both the Crucial and Intel SSD performed garbage collection, rendering the deleted file unrecoverable. The file on the Samsung SSD was partially recoverable using file carving but damaged so that a significant portion of the file was corrupt/missing.

- Seagate HD – recoverable
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – recoverable
- Patriot SSD – recoverable
- Samsung SSD – partially recoverable – file is damaged
- SuperTalent SSD – recoverable

Test 8, Part 2 – Two JPG Files, 8 MB partition, one deleted, quick format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if either of the image files were still recoverable.

Results

The results were identical to the quick format part of the previous two-file test. After the quick format, both files were unrecoverable on the SSDs where the deleted file was unrecoverable or damaged during the previous test (Crucial, Intel, Samsung). Both files were recoverable from the other SSDs and the control hard drive.

- Seagate HD – both recoverable
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – both recoverable
- Patriot SSD – both recoverable
- Samsung SSD – not recoverable
- SuperTalent SSD – both recoverable

Test 9, Part 1 – Two identical JPG Files, one deleted

Procedure

Two identical JPG files were saved to each drive using the evidence creation machine via eSATA. The evidence machine was running Windows 7, with TRIM support enabled. Once these files were copied to each drive, the drive was unmounted to ensure that the files were written to disk. Then, the disk was re-mounted and a single file was deleted, leaving the second file untouched. The drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine if the deleted image file was recoverable.

Results

The image was completely recoverable on only half of the SSDs. Both the Crucial and Intel SSD performed garbage collection, rendering the deleted file unrecoverable. The file on the Samsung SSD was partially recoverable using file carving but damaged so that a significant portion of the file was corrupt/missing.

- Seagate HD – recoverable
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – recoverable
- Patriot SSD – recoverable
- Samsung SSD – partially recoverable – file is damaged
- SuperTalent SSD – recoverable

Test 9, Part 2 – Two identical JPG Files, one deleted, quick format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if either of the image files was still recoverable.

Results

After the quick format, both files were unrecoverable on the SSDs where the deleted file was unrecoverable or damaged during the previous test (Crucial, Intel, Samsung). Both files were recoverable from the other SSDs and the control hard drive.

- Seagate HD – both recoverable
- Crucial SSD – not recoverable
- Intel SSD – not recoverable
- OCZ SSD – both recoverable
- Patriot SSD – both recoverable
- Samsung SSD – not recoverable
- SuperTalent SSD – both recoverable

Test 10, Part 1 – 60 numbered JPG Files, one deleted per minute over an hour

Procedure

This was the first of several tests designed to observe the behavior of solid state drives as files are deleted over a period of time. To begin, sixty numbered JPG files were created, each from the identical base image with a sequential number superimposed on the upper left corner of each image. The filenames of each file corresponded to the number in the image. A simple PowerShell script was created to delete one image at a time sequentially every minute over the course of an hour. All JPG files were saved to each drive using the evidence creation machine via eSATA. The evidence machine was running Windows 7, with TRIM support enabled. Once these files were copied to each drive, the drive was unmounted to ensure that the files were written to disk. Then, the disk was re-mounted and the script was executed, allowing each image file to be deleted. Once the script completed execution, the drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The drive image was then investigated to determine which deleted image files were recoverable.

Results

The images were completely recoverable on only half of the SSDs. Both the Crucial and Intel SSDs performed garbage collection, rendering all of the deleted files unrecoverable. Most of the files on the Samsung SSD were not recoverable, but a few of the files were partially recoverable using file carving but damaged so that a significant portion of each file was corrupt/missing.

- Seagate HD – all recoverable
- Crucial SSD – none recoverable
- Intel SSD – none recoverable

- OCZ SSD – all recoverable
- Patriot SSD – all recoverable but #1
- Samsung SSD – only six files were partially recoverable (damaged files), the rest were unrecoverable
- SuperTalent SSD – all recoverable

Test 10, Part 2 – 60 numbered JPG Files, one deleted per minute over an hour, quick format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if any of the image files were still recoverable.

Results

After the quick format, all files were unrecoverable on the SSDs where the deleted file was unrecoverable or damaged during the previous test (Crucial, Intel, Samsung). With the exception of a single file on the SuperTalent SSD, all files were recoverable from the other SSDs and the control hard drive.

- Seagate HD – all recoverable
- Crucial SSD – none recoverable
- Intel SSD – none recoverable
- OCZ SSD – all recoverable
- Patriot SSD – all recoverable
- Samsung SSD – none recoverable
- SuperTalent SSD – all files but #37 were recoverable

Test 11, Part 1 – 60 numbered JPG Files, one deleted per minute over an hour, TRIM Disabled

Procedure

This was the second of several tests designed to observe the behavior of solid state drives as files are deleted over a period of time. This test was identical to the previous, with the exception that TRIM was disabled in the operating system. The same PowerShell script was used to delete one image at a time sequentially every minute over the course of an hour. All JPG files were saved to each drive using the evidence creation machine via eSATA. Once these files were copied to each drive, the drive was unmounted to ensure that the files were written to disk. Then, the disk was re-mounted and the script was executed, allowing each image file to be deleted. Once the script completed execution, the drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine which deleted image files were recoverable.

Results

The image was completely recoverable on all of the SSDs. The results of these tests demonstrate the importance of the TRIM command with respect to automatic garbage-collection.

- Seagate HD – all files were recoverable, #1 required file carving
- Crucial SSD – all recoverable
- Intel SSD – all recoverable
- OCZ SSD – Drive failed, unable to complete test
- Patriot SSD – all files were recoverable, #1 required file carving
- Samsung SSD – all recoverable
- SuperTalent SSD – all files were recoverable, 1-6 required file carving, 7-60 normally

Test 11, Part 2 – 60 numbered JPG Files, one deleted per minute over an hour, TRIM Disabled, quick format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if any of the image files were still recoverable.

Results

After the quick format, files were unrecoverable on half of the SSDs (Crucial, Intel, Samsung) and recoverable on the remaining SSDs and control hard drive (Seagate, Patriot, and SuperTalent). While the OCZ SSD was unable to complete this test, based on the results of previous tests, we can predict that this SSD would have behaved similarly to the Patriot SSD. This test clearly shows the impact of the TRIM command and also the impact of different manufacturer's firmware implementations.

- Seagate HD – all recoverable
- Crucial SSD – none recoverable
- Intel SSD – none recoverable
- OCZ SSD – Drive failed, unable to complete test
- Patriot SSD – all recoverable
- Samsung SSD – none recoverable
- SuperTalent SSD – all files but #1 recoverable

Test 12, Part 1 – 60 numbered JPG Files, one deleted per minute over an hour, USB connection

Procedure

This was the third of several tests designed to observe the behavior of solid state drives as files are deleted over a period of time. This test was identical to the previous, with the exception that TRIM was enabled in the operating system and the drive was connected via USB instead of eSATA. The same PowerShell script was used to delete one image at a time sequentially every minute over the course of an hour. All JPG files were saved to each drive using the evidence creation machine via USB. Once these files were copied to each drive, the drive was unmounted to ensure that the files were written to disk. Then, the disk was re-mounted and the script was executed, allowing each image file to be deleted. Once the script completed execution, the drive was immediately unmounted and connected to the write blocker on the evidence collection machine and imaged. The image was then investigated to determine which deleted image files were recoverable.

Results

The image was completely recoverable on all of the functioning SSDs. The results of these tests further demonstrate the importance of the TRIM command with respect to automatic garbage-collection.

- Seagate HD – all recoverable
- Crucial SSD – all recoverable
- Intel SSD – all recoverable
- OCZ SSD – Drive failed, unable to complete test
- Patriot SSD – all recoverable
- Samsung SSD – all recoverable
- SuperTalent SSD – all files were recoverable, 1 required file carving, 2-60 normally

Test 12, Part 2 – 60 numbered JPG Files, one deleted per minute over an hour, USB connection, quick format

Procedure

Quick format of all drives from the previous test was performed using the evidence creation machine via eSATA. Drives were immediately safely removed from the system and imaged using the forensic write-blocker. File carving was run across each drive image to determine if any of the image files were still recoverable.

Results

After the quick format, files were unrecoverable on half of the SSDs (Crucial, Intel, Samsung) and recoverable on the remaining SSDs (Patriot and SuperTalent). While the OCZ SSD was unable to complete this test, based on the results of previous tests, we can predict that this SSD would have behaved similarly to the Patriot SSD. This test clearly shows the impact of the TRIM command and also the impact of different manufacturer's firmware implementations.

- Seagate HD – 45/60 recoverable
- Crucial SSD – none recoverable
- Intel SSD – none recoverable
- OCZ SSD – Drive failed, unable to complete test
- Patriot SSD – all recoverable
- Samsung SSD – none recoverable
- SuperTalent SSD – all recoverable

Observations and Analysis

Overall, there were a number of differences between the behavior of the solid state drives when compared to the control standard hard drive. In addition, the behavior of the solid state drives was not necessarily consistent between drive models and manufactures across each individual test. However, there were several consistent patterns observed across each SSD model. In general, SSDs were observed to behave differently than standard drives in many tests, but the scope and magnitude of these differences varied across drive manufacturers and models.

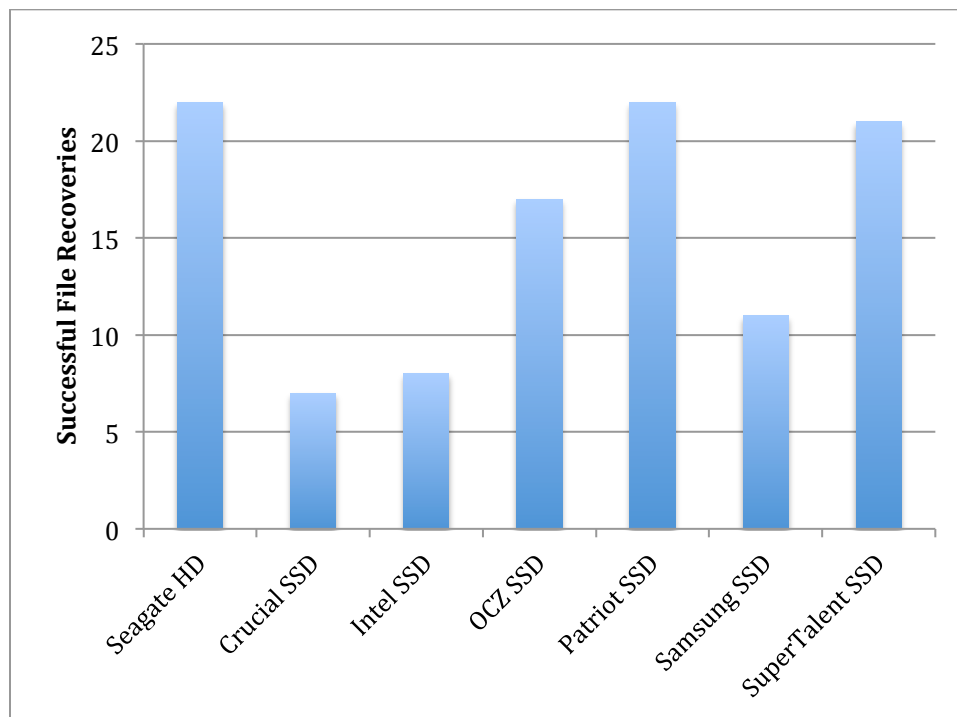


Figure 7: Overall recoverability of files across all tests, per drive

As seen in the graph, the success of recovering data varied significantly depending on the drive being tested. Some of the SSDs performed nearly identically to the control hard drive, whereas others demonstrated a significant decrease in the likelihood of data being successfully recovered.

Standing alone, however, this graph is somewhat misleading. The types of tests that were performed can be classified into two different categories: file deletion tests, and quick format tests, which were always performed following a file deletion test.

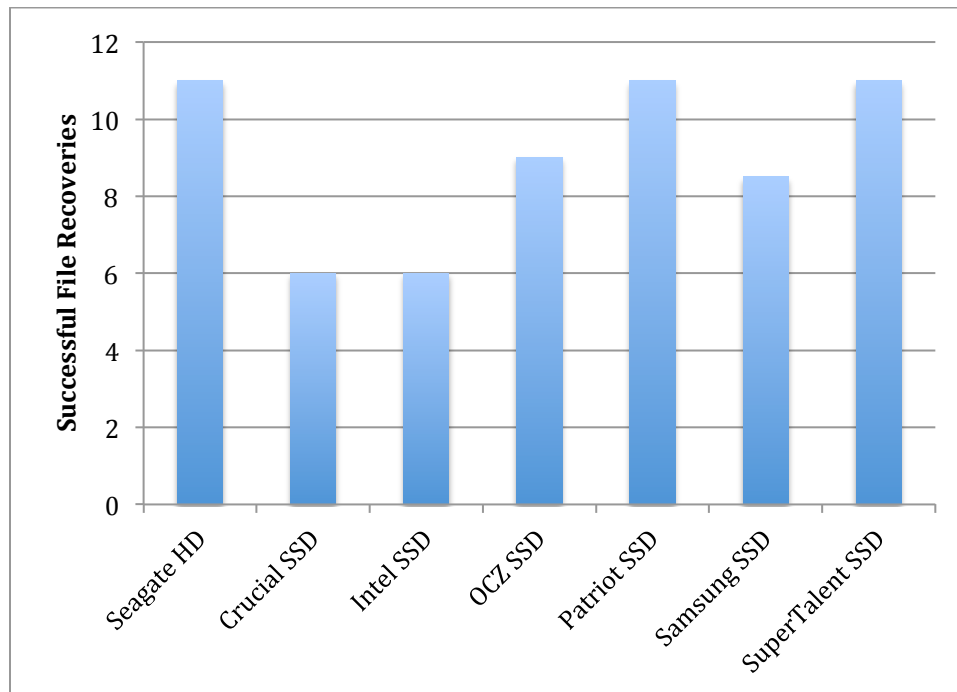


Figure 8: Overall recoverability of files across deletion tests, per drive

For file deletion tests, the behavior of two of the SSDs was identical to the control hard drive – files were recoverable in all tests. The OCZ SSD behaved similarly to these SSDs as well during all of the tests where it was functional. Two SSDs had unrecoverable files in a few of the tests, and the remaining two SSDs only allowed for data to be recovered in around 50% of the tests.

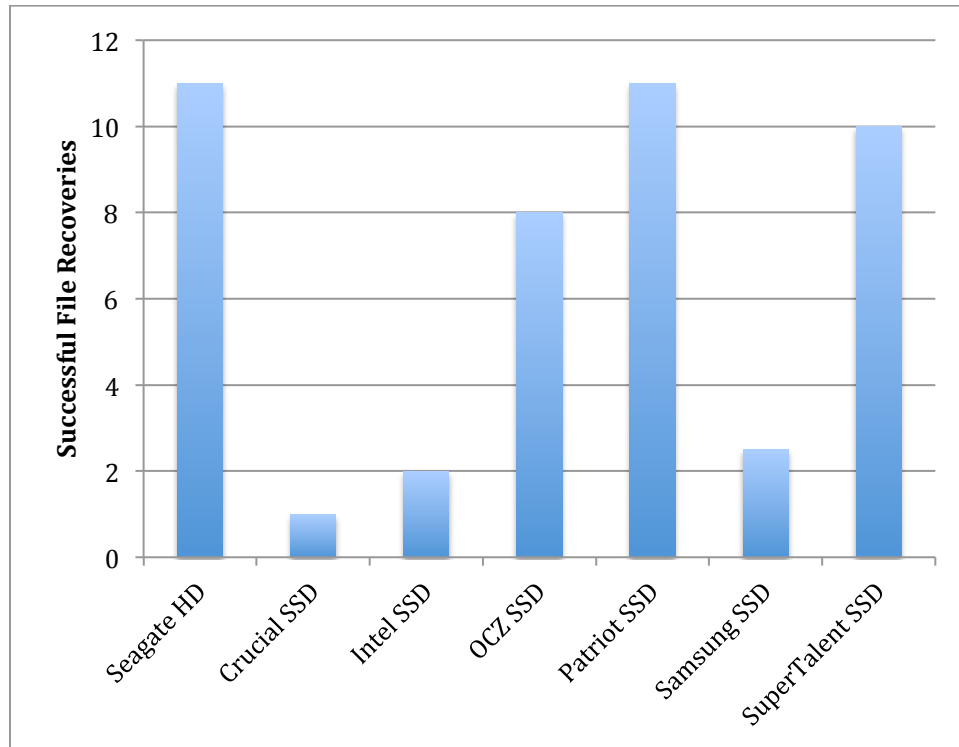


Figure 9: Overall recoverability of files - quick format tests, per drive

In the quick format tests, differences between SSDs and the control drive were even more apparent. Two of the SSDs continued to behave nearly identically to the control hard drive, while two different SSDs only allowed for recovery in one or two of the tests.

By far, the single most significant contributing factor to the results of any given test was the state of the TRIM command in the operating system at the time of the file deletion. Since the TRIM command is intended to notify the drive controller of a delete operation so that the drive may initiate its garbage-collection procedures, it is understandable that this setting resulted in such a significant change in behavior, as illustrated in the following figures.

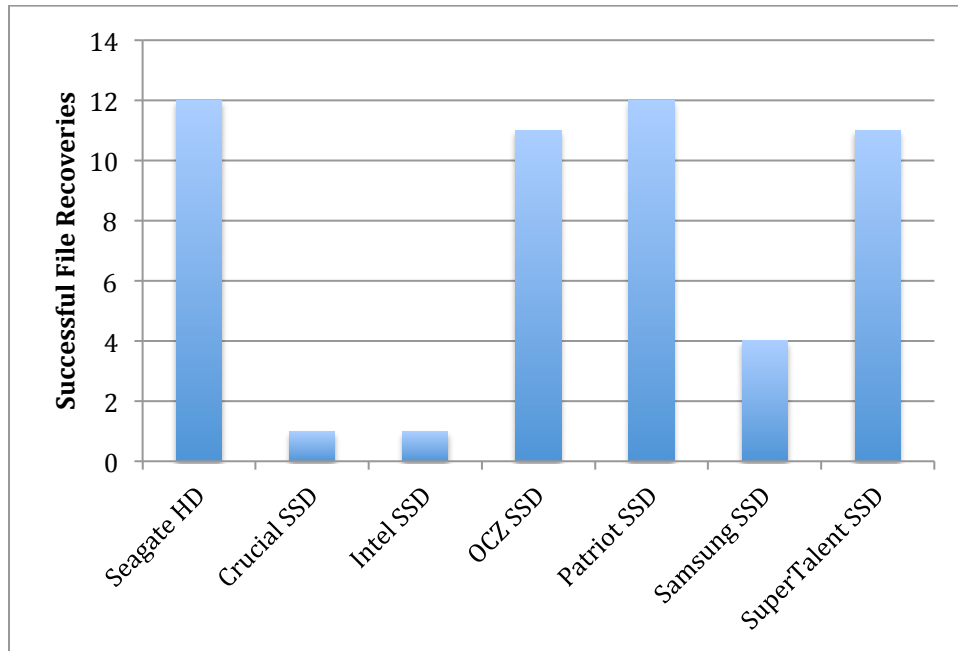


Figure 10: Overall recoverability of files across all tests where TRIM is enabled or available as a result of the operating system or interface, per drive

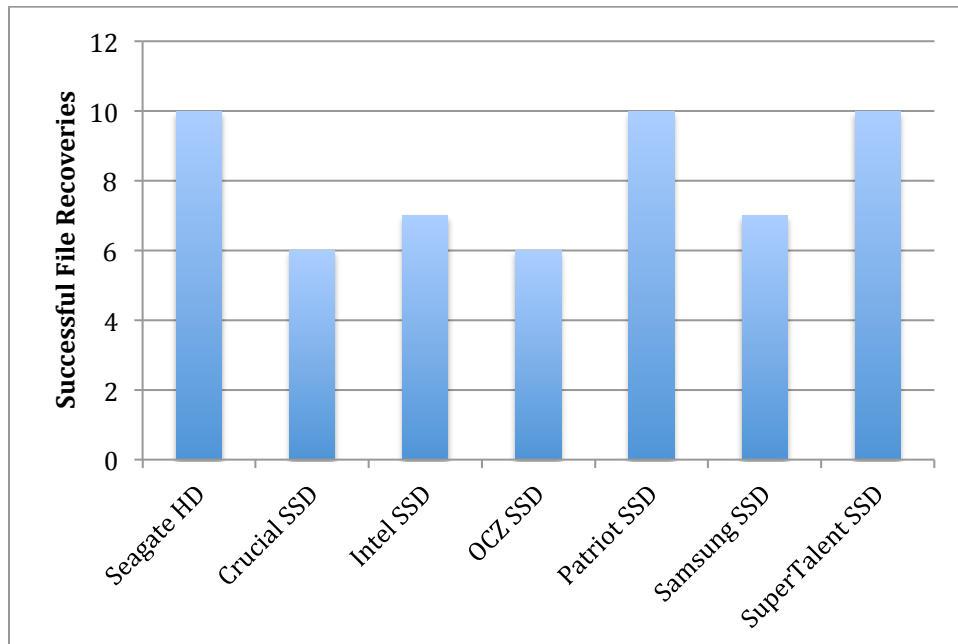


Figure 11: Overall recoverability of files across all tests where TRIM is disabled or unavailable as a result of the operating system or interface, per drive

Solid state drives with controllers featuring native TRIM support often purged data immediately or shortly after a file was deleted from the operating system where TRIM was enabled. Disabling TRIM for these drives resulted in file deletion behavior much more in line with the operation of a traditional hard drive. Solid state drives lacking TRIM support were significantly more likely to allow the recovery of files after deletion than their TRIM supporting counterparts.

Another slightly less common observation was the presence of a damaged or partially deleted file. This behavior appeared to be accompanied with a subset of the files on the drive being completely deleted, and seemed to indicate that the file contents either spanned across multiple flash cells or the garbage-collection process was not completely run across all flash cells containing the image data. Due to the nature of flash storage and the limitations concerning the erasure of cells (that is, cells can only be completely erased), these observations align with the architecture of the underlying storage hardware.

Solid state drives also behaved differently when a quick format command was issued by the operating system. When forensically analyzing a traditional hard drive that has been formatted with a quick format procedure (one that simply re-initializes the partition table and does not overwrite the data on the disk itself), it is often quite straightforward or even simplistic to recover the formatted data using file carving techniques, as the recoverable data is not truly overwritten on the disk (the same cannot be said of data that has been completely overwritten at least once, which is generally unrecoverable). Some of the solid state drives, on the other hand, appeared to initiate garbage-collection processes shortly after a quick format was issued, rendering data unrecoverable using file carving techniques.

Forensic Implications

The results of these experiments clearly demonstrate that solid state drives do not behave identically to traditional hard drives when attempting to forensically recover deleted files. Furthermore, the behavior of a given solid state drive cannot be generalized, as these characterizes were not consistent across the pool of test drives. This fundamentally shifts the digital forensics paradigm, where the recovery of deleted files from an un-sanitized magnetic hard drive is more or less assumed to always be possible.

Due to the variances in drive behavior, forensics investigators must be acutely aware of the specific drive and operating system combination that had been used at the onset of an investigation of an evidence bearing SSD. Knowing these factors in advance and performing tests on an identical drive (with identical firmware on a matching operating system) would provide an effective method for gauging the relative likelihood of successful recoverability for deleted data.

While many of the SSDs in the test pool exhibited lower rates of successful recoverability than the control drive, no SSD in the test pool purged data in every experimental run. This indicates that, at least at present, there is still value in performing traditional forensics techniques against a solid state drive. However, as drive controllers evolve and TRIM support becomes more universal, the likelihood of successful recoverability of deleted files in the future from evidence-bearing solid state drives would be expected to decrease. The forensics community must adapt to this technological change if they hope to continue to mine this data as viable, reliable evidence in the investigation and prosecution of criminal cases.

Conclusions

The results of these experiments consistently demonstrate that solid state drives generally cannot be considered to be equivalent to traditional hard drives from a forensic perspective. While there are a limited number of cases where a given set of drives performed similarly or identical to the control drive, there were significantly more scenarios where data was unable to be recovered from the solid state drives using traditional forensics techniques, or where the quantity of data that could be recovered from a solid state drive was severely diminished. Furthermore, these behaviors were not necessarily consistent across all solid state drives due to variations in the drive controller firmware and operation. This variation and inconsistency poses an additional challenge for forensic investigators, who must now consider significantly more challenges and complications when presented with evidence-bearing solid state drives.

Due to the variety of solid state drives on the market and the firmware differences across drive manufacturers, controllers, code revisions, and even individual drive models, it will be impossible to establish a single, conclusive standard for understanding how these drives will behave in all scenarios. That being said, there were a number of patterns observed throughout the experimental trials that may help to predict the behavior of a similar drive in a future test. The drives used in these tests capture a period of evolution in the solid state drive industry, where the technology was maturing and consumer adoption was increasing exponentially. The pool of drives tested effectively capture the evolution of this hardware and its behavior in many different scenarios. As these devices gain popularity and continue to see more widespread adoption, it is inevitable that the results of identical tests on newer drives may yield different results. Given the increased support of TRIM across modern operating systems and solid state drives, the author of this paper believes that the widespread adoption of these drives will pose a significant challenge for future forensics investigations. This challenge will only

increase as time goes on and further advancements are made in solid state drives, their controllers, and their firmware.

Future Work

The intent of this research was to effectively demonstrate differences in behavior between traditional hard drives and their solid state counterparts, as well as also highlighting variations between different SSDs as a result of the controller and firmware of the drive itself. This project lays the groundwork for the development of new forensics techniques to better address the challenges associated with forensically recovering data from these drives. One significant but unexplored possibility involves bypassing the drive controller and reading the contents of the underlying flash memory. This method would require an intimate knowledge of the controller behavior and specialized hardware. This possibility was determined to be outside of the scope of this research.

Works Cited

Bell, Graeme B., and Richard Boddington. *Solid State Drives: The Beginning of the End for Current*

Practice in Digital Forensic Recovery? Perth: Association of Digital Forensics, Security and Law,

2010. 5(3). *Journal of Digital Forensics, Security and Law*. Web. 1 Oct. 2011.

<http://researchrepository.murdoch.edu.au/3714/1/solid_state_drives.pdf>.

Bonetti, Gabriele, Marco Viglione, Alessandro Frossi, Federico Maggi, Stefano Zanero, and Politecnico Di

Milano. "A Comprehensive Black-box Methodology for Testing the Forensic Characteristics of

Solid-state Drives." *ACM Digital Library*. Association for Computing Machinery, 9 Dec. 2013.

Web. 1 Apr. 2016.

Chang, Li-Pin, and Chun-Da Du. *Design and Implementation of an Efficient Wear-Leveling Algorithm for*

Solid-State-Disk Microcontrollers. New York: ACM, 2007. Print.

Freeman, Michael, and Andrew Woodward. "Secure State Deletion: Testing the efficacy and integrity of

secure deletion tools on Solid State Drives." *ADF*. Proceedings of the 7th Australian Digital

Forensics Conference. Perth: n.p., 2009. 32-40. *Citeseer*. Web. 1 Oct. 2011.

<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.167.4914&rep=rep1&type=pdf>>.

Frosty, Alan. "Firmware Update Now Available - Addresses Bad Context 13x Error." *Solid State Drives*

Discussions. Intel Corporation, 17 Aug. 2011. Web. 08 Mar. 2016.

Grupp, Laura M., et al. *Characterizing Flash Memory: Anomalies, Observations, and Applications*. New

York: ACM, 2009. Print.

Gutmann, Peter. Secure Deletion of Data from Magnetic and Solid-State Memory. Department of Computer Science. University of Auckland, July 1996. Web. 4 Dec. 2011.

<http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html>.

King, Christopher, and Timothy Vidas. *Empirical Analysis of Solid State Disk Data Retention When Used with Contemporary Operating Systems*. N.p.: Elsevier Ltd, 2011. Digital Investigation 8. *ScienceDirect*. Web. 2 Oct. 2011. <<http://dfrws.org/2011/proceedings/17-349.pdf>>.

Leventhal, Adam. "Flash Storage Today." *ACM Queue* 51.7 (July-Aug. 2008): 24-30. Print.

Maghraoui,, Kaoutar El, et al. *Modeling and Simulating Flash based Solid-State Disks for Operating Systems*. San Jose: ACM, 2009. Print.

Moshayedi, Mark, and Patrick Wilkison. "Enterprise SSDs." *ACM Queue* July-Aug. 2008: 32-39. Print.

Shimpi, Anand Lal. "The Samsung SSD 830 Review." AnandTech. 24 Sept. 2011. Web. 1 Apr. 2016.

Vättö, Kristian. "SandForce TRIM Issue & Corsair Force Series GS (240GB) Review." *AnandTech*. 22 Nov. 2012. Web. 08 Mar. 2016.

Wei, Michael, et al. *Reliably Erasing Data from Flash-Based Solid State Drives*. San Diego: n.p., 2011. Print.