Technical Analysis of Behaviour of Overwriting Algorithms when Applied to Self-Optimized SSHD Drives

Dr. Andrew J. C. Blyth, PhD

ADISA Research Centre (ARC), Thrales End Business Centre, Thrales End Lane, Harpenden AL5 3NS

E: andrew.blyth@adisa.global

1.0 Introduction

Data protection legislation required that all technical and organisational measures are taken to protection personal data [8,9,10]. This principle applies throughout the life of the data. Research studies have shown that a particular weakness with data protection exists at the end of life for a data bearing physical artefact [11]. Standards such as NIST 800-88 [12] work by over writing data to all user space areas on a Host Disk Drive (HDD) [13]. The ATA protocol standard views a HDD as an array of blocks of data from zero to a maximum value (MAX_{LBA}) [2] where each block is called a logical block address (LBA).The algorithm that all standards use is to either write zero, one, or random data to every logical block address (LBA) on a hard drive. Commercial tools expand this definition to include DCO and HPA areas of a HDD. Current Host Disk Drive (HDD) technology has merged the Electromechanical hard dives with Solid State to include through-put and provide fast IO. Solid State technology works by providing a set of NAND Flash memory cells and a series of algorithms governing how data on the NAND Flash memory cells is management [1]. Data erase standards have not evolved in the past twenty years, thus the question relating to the ability of standard such as NIST 800-88 [12] to erase all data include the Solid State component remains.

```
begin
    count ← 0
    data ← {}
    block_size ← ata.identify.blocksize
    max_lba ← ata.identify.maxlba
    while count ≠ block_size do
```

```
        data = data ∪ {0}
        count ← count + 1
    done
    for each lba ∈ {0..max_lba) do
        write(data,lba)
    done
end
```

The function `ata.identify.maxlba` return the max number of logical block address on the device and the function `ata.identify.blocksize` return the block size of a logical block address. For most disks the standard block size is `512` bytes. The function `write(data,lba)` writes the block of data (`data`) to the logical block address (`lba`). The objective this report is to examine the behaviour of software overwriting algorithms when applied to Solid-State Hybrid Drive (SSHD). A number of data recovery methods will be used to explore the capabilities of COTS and Commercial data sanitization tools and how they perform when applied to SSHD. The following is the data sanitization overwriting algorithm for writing zero's to every logical block address on a hard drive that will be evaluated via standard open source tools. The following is the list of data recovery methods that will used to perform the evaluation.

- Non-Invasive/Non-Destructive - Commercial Forensic Tools such as Encase [4] and Access Data FTK [5] and open source tools [1].
- Invasive/Non-Destructive - Commercial Data Recovery Tools such as PC3000 UDMA/SSD and Salvation Data.
- Invasive/Destructive - Chip-Off Data Recovery Tools such as PC3000 Flash and VNR, and attack methods such as JTAG and SPI.

The basic experimental method applied follows that of the ADISA Product Claims testing process which involves structured/known-data being written to every logical block address on the SSHD via a defined and repeatable method. The media will be imaged and then a data sanitization method shall be executed on the media. A second image will be taken and the two images e analysed using the data recovery methods listed above. A second task assessed during this study is to explore how the operating system, Given the nature of the interface, identifies and accesses an SSHD.?

2.0 Solid-State Hybrid Drive Architecture

Solid-state hybrid drive (also known by the initialism SSHD) refers to products that incorporate a significant amount of NAND flash memory into a hard disk drive (HDD), resulting in a single, integrated device. The term SSHD is a more precise term than the more general hybrid drive, which has previously been used to describe SSHD devices and non-integrated combinations of solid-state drives (SSDs) and hard disk drives [1,3]. The fundamental design principle behind SSHDs is to identify data elements that are most directly associated with performance (frequently accessed data, boot data, etc.) and store these data elements in the NAND flash memory. This architecture has been shown to be effective in delivering significantly improved performance over the standard HDD. In Figure-1 we can see the basic SSHD architecture. The disk controller functions to decode the ATA commands and make use of a standard cache to store data being read from the HDD. The Flash controller functions to manage the Flash memory and controls the disk motor controller. The Flash memory functions as a buffer for the Disk controller storing boot data, writes to the hard drive.

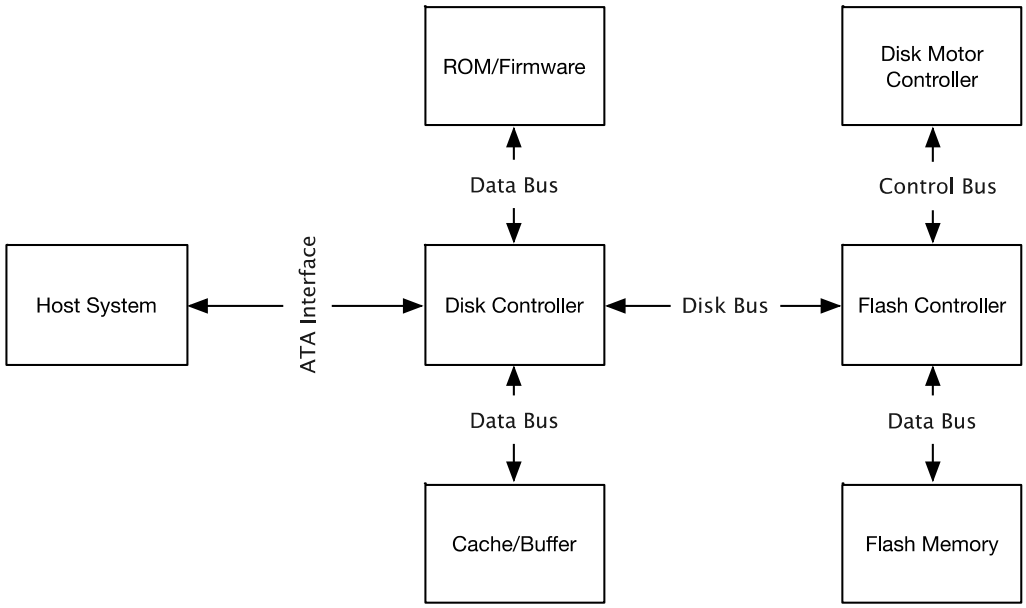

Figure 1 - Basic SSHD Architecture

The basic principle behind an SSHD is that it functions as a single device with an SATA interface. As far as the host computer system is concerned, the SSHD functions as a single ATA device and conforms to the ATA technical specification [2]. For the purposes of this study a Seagate Laptop ST500LM000 SSHD has been selected as the test subject. The Seagate Laptop ST500LM000 SSHD is a standard SSHD drive that complied with the SSHD architecture and ATA protocol standards.

Making decisions about which data elements are prioritized for NAND flash memory is at the core of SSHD technology. The SSHD using the NAND component of the drive as a buffer to which output is written to by the controller. The aim is for the NAND component of the SSHD drive to function as a Cache. SSHD can operate in one of two modes:

- Self-optimized mode
    - In this mode of operation, the SSHD works independently from the host operating system or host device drives to make all decisions related to identifying data that will be stored in NAND flash memory. This mode results in a storage product that appears and operates to a host system exactly as a traditional hard drive would.
- Host-optimized mode (or host-hinted mode)
    - In this mode of operation, the SSHD enables an extended set of SATA commands defined in the so-called Hybrid Information feature, introduced in version 3.2 of the Serial ATA International Organization (SATA-IO) standards for the SATA interface. Using these SATA commands, decisions about which data elements are placed in the NAND flash memory come from the host operating system, device drivers, file systems, or a combination of these host-level components. [2]

For the purpose of this investigation we will focus upon SSHD that function in Self-optimized mode, and in particular we will investigate the Seagate Laptop ST500LM000 SSHD. The Seagate Laptop ST500LM000 SSHD is a standard SSHD and its behaviour is indicative of all Self-optimized SSHD's.

Technical Analysis of Seagate Laptop SSHD

The Seagate Laptop ST500LM000 SSHD (Model ST500LM000-1EJ162), has 16,383 cylinders, 16 heads and 63 sectors. We can identify the drive's parameters using the `hdparam` tool.

```
$ hdparam -I /dev/sdb
/dev/sdb:
ATA device, with non-removable media
      Model Number:       ST500LM000-1EJ162
      Serial Number:      W3705BXW
      Firmware Revision:  DEM3
      Transport:          Serial, SATA 1.0a, SATA II Extensions, SATA Rev 3.0
. . . .
Configuration:
      Logical           max   current
      cylinders   16383 16383
      heads       16    16
      sectors/track     63    63
      --
      CHS current addressable sectors:    16514064
      LBA    user addressable sectors:   268435455
      LBA48  user addressable sectors:   976773168
      Logical  Sector size:                   512 bytes
      Physical Sector size:                  4096 bytes
      Logical Sector-0 offset:                  0 bytes
      device size with M = 1024*1024:      476940 MBytes
      device size with M = 1000*1000:      500107 MBytes (500 GB)
. . . .
```

Analysis of the above shows that the drive is identifying itself as a standard HDD. The next test that can conduct for on the SSHD is to detect the presence of a HPA. The following shows that the HPA is disabled and thus not present on the SSHD.

```
$ hdparam -N /dev/sdb
/dev/sdb:
 max sectors   = 976773168/976773168, HPA is disabled
\end{lstlisting}
We can detect the present of a the DCO via the following:\\
\begin{lstlisting}[frame=single]
$ hdparam ---dco-identify /dev/sdb

/dev/sdb:
DCO Checksum verified.
DCO Revision: 0x0002
```

```
The following features can be selectively disabled via DCO:
      Transfer modes:
            mdma0 mdma1 mdma2
            udma0 udma1 udma2 udma3 udma4 udma5 udma6
      Real max sectors: 976773168
      ATA command/feature sets:
            SMART self_test error_log security HPA
. . . .
```

The ATA interface supports LBA-48 commands and MAX_{LBA} is 976.773,168. As far as Technical components are concerned, the Seagate Laptop ST500LM000 SSHD makes use the following items:

- A LSI B69002V0 drive controller. The motor is controlled by a Texas Instruments SH6966 motor driver.
- A WinBond 25Q80BW16 64MB DDR2 IC acts as the cache for the drive that holds the firmware. This IC is of type 25Q80, with BW packaging.
- To manage flash memory, an eASIC/Seagate 50415 is utilized.
- The 8GB of Toshiba TH58TEG6D2HBA46 BGA/132 NAND flash is present for the "solid state" portion of the SSHD.

The Seagate Adaptive Memory acts as a large cache for frequently accessed data to increase performance, such as booting Windows or loading programs. From the user's perspective, the caching operates in the background, and is designed to work seamlessly and be invisible to the user. Analysis of the PCB showed that from the perspective of data recovery there are two memory integrated circuits worthy of investigation are:

- SPI FLASH - WINBOND / 25Q80BW16
- NAND Flash Solid State Memory - TOSHIBA / TH58TEG6D2HBA46.

The Bench Mark

To create a single of bench mark against which all data sanitization methods could be evaluated, a set of Seagate Laptop ST500LM000 drives had data placed on them. Each drive was connected to a Unix Machine running CentOS 6.9. The mount point for the ST500LM000 was identified using the following:

```
[93531.438391] scsi 2:0:0:0 Direct-Access     ATA      ST500LM000-1EJ16 SM15 PQ: 0
ANSI: 5
[93531.438602] sd 2:0:0:0 Attached scsi generic sg1 type0
[93531.438639] sd 2:0:0:0 [sdb] 976773168 512-byte logical blocks: (500 GB/465 GiB)
[93531.438642] sd 2:0:0:0 [sdb] 4096-byte physical blocks
[93531.438886] sd 2:0:0:0 [sdb] Write Protect is off
[93531.438889] sd 2:0:0:0 [sdb] Mode Sense: 00 3a 00 00
[93531.439030] sd 2:0:0:0 [sdb] Write cache: enabled, read cache: enabled, doesn't
support DPO or FUA
[93531.439748]  sdb: sdb1
[93531.440366] sd 2:0:0:0 [sdb] Attached SCSI disk
```

And then using the follow a fixed string of `0x5A, 0x5A, 0x5A, 0x5A, 0x5A, 0x5A, 0x5A, 0x5A` was written to every LBA on the device. Please note that this method for placing data on the device does not write data to the DCO and HPA, nor does it write data to the firmware.

```
$ yes "ZZZZZZZZ" > /dev/sdb
```

Data is placed on the device via the following set of LBA 48 Mode write commands.

- The WRITE SECTOR(S) command (OP Code: 0x30h)
    - This command writes from 1 to 256 logical sectors as specified in the Count field. A count of 0 requests 256 logical sectors.
- The WRITE SECTOR(S) EXT command (OP Code: 0x34h)
    - This command is mandatory for devices that implement the 48-bit Address feature set. This command writes from 1 to 65,536 logical sectors as specified in the Count field. A sector count value of 0 requests 65,536 logical sectors.

To validate that data had been written correctly to every Logical Block Address on the device Encase [4] and FTK [5] where used to image the SSHD and check that on the string `0x5A, 0x5A, 0x5A, 0x5A, 0x5A, 0x5A, 0x5A, 0x5A` was present.

3.0 Tool Technical Analysis

3.1 Tool: Open Source

To perform data erasure on the SSHD using open source tools the `dd` command was utilized. The following command was used to place zero's on the user space on the SSHD.

```
$ dd if=/dev/zero of=/dev/sdb bs=16M
```

The process for erasing the data on the SSHD was that the driver was connected to the UNIX system and the above command was executed.

Method: Non-Invasive/Non-Destructive

The Non-Invasive/Non-Destructive method for analysing the SSHD is to forensically image the device using tools such as Encase [4] and FTK [5]. In addition, a set of open source tools such TAFT [7].When both of these techniques where applied to the SSHD they identified nothing but zeros on the User Data area (0 to MAX_{LBA}) on the drive.

Method: Invasive/Non-Destructive

This method involves using standard data recovery tools to extract data from the device. The following data recovery methods are executed:

- The first method is data extraction via reading data from 0 to MAX_{LBA}. This method involves the data recovery tools having direct access to the ATA bus and generating/executing their own ATA commands.
- The second method using standard data recovery and a serial connection to the device to read the firmware located on the device.

Investigation of the SSHD using tools such as PC3000, etc., showed that the NAND Flash memory element of the SSHD is not visible, and thus the SSHD functioned as a single drive, all that was visible was the Firmware, SMART logs and User Data from 0 to MAX_{LBA}. Both methods listed above showed that there was no residual data left in either the Firmware/SMART-Logs and the User Data area from 0 to MAX_{LBA}.

Method: Invasive/Destructive

The Invasive/Destructive method for analysing the SSHD is to physically remove the `WINBOND / 25Q80BW16` and `TOSHIBA / TH58TEG6D2HBA46` integrated circuits and then using advanced data recovery tools to attempt to read them. All such tests yield negative results, in that no test data that had been written to the drive could be located.

4.0 Overwriting Conclusions

The technical analysis of the SSHD drives architecture (See Figure 1) shows that the SSD component of the SSHD drive is functioning as an inline buffer to the HDD component of the SSHD. Thus for both the Windows and Unix/Linux operating systems perspective the SSD component is not visible to the operating system. From a data recovery and computer forensic perspective, the act of writing controlled data from Zero (0) to MAX_{LBA} purges the buffer and overwrites the data every element/LBA of the SSHD. Thus rendering the data unrecoverable from the device. While it is true that no data could be recovered from the two integrated circuits `WINBOND/25Q80BW1` & `TOSHIBA/TH58TEG6D2HBA46`, it should be noted that that open source data erasure tools do not remap the drive in-terms of resetting the P/G lists. Thus if an LBA was to fail while holding any test data, the open source tools would not erase this data and thus technically data could still be present on the SSHD.

5.0 References

1. Rino Micheloni, Alessia Marelli and Kam Eshghi***, Inside Solid State Drives (SSDs)***, Springer, 2014.
2. The Serial ATA International Organization (SATA-IO), ***Serial ATA Revision 3.2 Specification***, Feb 2016.
3. Rino Micheloni, ***Solid State Drives (SSDs) Modeling: Simulation Tools & Strategies***, Springer, 2017.
4. Suzanne Widup, ***Computer Forensics and Digital Investigation with EnCase Forensic V7***, McGraw-Hill Education, 2014.
5. Fernando Carbone, ***Computer Forensics with FTK***, Packet Publishing, 2014.
6. Altheide and Harlan Carvey, ***Digital Forensics with Open Source Tools***, Syngress, 2011.
7. Arne Vidstrom, ***Computer Forensics and the ATA Interface***, Technical Report, Swedish Defence Research Agency, FOI-R--1638--SE, Feb 2015.
8. Peter Carey, **Data Protection: A Practical Guide to UK and EU Law**, OUP Oxford, 4th Edition, 2015.
9. Andreas Linder (Eds), **European Data Protection Law: General Data Protection Regulation 2016**, CreateSpace Independent Publishing, 2016.
10. Alan Calder, **EU GDPR: A Pocket Guide**, IT Governance Publishing, 2016
11. Jones, A., Dardick, G., Davies, G., Sutherland, I., Valli, C., Dabibi, G., **The 2009 Analysis of Information Remaining on Disks Offered for Sale on the Second Hand Market**, Proceedings of the 8th Australian Digital Forensics Conference, 2010.
12. Kissel, Scholl and Skolochenko, **Special Publication:800-88 - Guidelines for Media Sanitization**, Computer Security Division, Information Technology Laboratory. NIST Report 800-88, 2006, http://dx.doi.org/10.6028/NIST.SP.800-88r1
13. Abdulllah Al Mamun, GuoXiao Guo and Chao Bi, **Hard Disk Drive: Mechatronics and Control**, CRC Press, 2007.