THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 5 | Number 3 Article 1

2010

Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?

Graeme B. Bell
Murdoch University, Perth

Richard Boddington
Murdoch University, Perth

Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?

Graeme B. Bell and Richard Boddington[1]
School of IT, Murdoch University, Perth, WA 6150, Australia.
{G.Bell, R.Boddington}@murdoch.edu.au.
Tel +61 89360 {6533,2801}. Fax +61 89360 2941

ABSTRACT

Digital evidence is increasingly relied upon in computer forensic examinations and legal proceedings in the modern courtroom. The primary storage technology used for digital information has remained constant over the last two decades, in the form of the magnetic disc. Consequently, investigative, forensic, and judicial procedures are well-established for magnetic disc storage devices (Carrier, 2005). However, a paradigm shift has taken place in technology storage and complex, transistor-based devices for primary storage are now increasingly common. Most people are aware of the transition from portable magnetic floppy discs to portable USB transistor flash devices, yet the transition from magnetic hard drives to solid-state drives inside modern computers has so far attracted very little attention from the research community.

Here we show that it is imprudent and potentially reckless to rely on existing evidence collection processes and procedures, and we demonstrate that conventional assumptions about the behaviour of storage media are no longer valid. In particular, we demonstrate that modern storage devices can operate under their own volition in the absence of computer instructions. Such operations are highly destructive of traditionally recoverable data. This can contaminate evidence; can obfuscate and make validation of digital evidence reports difficult; can complicate the process of live and dead analysis recovery; and can complicate and frustrate the post recovery forensic analysis.

Our experimental findings demonstrate that solid-state drives (SSDs) have the capacity to destroy evidence catastrophically under their own volition, in the absence of specific instructions to do so from a computer.

Keywords: digital evidence, digital forensic analysis, self-contamination, solid-state drive, SSD, garbage collection, write-blocker

1. INTRODUCTION

Evidence stored on electronic media normally contains useful metadata, such as key dates, times, and file antecedents, and so is sometimes considered superior to conventional paper evidence. Its persistence in recording important data provides potentially valuable information relating to events under investigation (Flusche, 2001; Janes, 2000). However, digital evidence is easily mutable and contamination-prone if ineffective recovery processes are used. Any loss of content or alteration of metadata can impair its legal admissibility and diminish its evidentiary weight during court proceedings (Ashcroft, 2001; Carrier & Spafford, 2003). Digital evidence is often technically complex and is difficult for most legal practitioners and judges to understand fully. It is even more confusing to layman jurors with little or no technical understanding of the nature of the evidence presented, or its evidentiary worth (Caloyannides, 2001; Carlton & Worthley, 2009; Edwards, 2005; Etter, 2001; Losavio, Adams & Rogers, 2006).

[1] Joint primary authors.

Digital evidence requires validation to test its reliability and gauge its admissibility and weight. It is important to identify any possible contamination or loss of data that has occurred during the recovery processes (Boddington, Hobbs & Mann, 2008). If a forensic image of a drive was (or was perceived to have been) altered during recovery, the burden of proving the image's integrity rests with the party tendering the evidence. Otherwise, the opposing party is able to question the integrity of the forensic image because of deficient recovery and preservation processes (Berg, 2000). Berg predicted that questions about the authenticity of a forensic image and whether it was tampered post-capture were likely to arise and would be difficult to refute (Berg, 2000).

To minimise contamination, the recovery process should prevent overwriting of data on the exhibit drive. For example, during dead analysis recovery all processes are terminated by de-powering the system before making a forensic image of the drive, thereby, theoretically reducing the risk of alteration to the data (Carrier, 2005). A cryptographic hash is calculated to allow parties to show that the data has not changed if the recovery process is repeated. Any change between hash values would show that data has been modified and would raise alarm about the integrity of the forensic image (Carrier, 2005). The recovery of digital data must be completed without causing change to the data thereby minimising future challenges during legal proceedings. When change does occur, the nature and cause for the change must be explained and described convincingly (Ghosh, 2004).

Contamination-avoidance during recovery is essential and depends on effective, error-free data recovery from digital devices. Traditionally, write-blocking hardware combined with bit-stream image copying processes is used (Kenneally & Brown, 2005). This is considered to allow data recovery and is believed to enhance validation of the reliability, accuracy, and completeness of the recovered evidence (Kenneally & Brown, 2005).

In this paper, traditional assumptions about the behaviour of digital media are shown not to hold universally. Evidence stored on modern internal primary storage devices can be subject to a process we label 'self-corrosion'. What is meant by this is that even in the absence of computer instructions, a modern solid-state storage device can permanently destroy evidence to a quite remarkable degree, during a short space of time, in a manner that a magnetic hard drive would not. Here, the phenomenon of solid-state drive (SSD) self-corrosion is proven to exist through experimentation using real world consumer hardware in an experimentally reproducible environment.

This paper is structured in the following way. An introduction to computer based information storage is presented that notes historical gaps between storage abstractions and real world behaviour (Section 2). This leads into a theoretical discussion of the behaviour of solid-state drives, and in particular, the aspects of these increasingly popular devices that cause 'self-corrosion' to take place (Section 3). Four experiments are described, and their corresponding quantitative and qualitative results are given (Section 4), and are followed by an interpretation of the results (Section 5). Based on this, we provide recommendations and guidance for anyone working with solid-state drives as a source of evidence (Section 6). The paper ends by discussing the future of storage technology and offering our conclusions (Section 7, 8). Readers are then presented with two Appendices describing the experimental environment (Appendix I) and offering source code listings for the experimental software (Appendix II).

2. STORAGE ABSTRACTIONS AND REAL WORLD BEHAVIOUR

Every kind of information that can be stored on a computer is represented at some point as a pattern of 1s and 0s, for the purposes of computation and storage. These 1s and 0s are, however, merely a convenient interpretation for human use – not a physical reality. In practice, what we consider to be a stored '1' or '0' is actually a complex pattern of some physical phenomenon such as magnetism or electric charge that exists at a particular place in a physical device. This idea is best demonstrated by example. In the present day, the main memory of most computers (often referred to as RAM) represents patterns of 1s and 0s by either charging (or not charging) a collection of tiny, battery-like capacitors. To write data, some of these memory cells are selectively filled with electric charge; to read the data back out again, the memory controller can 'open the gate'

and monitor how electricity comes pouring out of each cell. This seems simple enough, yet this type of data storage has unusual characteristics and complications that might not be guessed by a casual user. For example, such small, fast capacitors cannot store charge permanently or even for long periods, so it is necessary to regularly and automatically read out and recharge the memory to prevent data from silently fading away. We call such memory 'volatile' or 'dynamic' RAM, and we do not use it for long term storage of data, because the refreshing process requires a continual source of electrical power which can be inconvenient or impossible to ensure. This example demonstrates that there is a gap between what we imagine data storage is doing 'in the abstract' vs. the limitations, peculiarities and internal behaviours of real world storage devices.

Alternative physical technologies have been used to represent data that is being held for the long term. Magnetic tapes and disks store 1s and 0s as magnetic domains (essentially, small magnets) on a physical surface that either lie north-to-south or south-to-north. Again, the physical nature of a bulk storage medium imposes limits or peculiarities upon how information may be stored and accessed within it. Magnetic tapes can only be accessed in sequential order by playing the tape forward and backward; and magnetic disks induce delays and errors due to the need to rotate a magnetisable surface at high speed only microscopic distances from an electromagnetic read/write device. Optical media such as CDs, DVDs and Blu-Ray allow cheap, read-only bulk storage by relying on the physical interference of light-waves as they bounce from miniature dips and plateaus on a reflective surface; but burning the surface can be a relatively slow and inconvenient process, and the media is prone to deterioration over time and must be prepared appropriately before use.

As each of these new digital storage technologies has become available, a variety of tradeoffs has been made in order to gain access to increases in speed, cost effectiveness, capacity and convenience. It is therefore in the nature of computing that we perceive regular paradigm shifts in the ways that we store and process information. The most overt example is found in portable storage media, which has been visibly revolutionised many times over: punch cards replaced by portable magnetic tapes; then replaced in turn by magnetic floppy disks; optical disks (CD-R); portable external hard disks; and most recently, fast, reliable and compact USB-based flash memory sticks. Despite vast differences in the essence of internal operations within these physical systems, their similarities in terms of how they are stored, placed into a computer, and accessed have allowed repeated society-wide storage technology transitions.

In this paper, however, we are most interested in the primary storage of computers, the hard drive - a component which most normal users do not physically handle or observe. Hard drives are of primary importance in ascertaining the activities and information associated with a human in any matter of forensics or law. Surprisingly, the technology behind the hard disk drives in our computers today has existed since the 1950s (Hughes, 2002), and fixed, rotating magnetic platters have been the dominant form of primary storage in computers for a truly remarkable 50 years. Despite their pedigree, relatively few complexities have been introduced to regular hard disk control systems; generally, controller chips only try to optimise short-term disk accesses, cache for efficiency, and compensate for damage. The 3.5" hard disk was first produced in 1983, and consumer hard disk drives have not changed dramatically over the last 25 years in terms of physical size, appearance, or externally observed behaviour, despite decades of progress in technologies that improve their speed, reliability, cost and capacity.

Their simplicity and consistency has been a tremendous benefit for those operating in the domains of policing, forensics, and law. Well-established human processes and digital technologies exist, such as dead analysis and write-blockers, which make it convenient and acceptable for investigators to capture physical hard disks, extensively interrogate them for digital evidence, automatically interpret, summarise and present that evidence, and validate the quality of gathered evidence in front of a court (Carrier, 2005). These long-established, internationally accepted procedures even cover situations such as the automated recovery of court-submissible evidence which a defendant has previously attempted to delete. Indeed, the peculiarity of 'deleted, but not forgotten' data which so often comes back to haunt defendants in court is in many ways a bizarre artefact of hard drive technology. It derives from the fact hard disks have slow access speeds relative to their capacity for storage (which makes complete erasure very inconvenient), and from the fact that there is no performance penalty incurred for writing over existing data (which makes complete erasure unnecessary). Put simply, the

last few decades have in many ways been a golden era for digital evidence. Digital evidence stored on magnetic hard disks is awkward to expunge permanently. Hard disks generally provide the largest source of digital evidence, and recovery of deleted data is possible through the ubiquity, simplicity and homogeneity of modern drives. Police officers, lawyers, judges and juries stand a modest chance of being able to evaluate such evidence robustly. Disputes, therefore, arise mostly around the periphery of the data, and are generally argued in terms of the human interpretation of the significance and meaning of data, rather than in terms of the suitability of physical evidence gathering methods, or data extraction processes employed (Yasinsac et al., 2003).

This situation may be in the process of changing. Solid-state drives are faster and more complex cousins of the USB memory stick, which in the last few years have received strong interest (Chen et al., 2009). They are typically presented with the physical appearance and connectors of a regular hard drive, and are perceived as possessing advantages over traditional magnetic drives. Generally, they are faster, less heavy, less power-demanding (allowing longer battery life and less use of fans), allow random access to any part of the data stored without penalty, have lower latency, can cope with hostile environments (heat, cold, vibration, moisture, shock), do not introduce vibration, and offer good long-term reliability, depending on how they are used (Chen et al., 2009; Ekker et al., 2009; Drossel, 2007). Surprisingly, modern SSDs have a heritage in technologies that are as old as magnetic disks (e.g. "core memory"), and flash-based drives have had niche uses in domains such as military equipment for some 15 years or so (Odagiri, 2010; Drossel, 2007). It is only recently through the popularity of USB sticks, digital music devices, and camera storage that these flash-based devices have achieved a price point that makes their desirable capabilities affordable for the general population. As this has happened, further research and development have been undertaken by manufacturers to ameliorate unusual quirks perceived by consumers (described below), which have resulted in a significant increase in the complexity of these devices, particularly, their internal controller chips.

It is the view of the authors that the increasing popularity and increasing complexity of solid-state drives is a matter of great importance for everyone working with digital evidence. We believe (with the support of experimental findings) that existing processes, technologies, procedures and assumptions relating to digital evidence are rapidly reducing in appropriateness and utility as SSDs increasingly enter use in retail/consumer computers. This could have tremendous practical importance for capture of computing equipment, forensic practice, and court validation of digital data. Put simply: the SSD technology which is replacing magnetic hard drives inside many computers is neither simple, well understood, nor homogenous; rather it is complex, poorly documented, and highly heterogeneous. Worst of all, it is *active* - that is to say, the SSD may act under its own initiative, and may undertake quite remarkable (and highly evidence-destructive) actions even in the absence of write commands from a computer, potentially regardless of efforts by police and forensic analysts to prevent invalidation of evidence. We will now look at the theory of this problem and its technological origin.

3. THE PROBLEM POSED BY SOLID-STATE DRIVES

Let us begin by briefly contrasting the behaviour of a magnetic hard disk (HDD) and a flash memory-based solid-state drive (SSD). Magnetic HDDs operate by creating or detecting oriented magnetic fields in fixed regions of a magnetic surface, known as blocks. Each block has a 3D position and area within the disk known as a cylinder/head/sector address. This specifies the particular platter of the disk (modern disks have several magnetic surfaces), a particular track of the disk (distance from the centre) and the particular sector (angular distance around the disk, e.g. 'which slice of the pie'). When the disk is asked to write to a particular block, it translates this mathematically into the cylinder/head/sector position and magnetises the magnetic surface of the disk in that region appropriately. If there is already old discarded data in that location, the data is automatically converted to the new values, in much the same way as recording some TV over a used recordable video tape does not require any special preparation of the medium, and essentially obliterates the data that was there before. Very rarely, a sector of the disk will stop storing data correctly. If this happens, the disk controller will make a note of the problematic sector and will redirect any requests to a small area of alternative spare space each time someone attempts to access the broken sector.

In contrast, SSDs operate by storing data in (typically) 512kB blocks, subdivided into (typically) 4kB pages. These pages/blocks are comprised of large arrays of NAND transistors (Olson & Langlois, 2008). These

NAND transistors are essentially very similar to the NAND logic chips used to build computer processors (CPUs). However, they have an extra gate, known as a floating gate, which is used to trap charge (essentially, to store charge on an extremely limited scale as a capacitor might). This arrangement is very stable and can allow microscopic quantities of charge to be stored for years without leakage and without requiring a supply of power. The term *solid-state* refers to the fact that the data is stored in fixed arrangements of electronic transistors, which are in turn part of fixed materials. These transistors can be read to in tens of microseconds and written to in hundreds of microseconds, which compares very favourably with hard disks, whose latency for read/writes is usually 3-10 milliseconds, i.e. 30-3000 times slower (Chen et al., 2009).

Early-generation SSDs were found to have two unusual problems. The first problem is wear. Individual blocks within the SSD can only be written to perhaps 10000-100000 times before they are at risk of failing (Olson & Langlois, 2008). Consequently, a *wear-leveling* scheme is used which avoids writing data continually to the same place. Instead, a translation layer is used that keeps track of where the computer 'thinks' it is writing. As the computer writes to one block repeatedly, the SSD quickly translates each write to a new and less-used location; so that the computer is unaware it is actually storing the data in a different place each time. This significantly reduces the physical damage incurred by the blocks when a file is updated multiple times. To use an everyday analogy, you might imagine a hotel where the hotel manager renumbers the rooms each day to ensure that every room gets used equally, frustrating the actions of any guests who are trying to reserve a suite they prefer. Essentially, this process is similar to the error-mapping process used by hard drives, but it is now taking place for potentially every block of the whole SSD, with the consequence that the user has little idea where data is really being stored inside the drive's memory pages. The part of the controller that keeps track of where each block is being written is known as the Flash Translation Layer (FTL). SSDs generally ship with a percentage of ‘spare pages’ to facilitate this wear-leveling process. Figure 1 demonstrates this idea.

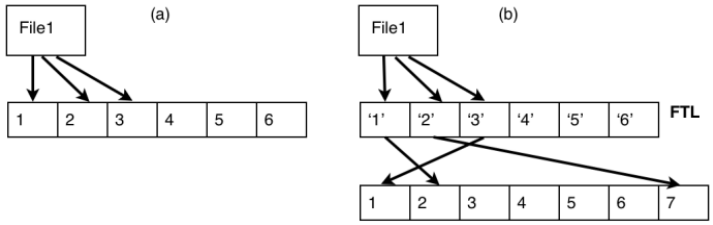

Figure 1: File1 is using blocks 1, 2, and 3 on the drive. (a) In the case of the hard drive, these blocks are generally used directly. (b) In an SSD, the FTL masks the real arrangement of data and hides the behaviour of the drive.

Wear-levelling, or alternatively, the extensive use of a drive’s storage capacity, can lead to a second problem: a significant slowdown in transfer speed. This occurs because flash technology generally requires blocks to be erased electronically before they can be used again. This is unlike the ‘write-over-old-data’ property of magnetic tapes and disks. The erase process is very slow compared to reading and writing, taking up to 10 milliseconds of time. Furthermore, for technological reasons, only entire blocks can be erased, not individual pages. As SSD drives fill up, even a change to a single byte may result in an entire block needing to be read/erased and rewritten, resulting in an overt slowdown in performance. Many manufacturers have therefore added routines to the drive controller chip that can pre-emptively act to ameliorate the 'reset' problem. One common strategy is known as Garbage Collection or ‘Self Healing’. The underlying philosophy is to cautiously identify areas that are not in use, and reset them as soon as possible. Various strategies can be used to achieve this. The key problem, however, is that the drive controller does not know which operating system it will be used with; consequently, it cannot be sure which parts of the drive storage will contain deleted and redundant data as opposed to meaningful data, since that is decided by the operating system's mechanisms for organising information (e.g. FAT32, NTFS, HFS+, and ext3 filesystems). However, the drive controller could note that any page which has already been copied elsewhere as part of a wear-levelling process is no longer needed. Consequently, it can be reset at any suitable opportunity. This is somewhat effective on a mostly-empty drive,

but as the drive gets increasingly full there is a tendency for almost every block to be in use. Worse still, the operating system does not normally tell the disk when it thinks a file has been deleted and is no longer needed. The drive controller chip may be left with the belief that all data stored on the disk is useful, even when much of the data is redundant because the original file metadata that linked to it has been deleted. As a result, there may be no convenient opportunity to reset many cells and drive performance may still be poor. For this reason, Samsung, a major provider of drive controllers, decided to develop an algorithm that would work exclusively with the NTFS filesystem. The algorithm (whose details have not been revealed by the company) is said to be capable of looking at the 'used/unused' aspects of an NTFS filesystem by examining the free space bitmap. Consequently, it is able to discover areas that have been marked as unused from the operating system's perspective. It then pre-emptively resets those blocks, making them ready for any forthcoming write requests. These SSDs are more akin to computers than 'dumb storage'; they react to commands from their host computer, but sometimes act under their own initiative.

We felt it might be possible that some of these Garbage Collection technologies could be triggered to activate in scenarios that could occur during physical capture, forensic analysis or validation processes, in the absence of explicit write commands from the computer (the marking of the file as deleted would be an event of the past). If garbage collection were to take place before or during forensic extraction of the drive image, it would result in irreversible deletion of potentially large amounts of valuable data that would ordinarily be gathered as evidence during the forensic process - we call this 'corrosion of evidence'. But this is only the first problem. The second problem is that any alteration to the drive during or after extraction may make validation of evidence difficult. Imagine the situation faced by a forensic analyst who takes an image of a drive (with checksum) in order to analyse it. After taking the image, the drive is powered up prior to validation. At this point, the disk's controller decides to carry out a phase of garbage collection and deletes part of its own data. When the forensic analyst's image is compared against the 'original' copy, it appears that the analyst has been working with a tainted copy, to which someone has artificially added some 'evidence'. Yet ironically, in this example it would be the original disk that has become tainted by its internal control chip modifying the disk contents. Potentially, though, without knowing when garbage collection is taking place, it is impossible to be certain that any copy taken at any time is authentic to the condition of the computer at or around the time of physical seizure of the apparatus.

We were unable to obtain any authoritative documentation on the behaviour of the garbage collector routines included in modern SSD drive controller chips, so we developed experiments that allow us to prove that it is possible for drives to modify themselves very substantially without receiving instructions to do so from a computer. The most important question we are exploring is whether SSDs can present entirely different evidence, compared to HDDs, when interaction with the drive has taken place in precisely the same way.

We also wanted to explore two other situations. Firstly, would an experienced human investigator be able to rebuild evidence using existing software tools after self-corrosion has taken place? Secondly, is it possible that a traditional forensics tool such as a physical write-blocker - which is regarded as a standard in terms of preservation of digital evidence - could still trigger or permit automated self-corrosion of the evidence on an SSD by the drive itself?

Recall that the SSD controller chip that runs the garbage collection routine resides *inside* the SSD, not within the main part of the computer. This should imply that placing a write-blocker between the computer and the disk could be expected to have no effect (or at least, no direct effect owing to the write-blocking aspect; it is impossible to guess what impact unusual patterns of communication might have upon controllers and garbage collection algorithms).

The following section details our experiments. Please keep in mind that developing controlled, reproducible experiments for SSDs that explore their behaviour is presently difficult due to the large number of variables that may potentially interact with the experiment, because of the lack of clear documentation. It is impossible to know which aspects of the computer, filesystem, or drive state are considered important in the view of the programmers writing the garbage collection routines. We have therefore attempted to describe our apparatus and experimental method in sufficient detail to allow these experiments to be as reproducible as possible. Some

new programs were written to allow the experiment to operate in an appropriate timescale (e.g. one of the authors produced a forensic sampling program, which evaluates and compares hundreds of samples taken from across the drive in the space of seconds, rather than reading the whole drive, which would take a substantial fraction of an hour. This allowed the behaviour of the disk controller to be observed in close to real time during the experiments, yielding interesting results). The code listings for the software produced for this paper are freely available in Appendix II at the end of the paper.

4. EXPERIMENTAL INVESTIGATION OF SSD & HDD BEHAVIOUR

4.1 Experiment 1: Do HDDs and SSDs preserve deleted data equally?

Purpose of experiment: This experiment examined whether SSDs continue to store data after deletion, (providing it is not overwritten) in the same manner as traditional magnetic hard drives.

Method of experiment: We filled each drive with files containing the word 'EVIDENCE', quick-formatted the drive (which resets the filesystem but does not delete the file contents), and then measured the extent to which the drive continued to contain data from the 'traditionally recoverable' data files.

Detail of experiment: The experiment was carried out as follows. The standard experimental machine and configuration was used (see Appendix I). The 64GB P64 Corsair SSD and the 80GB Hitachi HDD were tested. During each test, the drive was connected to the secondary SATA channel on the motherboard and the machine powered on. After login the drive was 'quick formatted' with the command *diskmgmt.msc*, producing a new single primary NTFS partition covering the entire drive. A 196KB template file was selected containing the word 'EVIDENCE' repeated 25000 times. The partition was filled with copies of the template file using the *Fill* program in MSDOS. The copied files were numbered 1.txt, 2.txt, 3.txt... ending when the filesystem would not allow any further files to be written (all logical blocks were full). Upon the drive being essentially filled with data, the fill program was stopped. The drive was then immediately quick formatted as before using *diskmgmt.msc*, and then immediately shutdown to a power-off state. From the time the drive was filled with data to the time of complete shutdown was approximately 1 minute, including the time to carry out the quick format. The machine was then left turned off for approximately 10 seconds.

The machine was then started up, and was logged into as soon as the login screen became available. The program *Go* was then started in the Cygwin environment. The *Go* program repeatedly ran a second program, *Comparer* which sampled data for approximately 3 seconds, with 5 second gaps between each set of samples. The overall statistics for the sample from comparer were recorded into a file automatically by the *Go* program. The *Go* program was started as soon as possible after login (approximately 1 minute from the time of power on) and ended 15 minutes later. *Comparer* sampled 10KB of data from 60 positions in the drive separated regularly by 1000MB (and 80 positions, in the case of the 80GB drive)[2]. This data was tested by *Comparer* to see what portion of the sampled bytes were 'zero bytes'. Experiment 1 was repeated three times in total to the SSD and HDD to determine whether the results occurred consistently and reproducibly.

Expected result: We expected to see both the SSD and HDD start with a very high fraction of non-zero data representing the 'pre-formatting' files. We expected the HDD to continue to hold this data without modification, as there is no process erasing or overwriting the old data. We originally had expected to see the SSD 'forget' data over 30-60 minutes as the Garbage Collection inside the SSD explored the disk and erased the unused parts of the disks by itself.

Actual result: The results found are shown in Figure 2. We were astonished to see that contrary to the informal forum comments we had noted, suggesting that the drive should be left for one hour (or in some cases, “overnight”) to carry out a partial garbage collection process, Garbage Collection consistently began around three minutes after power-on, and consistently wiped almost all the drive (according to the samples taken) in around three minutes. The traditional magnetic HDD results stood out in stark contrast, with the disk data

[2] The sampling configuration was tested to ensure that this method of sampling would be consistent with larger or more regularly taken samples. The numbers chosen here allowed the drives to be analysed several times per minute, which can be seen here to give substantial insight into the behaviour of the garbage collection system.

remaining unchanged, and thus available for forensic recovery.

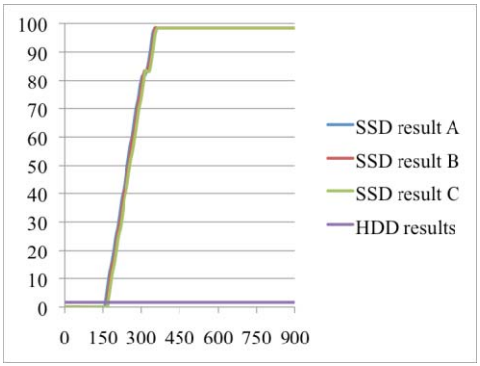

Figure 2: The X axis shows time and the Y axis shows the approximate percentage of the drive that has been zeroed out. In all three SSD runs, around 160 seconds from the log-in time (i.e. around 200 seconds from power-on), the SSD begins to wipe the drive. After approximately 300 seconds from log-in, the SSD consistently appears to pause briefly before continuing. 350 seconds after log-in, the SSD's pre-existing evidence data has been almost entirely wiped. In contrast, the HDD controller does not purge the drive.

Following the first completed run of Experiment 1, the 'post-wipe' drives were taken to be further examined by an experienced digital forensic examiner. The process and results are described in Experiment 2.

4.2 Experiment 2: A human forensic re-analysis of Experiment 1.

Purpose of experiment: To determine through conventional dead analysis using forensic imaging the extent to which an SSD and HDD configured in the same way have purged data following a 15-minute period connected to a SATA port without write instructions.

Method and detail of experiment: The drives were taken following the end of one run of Experiment 1 and reattached to the forensic analysis computer via a USB write-blocker, a Tableau Forensic SATA Bridge. The analysis computer and write-blocker were then powered-on and forensic images made of each drive using a conventional forensic imaging program, X-Ways Forensics version 15.7 SR2 to store the image on a 'clean' drive in accordance with conventional forensic practice. Examination of the image was then undertaken to inspect the data and metadata present on the drive.

Expected result: We expected to see the HDD had continued to hold the data from the previous filesystem with minimal modification, as there was no process erasing or overwriting the old data. We expected to see the SSD had 'forgotten' data because of the garbage collection inside the SSD exploring the disk and had erased the unused parts of the disks by itself.

Actual result: Forensic imaging was concluded 33 minutes after imaging began. The results found are shown in Tables 1 and 2. We observed that the garbage collection on the SSD had substantially erased data and metadata from the previous filesystem which was intact on the HDD. This resulted in the SSD drive being almost completely wiped of data. Evidence was found supporting the prior existence of only 1,090 files from the original 316,666 files present before the quick format. The magnetic HDD results stood out in strong contrast, with almost the entirety of the original file data being preserved without modification as expected, and thus fully available for forensic recovery. Evidence supporting the prior existence of 395,672 of the 395,673 files created during Experiment 1 was found. Evidence supporting the prior existence of a further 12 files relating to previous use of the HDD were found in the unallocated space.

Device **64GB Corsair P64 SSD**	**Files recovered**	**Description of recovered files**
1,064 deleted 'evidence' text files were recovered from the drive.	604	File name and temporal metadata recovered and file partially readable (size =195 KB) but could not be recovered as original text documents.
	460	File name and temporal metadata recovered but not readable (size =195 KB) and could not be reconstituted as original text documents.
An additional 26 deleted 'evidence' text files were carved from unallocated space.	23	File name and temporal metadata recovered and files readable (size =195 KB) but could not be reconstituted as original text documents.
	3	File name and temporal metadata recovered but not readable (size =195 KB) and could not be reconstituted as original text documents
Total files recovered	**1,090**	**Compared to 316,666 files present prior to quick format & experiment 1 study period.**

Table 1. Recovered deleted files from the 64GB P64 Corsair SSD.

Device **80 GB Hitachi Deskstar**	**Files recovered**	**Description of recovered files**
395,762 deleted 'evidence' text files were recovered from the drive	395,761	File name and temporal metadata recovered and files were readable (size =195 KB) and could be reconstituted as original text documents.
	1	File name and temporal metadata recovered but file not readable (size = 0 KB) and could not be reconstituted as an original text document.
12 deleted 'evidence' text files were carved from unallocated space.	12	File name present, no temporal metadata. Files were not readable and could not be reconstituted as original files. These entries were associated with the previous user of the disk and are disqualified from the experiment totals.
Total files recovered	**395,762**	**Compared to 395,673 files present prior to quick format & experiment 1 study period.**

Table 2. Recovered deleted files from the 80GB Hitachi Deskstar SATA drive.

Examination of recoverable files from SSD forensic image: 1,064 'EVIDENCE' files were extracted from the image. A comparison of the data content of these files with the control sample produced the following results.

i. Zero files were recovered fully intact.

ii. A single file was recovered containing 101,696 correct bytes of 200,000 (approximately 50% recovered). This was the maximum level of recovery achieved for any file.

iii. 5 further files were recovered 35% complete.

iv. The remaining 1,058 files had at least 82% damage.

v. 416 recovered files were completely empty (100% damage).

vi. The mean average % damage to the 1064 files was: 193,107,966 bytes out of 212,800,000 possible bytes, yielding 90.8% typical case damage to the evidence in the 'recoverable' files.

In summary: Evidence supporting the prior existence of only 0.34% of the files in the original filesystem survived a quick format followed by garbage collection (1,090 of 316,666). Zero complete files survived garbage collection. The file data content that survived garbage collection in total represented only 0.03% of the data that was originally written to the drive. It appears that the garbage collection that took place within a few

minutes during Experiment 1 had affected every file on the disk, had deleted metadata supporting the existence of 99.66% of files, and had prevented recovery of 99.997% of file data.

Examination of recoverable files from HDD forensic image: Essentially all of the files were suitable for extraction. A sample of 20MB of files (100 files) was selected for evaluation to determine the extent of damage to typical files. Of these, 100% of the files were intact with the original data. No damage to the data of any file was discovered. This is in line with what would normally be expected from a drive upon which a quick-format was performed.

4.3 Experiment 3: Is the experimental sampling itself affecting the result?

Purpose of experiment: To explore whether the regular data-sampling process used in Experiment 1 could have interfered with or influenced the behaviour of the SSD drive during the experiment. Recall that SSDs are a relatively untested medium as far as forensic analysis is concerned. Could the process of study itself be triggering the effect?

Method & detail of experiment: Here, the Corsair P64 SSD used in Experiment 1 was filled/formatted as in Experiment 1. However, an exploratory sample was not taken until 20 minutes after Windows had loaded.

Expected result: We expected that the sampling process would have no significant impact on the general behaviour of the disk, other than that it might perhaps have slowed down the garbage collection slightly.

Actual result: When read at the 20-minute mark, the SSD drive was discovered to have purged almost all of its data in the same manner as was seen in Experiment 1. This matches the expected result.

4.4 Experiment 4: Can write-blockers prevent self-corrosion?

Purpose & reasoning behind experiment: The previous experiments demonstrate that simply turning on the computer while an SSD drive is connected to a SATA channel can result in almost complete eradication of traditionally recoverable evidence from the drive as a consequence of a block deletion program running inside the drive controller itself, rather than because of deletion commands emanating from the host computer. However, we wished to explore what would happen if the drive were connected to e.g. a USB-based write-blocker. Would the existing results be reproduced exactly? On the one hand, one might expect the SSD to behave identically, since the data-destructive behaviour is being initiated from inside the drive shortly after power is made available. On the other hand, perhaps the garbage collector is being triggered following SATA initiation/configuration signals that would take place when connected normally to a computer. In that situation, garbage collection might not take place in the presence of a USB write-blocking bridge. No experiment can conclusively rule-in a write-blocker as a solution to the observed problem, as every drive and every write-blocker could interact in a completely different way. However, this experiment is capable of ruling-out write-blockers as a potential cure-all if evidence is lost that would traditionally have been preserved on a magnetic hard drive with a write-blocker.

Method & detail of experiment 4A: An initial experiment was designed as per Experiment 1 with the SSD drive. However, during the ten-second power-off period, the drive is disconnected from the SATA channel of the host computer, and connected to a write-blocker. The write-blocker is powered on. The write-blocker is then attached to the computer. The computer is turned on and *Go* is then run after logging into Windows.

Expected result: We expected the self-corroding behaviour should occur as before, but were not certain. We suspected the write-blocker's hardware might interact with the drive such that garbage-collection did not become initiated.

Actual result: This initial experiment structure was abandoned after the first runs indicated that garbage collection was not taking place; *Comparer* had reported the drive's contents were 99.94% non-zero data. Recall that we are attempting to discover if there are circumstances in which write-blockers are not the panacea they are believed to be. We speculated that perhaps the write-blocker's hardware might be affecting communication with this particular drive in other ways besides simply blocking write commands. A second variation of the experiment was designed to try to explore the matter further.

Method & detail of experiment 4B: A second experiment was designed based upon Experiment 4A. The write blocker was again connected immediately after the quick format and power-down. The experimental protocol is the same except that the computer is now left idle at the login screen for 20 minutes before login. Then, the user logs in and runs *Go* for 15 minutes.

Expected result: We expected the self-corroding behaviour should occur as before, but were unsure if the write-blocker might interact with the drive in such a way that the garbage-collection routine did not become initiated.

Actual result: Garbage collection takes place and the drive content is modified! However, the results are different from Experiment 1. Now, *Comparer* suggested on all three runs that the drive was repeatedly 18.74% zeroed-out. This compares with 0.06% zeroed-out if measurements are taken after the 'quick format' is completed or when following the 4A protocol. This is a fascinating result, and we believe it demonstrates *the first ever case of a drive having substantial amounts of evidence purged from it despite the use of a conventional hardware write-blocker*. This purging of data blocks from the drive clearly takes place without a request from the host computer. It occurs on a short timescale and in an experimental setup that is a realistic representation of a situation that could happen in the real world (the forensic analysis computer is turned on, but not logged into until 20 minutes later - e.g. phone call, coffee break). To confirm the result further, we ran *Comparer* again with ten times as many samples, and ten times as much data per sample, yielding 600 samples of 100kB regularly spaced across the drive every 100MB. The result from this more detailed comparison was in line with the faster sample: 14.8% erasure was discovered. We have so far been unable to discover why only one sixth of the drive is being wiped. However, we are confident it will be possible to find other experimental conditions, write-blockers, and drives that will allow greater degrees of self-corrosion of evidence. Furthermore, although this result is the first 'bad apple' for write-blockers, it sours the whole bunch in terms of the degree of confidence placed in this forensic technology.

Closing note: As we observed before, there are many types of SSD drives, many firmwares, and many approaches to garbage collection. There are also many forensic write-blocking tools and means of connecting these to a host computer. It is possible that certain combinations of drives, firmware, garbage collectors and write-blockers may combine to facilitate or prevent a particular hardware configuration from wiping the disk. This does not imply that e.g. a particular write-blocker will uniformly bring about or prevent the garbage collection process for a particular family of disks or for SSDs generally. Unfortunately, 'anything can happen' for a particular combination, and SSD disks, flash technologies, controllers and firmware are a fast-moving and unpredictable target.

5. INTERPRETATION AND DISCUSSION OF RESULTS

Overall, these results seem remarkable. Experiments 1 and 2 show clearly that SSDs do not behave in the same manner as HDDs, and are quite capable of essentially near-complete corrosion of evidence entirely under their own volition. In contrast, that same evidence is shown here to be almost entirely recoverable (both filesystem metadata and data) if the sample drive is a magnetic drive undergoing forensic analysis. Here, the 'EVIDENCE' files can be seen as a proxy for complex filesystem metadata or everyday data blocks.

In the case of the HDD we have seen that almost all files are preserved after the user runs a quick format. The sample taken indicates data in those files could later be recovered perfectly. In contrast, with the SSD we saw that shortly after a reboot, the entirety of the files were damaged and almost all were purged completely, including their filesystem metadata records. After only a few minutes of sitting idle, only a single file among 316,666 was even 50% recoverable; and only 0.03% of data was recoverable. The contrast is startling.

The timeframe and consistency with which data is purged is particularly astounding; the time taken to purge the data is only a small fraction of the time taken to write data all across the disk in the first place. The drive initiated block erasure without any trigger from the user or from an automatic program on the host computer, which seems to indicate some kind of timer-based system being used by the garbage collection inside the SSD.

It is particularly interesting to note that the garbage collection process was not interrupted by the sampling process that took place. This result is tremendously important in terms of validation of evidence. If the drive is purging data far faster than the analyst can extract it, and the process of purging can begin and continue while

the analyst is extracting the data, how can the analyst hope to capture a complete, frozen image of the disk that is representative of the disk state at capture time?

The consistency of the results between runs of the experiments was not unexpected. Filling a drive up in an essentially identical way with identical files should trigger an identical or near-identical response from a drive. We do not feel there is new information to be gained from repeating these experiments further ourselves.

Another interesting concern is the physical appearance of the drive while garbage collection is taking place. The drive does not emit any flickering light; nor does it whirr, vibrate, or click as a traditional magnetic drive might. It looks no different when it is powered on or when it is powered off and remains silent throughout the process. This means there is tremendous potential for digital evidence to be corroded in a physically undetectable manner.

The results showing that purging of evidence from a drive can take place despite the presence of a physical hardware write-blocker are especially important because they illustrate the dangers of placing too much confidence in forensic processes and technologies, which can become ineffective or misleading as technology progresses.

We end this brief discussion of results by presenting a thought experiment, consisting of real world scenarios that we feel would be hard to distinguish. They are themed around the outcomes seen in the experiments.

Scenario 1a/1b: An innocent member of the public decides to reformat their drive and reinstall Windows because of a virus or slow performance (alternatively, a criminal has a drive containing evidence of their activities and reformats it). They 'quick format' the drive, but then decide to make a cup of tea before continuing. Meanwhile, the SSD's controller chip analyses the new filesystem and determines that few of the disk blocks are in use. The SSD resets most of the data blocks to prepare them for use, purging all of the data that was previously on the disk. When police seize the computer a few minutes later, they find it to be almost completely empty of data. A forensic analyst later wonders: was there ever anything illegal there, and if so, did the suspect knowingly purge that illegal data from the drive?

Scenario 2a/2b: An innocent member of the public (or a criminal) quick formats an SSD or deletes their files, for innocent (or nefarious) reasons. Police seize the computer a few seconds later. Upon being connected to power, the drive begins to erase itself even while the forensic investigator is trying to read data from the drive. The forensic investigator suspects the existence of a ‘logic bomb’ intentionally put in place to prevent data from being used as evidence.

These scenarios show that the results of this paper open up serious concerns for police operations management, forensic analysis, and court argument as an interpretation is sought for the absence of data.

6. RECOMMENDATIONS AND GUIDANCE

We suggest that:

1. Solid-state drives of all types and data stored on such drives should be immediately and henceforth considered to be a 'grey area' as far as forensic recovery and legal validation are concerned until extensive studies have been made of drive and data behaviour.
2. Processes that are corrosive/erasive towards data marked for deletion may take place extremely suddenly, extremely quickly and completely automatically without human awareness or control.
3. Present-day evidence indicating 'no data' does not authoritatively prove that data did not exist at the time of capture.
4. Evidence of deleted data being permanently erased or partially corrupted is not evidence of intentional permanent erasure or corruption.

5. Cases where disk image checksums do not match at the end of the process of forensic analysis should be considered carefully to establish if the original or subsequent images could have been taken during or after a garbage collection process.
6. Past metadata and data blocks may be deleted without warning and without the opportunity to realise that they had existed at time of capture.
7. Garbage collection can occur either following file deletion or following (quick/full) formatting of a disk. 'Quick' formatting is therefore no longer necessarily distinct from 'full' formatting.
8. Formatting of disks is a normal and reasonable activity that an innocent person might choose to do e.g. to improve the performance of an SSD drive, to tidy up the disk etc. yet may completely eradicate evidence from a disk. Such eradication of evidence may occur within minutes.
9. We cannot guarantee previously deleted file data to be preserved on an SSD, regardless of whether the drive image was taken during a 'live' capture of evidence or following a 'dead' capture of evidence.
10. Drives can clearly self-modify their data after physical evidence has been gathered, despite best practice efforts by forensic analysts to prevent such behaviour using traditionally effective means such as write-blockers.
11. A software or hardware-based write-blocker does not protect against the drive's internal firmware choosing to wipe data from the drive.
12. The speed at which corrosion of digital evidence takes place should be expected to increase even further as garbage collection algorithms become more aggressive in cleaning up, and drives become faster, and more powerful controller chips become available.
13. We feel it would be an unwise investment of time for analysts to try to develop workaround procedures that operate against the drive controller behaviour specifically identified in these experiments. It will not be useful; even now, a new firmware has been released for the drive, and newer models are on the market, whose behaviour cannot be known a priori. SSDs are an exceptionally fast-moving target.
14. We think it would be an unwise investment of time to develop procedures for physical asset capture whereby operators attempt to distinguish SSDs from HDDs. This process is difficult due to the similar physical appearance of the drives and the need to gain access to the computer's internals. Also, each SSD might require unique and perhaps complex treatment to try to prevent the operation of garbage collection. Furthermore, it is futile - the present-day development of hybrid disks that incorporate both HDD and SSD technologies is likely to frustrate completely any effort to distinguish SSDs from HDDs at time of capture.
15. We think it would be unwise to assume that irreversible file erasure suggests intent to destroy evidence in cases where a defendant has quick-formatted a drive prior to police seizure. There are reasonable circumstances (viruses, slow performance, upgrading) that might cause normal and innocent people to quick format their drive without realising that garbage collection would cause the appearance of 'secure deletion'.
16. Breaking open the drive casing to try and disable/remove the drive controller and prevent a garbage collector from running would probably be a very significant technical challenge, given the extent to which the drive controller is bound to the data via the flash translation layer, and given the wide diversity of drives, controllers and memory that exist.
17. Running a live analysis might allow a running garbage collector to delete data unhindered; running a dead analysis might cause a garbage collector to activate in the forensics lab. There is no simple answer to this problem.
18. It will probably be impossible to legislate against garbage collection.

19. It is possible that the issues found in this paper will later come to affect USB flash drives as well; having invested in technology that makes NAND flash systems run much faster, it would be very peculiar if manufacturers failed to adopt it across their product range. The increasing availability of high-speed interfaces for portable media such as USB3 makes it likely that we will see more of these complex, evidence-corroding drive controllers in portable USB drives in future.
20. The natural state of digital storage is not to 'preserve deleted data' as magnetic drives have done for the past few decades. Future physical storage technologies can be expected to purge deleted files to improve read and write speeds, if technology allows such a purge to be performed quickly.
21. It seems very unlikely that international manufacturers can be collectively persuaded to provide a universal mechanism to disable garbage collection; if anything, garbage collection is likely to become even more aggressive. Perhaps the genie is already out of the bottle?

7. THE FUTURE OF DIGITAL STORAGE?

Here, we document numerous concerns for the future that are still emerging.

Aggressive garbage collection: In the last few years, there has been an arms race in terms of SSD controller technologies, which has resulted in a proliferation of controller chips and firmware programs. The established trend is to reset pre-emptively no-longer-needed flash storage cells in an increasingly aggressive manner. Consequently, even in the absence of a computer, it is potentially possible for a powered (but disconnected) SSD drive to erase 'deleted files' or even erase entire quick-formatted disks. It is also possible that future controllers will engage in 'aggressive garbage collection' activities even while new data is being written to the drive providing it does not interrupt the overall performance of the drive. It is hard to see what actions could be taken by a forensic investigator to ascertain if a given, unknown drive will proceed to erase data permanently upon being supplied with power during data extraction.

Faster SSDs: In these experiments, it was shown that an SSD could wipe its own data permanently (without explicit instructions to do so) in the space of 2-3 minutes. Yet the trend is for faster controllers and faster memory, such as recent "DDR" toggle-mode flash (Samsung Electronics, 2010). Consequently, the window of opportunity for drives to wipe themselves completely is getting larger, and it would not be surprising to see drives with considerably faster or more parallelised 'reset' processes in future, given that this is perceived as one of the few remaining problems for SSDs today.

Hybrid SSDs: Presently, magnetic hard disk manufacturers are integrating SSDs and HDDS into single drives that use the SSD as a gigantic cache for data (Beeler, 2010). Essentially this extends the idea of the DRAM cache that is already built into SSDs. However, such integration might mean that some of the issues observed in this paper could begin to affect traditional hard drives, for at least a sizeable fraction of the data being stored.

Host-side garbage collection: TRIM is a new standard for computer-to-disk (ATA) communication, which allows operating systems such as Microsoft Windows 7 to inform a drive that an area of data is no longer being used and can/should be erased (Stevens, 2010). As TRIM enters widespread use through the uptake of Windows 7 and other modern TRIM-supporting operating systems, recovery of deleted files and old metadata will become extremely difficult, if not impossible. The difference between TRIM and the results found here is that TRIM commands are issued by the operating system at the time a file is deleted; it is not a pattern of behaviour that is initiated from within the drive itself in the absence of instructions from the host computer. Furthermore, TRIM is an action that is typically undertaken immediately, upon the data associated with a single file, at the time of intentional direct individual file deletion. In this paper we have looked at actions that take place on the scale of the entire drive, and that take place subsequent to user actions and in the absence of direct deletion of individual files.

8. CONCLUSIONS

The results found in this paper may have significant implications for legal matters involving digital evidence,

most especially in those cases where digital data is alleged to have been deleted intentionally or deliberately permanently wiped by a defendant. Given the pace of development in SSD memory and controller technology, and the increasingly proliferation of manufacturers, drives, and firmware versions, it will probably never be possible to remove or narrow this new grey area within the forensic and legal domain. It seems possible that the golden age for forensic recovery and analysis of deleted data and deleted metadata may now be ending.

APPENDIX I: EXPERIMENTAL ENVIRONMENT

Physical computer:

Dell Precision Workstation R3400. Bios A01. 3GB RAM, Intel Q6600/2.4Hz, 232GB SATA C: HDD used as system drive. Network adapter present but not connected during experiments.

Software:

Windows XP V5.1 (Build 2600.xpsp_sp3_gdr.100216-1513 : Service Pack 3).

No antivirus or remote desktop management programs (or similar). System restore disabled on all drives. Indexing service disabled in 'Services'.

Cygwin version CYGWIN_NT-5.1 1.7.6(0.230/5/3) with bash, awk, nano, cmp, bc as of 2010/08/16 16:00. DOS command prompt with edit.exe.

Sample HDD: Hitachi Deskstar HDS728080PLA380 SATA. 80GB. Rev A00.

Sample SSD: Corsair P64 CMFSSD-64GBG2D. 64GB. Rev. VBM19C1Q.

SATA drives were connected on secondary SATA channel as D: drive during experiments. The disks were not used as system drives. Drives were configured as single NTFS partition "basic drives" with default parameters.

Write-blocker: Tableau Forensic SATA/IDE Bridge Model T35e with USB. The device was validated as follows. The authors attempted to write to drives or format them, and were unable to. The authors calculated hashes on a regular hard drive with and without the device and obtained the same result.

APPENDIX II: SOFTWARE CODE LISTINGS

These programs were written by Graeme B. Bell and are provided to support your own experimentation. You may freely use them and modify them, at your own risk, providing you acknowledge the author and this article in your writing.

template.sh : Cygwin script. Type ‘./template.sh’ to run.

Creates a template file consisting of the word EVIDENCE written 25000 times.

```
#!/bin/bash
rm ~/template
for i in {1..25000} ; do echo -ne "EVIDENCE" >> ~/template; done
```

fill.bat : BAT file for the DOS command environment. Type ‘fill’ to run.

Fills the D: drive with copies of the template file named in ascending order.

```
@echo off
echo Starting
@SET count=1
:while
@IF %count% GTR 1000000 (GOTO wend)
copy template D:\%count%.txt > NUL
@set /a count=count+1
@GOTO while
:wend
echo Finished making files
```


go.sh : Cygwin script. Type "./go.sh" to run.
Runs the comparer script with 5 second delays, comparing disk vs. zeroes.

```
#!/bin/bash
while true; do ./comparer.sh /dev/sdb /dev/zero >> experiment-out
sleep 5
done
```

comparer.sh : Cygwin script. Needs cmp/awk/bc. "./comparer A B" to run.
Compares samples from 2 files (or streams) at fixed intervals over a given size.

```
#!/bin/bash
samplesize=10000;    # bytes to sample
size=80;   # size of disk, gigabytes. PLEASE REMEMBER TO SET THIS.
gap=1000;   # gap between samples in megabytes.

echo; echo -n "Comparing $1 and $2 at: " ; date
echo -n > sample

for (( P=0; P<size*1000; P=P+gap )); do  # "P" will be position.
cmp -bl $1 $2 -n $samplesize -i"$P"MB | wc -l >> sample
totalsamples=$((totalsamples+samplesize))
samplecount=$((samplecount+1))
done

difftotal=`cat sample | awk 'BEGIN{t=0}{t += $1}END{print t}'`
echo "Different bytes: $difftotal of $totalsamples"
percent=`echo "scale=2; 100-(($difftotal*100)/$totalsamples)" |bc`
echo "Volumes are $percent% identical "
echo -n "Comparison ended at: "; date
echo "Sampled: $samplecount samples of size $samplesize over $size GB"
```


ACKNOWLEDGEMENTS

We thank the School of IT at Murdoch University for experimental apparatus.

AUTHOR BIOGRAPHIES

Dr. Graeme B. Bell holds a BSc(Hons) 1st Class and a PhD in Computer Science from the University of St. Andrews, UK. He has taught and has carried out research within Computer Science at the University of St. Andrews, UK, Ming Chuan University, Taiwan, and Murdoch University, Australia. His interests include digital forensics, steganography, steganalysis, web security, robotics and artificial intelligence.

Mr. Richard Boddington holds a BSc(Hons) 1st Class and is undertaking PhD research in digital evidence validation at Murdoch University, Australia where he teaches and researches information security and digital forensics. He has a police and security intelligence background and provides forensic analysis and expert testimony for the legal fraternity in a range of civil and criminal cases.

REFERENCES

Ashcroft, J. (2001). Electronic crime scene investigation: A guide for first responders. Washington: U.S. Department of Justice.

Beeler, B. (2010). Seagate Momentus XT Review. Storage Review, May 24, 2010. http://www.storagereview.com/seagate_momentus_xt_review

Berg, E. C. (2000). Legal ramifications of digital imaging in law enforcement. Forensic Science Communications, 2(4).

Boddington, R., Hobbs, V. J., & Mann, G. (2008). Validating digital evidence for legal argument. Paper presented at the SECAU Security Conferences: The 6th Australian Digital Forensics Conference, Perth, WA.

Caloyannides, M. A. (2001). Computer forensics and privacy. Norwood, Minnesota: Artech House.

Carrier, B. (2005). File system forensic analysis. Upper Saddle River, New Jersey: Addison-Wesley.

Carrier, B., & Spafford, E. H. (2003). Getting physical with the digital investigation process. International Journal of Digital Evidence.

Carlton, G. H., & Worthley, R. (2009). An evaluation of agreement and conflict among computer forensics experts. Proceedings of the 42nd Hawaii International Conference on System Sciences.

Chen, F., Koufaty, D. A., & Zhang, X. (2009). Understanding intrinsic characteristics and system implications of flash memory based solid state drives. In Proceedings of SIGMETRICS '09 (Seattle, WA, USA, June 15 - 19, 2009).

Drossel, G. (2007) Solid-state drives meet military storage security requirements. Military Embedded Systems, White Paper, OpenSystems Publishing, 2007.

Edwards, K. (2005). Ten things about DNA contamination that lawyers should know. Criminal Law Journal, 29(2), 71 - 93.

Ekker, N., Coughlin, T., & Handy, J. (2009). Solid State Storage 101 – An introduction to Solid State Storage, SNIA White Paper, January 2009. http://www.snia.org/apps/group_public/download.php/35796/SSSI%20Wht%20Paper%20Final.pdf

Etter, B. (2001). The forensic challenges of e-crime. Australasian Centre for Policing Research, 3(10), 1-8.

Flusche, K. J. (2001). Computer forensic case study: Espionage, Part 1 Just finding the file is not enough! Information Security Journal, 10(1), 1 - 10.

Ghosh, A. (2004). Guidelines for the management of IT evidence. Paper presented at the APEC Telecommunications and Information Working Group 29th Meeting. From http://unpan1.un.org/intradoc/groups/public/documents/APCITY/UNPAN016411.pdf

Hughes, G. F. (2002, 7 November 2002). Wise drives. Spectrum, 39, 37 - 41.

Janes, S. (2000). The role of technology in computer forensic investigations. Information Security Technical Report, 5(2), 43 - 50.

Kenneally, E. E., & Brown, C. L.T. (2005). Risk sensitive digital evidence collection. Digital Investigation, 2(2), 101 - 119.

Losavio, M., Adams. J., & Rogers, M. (2006). Gap Analysis: Judicial experience and perception of electronic evidence. Journal of Digital Forensic Practice, 1, 13 - 17.

Odagiri, H., Goto, A., Sunami, A., & Nelson, R. R. (2010). Intellectual Property Rights, Development, and Catch Up: An International Comparative Study. Oxford University Press. pp. 224–227. ISBN 0199574758.

Olson, A. R., & Langlois, D. J. (2008) Solid State Drives (SSD) Data Reliability and Lifetime. National Media Lab White Paper April 2008. http://www.imation.com/PageFiles/1189/SSD_Gov_DataReliability_WP.pdf

Samsung Electronics. (2010). Samsung Introduces High-speed 512GB SSD Utilizing New Toggle-mode DDR NAND Memory, Samsung Electronics. Jun 17, 2010. http://www.samsung.com/us/aboutsamsung/news/newsIrRead.do?&news_ctgry=irpublicdisclosure&news_seq=19483&page=1

Stevens, C. E. (2010). TRIM – DRAT / RZAT clarifications for ATA8-ACS2 (Draft). Working Draft Project American National Standard. http://t13.org/Documents/UploadedDocuments/docs2010/e09158r2-Trim_Clarifications.pdf

Yasinsac, A., Erbacher, R. F., Marks, D. G., Pollitt, M. M., & Sommer, P. M. (2003). Computer forensics education. IEEE Security & Privacy, 1(4), 15 - 23.

