GANGRENE: Exploring the Mortality of Flash Memory

Robert Templeman[1,2] and Apu Kapadia[1]

[1]Indiana University Bloomington
[2]Naval Surface Warfare Center, Crane Division

Abstract

Flash memory is used for non-volatile storage in a vast array of devices that touch users at work, at home, and at play. Flash memory offers many desirable characteristics, but its key weakness is limited write endurance. Endurance limits continue to decrease as smaller integrated circuit architectures and greater storage densities are pursued. There is a significant body of published work demonstrating methods to extend flash endurance under normal use, but performance of these methods under malicious use has not been adequately researched.

We introduce GANGRENE, an attack to accelerate wear of flash devices to induce premature failure. By testing a sampling of flash drives, we show that wear can be accelerated by an order of magnitude. Our results offer evidence that vendor-provided endurance ratings, based on normal use, ignore this underlying vulnerability. Because of the high penetration of flash memory, the threat of such attacks deserves attention by vendors and researchers in the community. We propose recommendations and mitigations for GANGRENE and suggest future work to address such vulnerabilities.

1 Introduction

Flash memory has achieved virtual ubiquity with a presence in a diversity of products including, but not limited to: computing devices, industrial controllers, automobiles, and medical devices. Many people use flash in a basic form as transportable media to move and archive files. Other instantiations, including phones and tablets, integrate flash as the primary mass-storage element. Thus it is increasingly the case that flash devices are responsible for the storage of personal and/or critical information and our reliance on this technology is increasing.

Finite flash write endurance. While flash has desirable characteristics, there exists a unique limitation in its finite write endurance. Each write-and-erase cycle (WE cycle) in a flash device requires a high-voltage to tunnel electrons across an oxide insulation [12]. Repeated operations degrade this oxide to the point that the memory becomes unusable. There are three primary flash technologies: SLC (single-level cell), MLC (multi-level cell), and TLC (three-level cell). These three categories of flash vary in both their storage density and write endurance, these two properties being inversely related. The typical write endurance for flash ranges from as few as 750 to 100,000 WE cycles.[1] Given the same footprint, MLC flash stores 2-bits per cell (quantization of four voltage levels) where SLC flash stores one bit per cell (represented by two voltage states). This increase in storage density comes at the expense of endurance ratings that are one order of magnitude lower. The difference between MLC and TLC technology is similar.

Extending flash write endurance. The appeal of greater storage density results in a preponderance of products that contain flash devices with low endurance ratings. Many methods have been developed to extend the effective lifetime of flash-containing devices. The goal of such mitigations is to reduce the number of oxide-degrading erase-and-write operations to the flash memory. This reduction is enabled by a universally used architecture in which indirect addressing is managed by a flash controller. Flash controllers are commonly application-specific integrated circuits (ICs) that are the interface between raw flash and interfaces including USB (universal serial bus), Serial Attached SCSI (SAS), and Serial Advanced Technology Attachment (SATA). Prior work details over-provisioning, wear-leveling approaches [5], garbage collection methods [1, 7], data deduplication schemes [11], and caching [15] among other efforts to maximize performance and usability of flash-based systems. In all cases commercially available

[1]AnandTech: Understanding TLC NAND. `http://www.anandtech.com/show/5067/understanding-tlc-nand/2`

flash controllers provide no details of the internal mechanisms that are used to extend life, as such details remain the competitive advantage in a crowded market space. Accordingly, we treat flash devices as black boxes.

Attacking flash write endurance. We seek to assess the impact of storage devices with limited write endurance through the lens of security. Existing papers and vendor materials address flash device longevity only under normal usage scenarios [8] (Section 6 references more of these methods). Prior work has acknowledged a potential for exploiting the WE cycle endurance limitation by malicious attack [2, 13], but there has been no work focused on such attacks. No existing published work has researched the worst case rates of degradation that are possible and the feasibility of attacks on flash device lifetime. We hope to spur such research. We demonstrate that vendor endurance claims must be used with caution and provide possible defenses against attacks on write endurance.

Our contributions:

- **We explore flash write endurance under abnormal use.** The bulk of existing research and publicly available information addresses flash endurance under normal, low-duty cycle use. We seek worst-case conditions which may be sought by malicious actors, and highlight this problem as an important area for exploration.
- **Introduction of GANGRENE, an attack on flash endurance.** We develop an attack demonstrating that effective wear rates can be significantly accelerated. We frame the possible methods of a GANGRENE attack and address considerations for overcoming flash controller algorithms.
- **Defending against GANGRENE.** We propose solutions and mitigations for addressing the vulnerability of flash drives to write-endurance attacks.

Outline In Section 2 we describe the architecture of USB flash devices and their interactions with the host machines with which they are used. We introduce and explain GANGRENE in greater detail in Section 3. Section 4 contains an analysis of flash endurance attacks and our empirical test results. We discuss our results and the impact to flash devices in general while proposing high-level solutions in Section 5. Section 6 discusses related work and Section 7 concludes.

2 Background

Flash memory is a non-volatile memory technology that resides in most digital systems for storage. Implementation methods vary greatly and reliance of the host system on flash storage depends on the specific instantiation.

Figure 1 depicts a breakdown of the elements that are present in a typical system (e.g., Windows 7) that interface with a USB flash drive. The flash drive itself is generally dominated by two ICs: a flash controller IC and the flash memory IC. The flash memory and the controller interact through a discrete digital electrical interface.

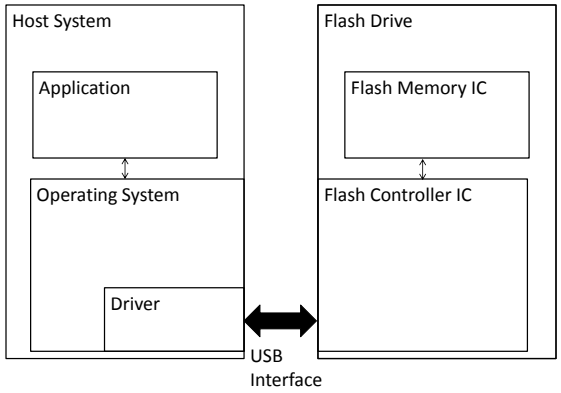

Figure 1: Illustration of a USB flash drive system depicting the major functional elements. Note that life-extending methods are implemented within the flash controller. Note also that the Host System architecture is a software stack and that blocks in the Flash Drive architecture represent hardware elements.

Flash memory. The flash IC is an array of cells where the individual charges are stored. These cells are organized into pages, which are the smallest units on which writes can be performed. These pages are organized into blocks, which are the smallest units that can be erased. Herein lies a unique property of flash devices where the architecture of a particular device imposes fixed-size units of operations. This manifests as write amplification, where the layout of a flash device and the algorithms used to perform write and erase operations can carry an overhead such that more cells can be affected than the minimum number of cells needed to store a given amount of data [10].

Flash controller. The flash controller IC generally has an integrated interface controller, which is commonly USB. The flash controller IC is a black box that is responsible for implementing algorithms and techniques to manage and extend the lifetime of flash. The controller seeks to minimize wear through wear-leveling algorithms, garbage-collection methods, use of random access memory, and other schemes. The flash controller IC maintains a Flash Translation Layer (FTL), which is a dynamic mapping between allocation units (AUs) and the physical block and page addresses of the flash system. The flash controller receives commands for AU functions from the operating system (OS) and driver. At

a minimum flash controllers level wear by distributing write activity over the entire medium, thereby preventing localized areas of high wear. Performance of the wear-leveling function can depend on the size of the data being written, but not the entropy of the data [10]. More sophisticated flash controller methods including caching, deduplication, and compression depend on both the size and entropy of the data. The performance of these methods declines as data size and content randomness increase. All of these functions are performed by the flash controller with no visibility outside of the flash device.

Host system. The OS and driver work in concert to perform much of the data transfer tasks between the application and the flash drive. These elements receive commands from the application for functions that are translated into AU operations in the file system. The OS allows basic file system functions including the creation, movement, modification, and deletion of files. Physical file and data locations are abstracted and only pointers to files in the virtually-addressed file system are accessible.

While the aforementioned system description is representative of many implementations, variants may perform the controller function in software or when integrated within the flash memory IC with additional circuitry. Some of these alternative approaches are used in mobile devices.

3 GANGRENE: Attacking Flash

Objective. The objective for an attack on flash is to cause flash degradation, which is affected by maximally flipping bits in the flash medium. Within flash cells logical 1s denote the erased state while 0s represent the written, or set state. Working against this objective is the flash controller that has a plethora of methods that act to minimize or prevent wear, which were discussed in Section 2. This objective must be realized with no a priori knowledge of the target device, neither insight into the types of algorithms used nor into the organization of the device.

GANGRENE attack. We introduce GANGRENE, an application that attacks flash endurance. The success of GANGRENE is realized by changing as many bits as possible through each write operation and inducing maximum write amplification. As established in Section 2, there is limited visibility of internal flash operations and only a small number of functions that can be used. The key variables that remain under control of GANGRENE are the size and contents of the files that are written.

To be effective in a more general case, GANGRENE must be able to overcome the efforts of a flash controller employing a plurality of methods to eliminate and/or minimize write operations. We assume the target device has a controller capable of caching, compression, and deduplication. These methods are proven effective with smaller files containing payloads such that the entropy is low. GANGRENE eschews deterministic patterns and file sizes to present more complex data to the target controller. Thus, write activity occurs though writing files with random file sizes and random contents to flash media.

File contents are randomized uniformly with 1s and 0s. While GANGRENE makes a concession to flip only one-half of the bits on average, this decision is necessary given our lack of a priori knowledge of target devices and desire for general use. While the file size has an effect on the performance of caching, compression, and deduplication algorithms, we also vary file size to affect write amplification of the target device. The physical configuration of a given flash device is fixed and write amplification is often a function of the quantity of data that is written in an operation [10]. To be effective in the general case, we vary the file size uniformly with a maximum file size of N. There is a trade-off between sufficiently large files to overcome flash controller attempts to minimize write activity and impacts to host system performance including, notably, memory.

GANGRENE must obviously maximize the duty cycle and the write/read ratio of operations commanded to the target device. Every write is followed immediately by a delete operation to induce garbage collection activity and avoid detection made possible by filling a drive to capacity. This write/delete sequence loops infinitely.

Other practical considerations. The attack must take into account overhead loading of the host system such that it remains undetected by manifestations of degraded system performance. Thus, the computational load, memory footprint, and amount of space necessary for writing on the target flash media must be considered during implementation.

4 Evaluation

Our objective is to begin a characterization of flash device performance against attacks on endurance. To that end, we must first establish evaluation criteria.

Analysis. Flash devices employ a variety of mechanisms to extend endurance and maximize performance. While analysis of these individual methods have been explored in detail, we seek a more general system-level analysis that provides utility for estimating the vulnerability of flash drives to malicious attack and metrics for describing endurance. A key notion of our analysis is the acceleration A of degradation during use. Values $A > 1$ indicate more data is written to the device than commanded and the resulting wear is accelerated. It is conceivable that $A < 1$ for controllers that effectively reduce or eliminate certain write operations that are com-

manded. In our testing we make no assumptions about A for a device, but we can empirically derive the acceleration factor if we observe *LIFE* through destruction of the device.

Equation 1 provides an estimated device lifetime *LIFE* in units of acceleration A, reported capacity C, and vendor-provided write endurance cycles E_{max}:[2]

$$LIFE = \frac{C \cdot E_{max}}{A}. \quad (1)$$

A sample application of this analysis shows that a 2 GB flash device with a nominal endurance of 10,000 cycles and a presumed A=1 can be estimated to write 20,000 GB before failure. Similarly, if a flash device writes 1,000 GB before failure, has a nominal 10,000 cycle endurance rating, and a 2 GB capacity, we can observe an acceleration A=20. The design of the flash system and flash controller seeks to maximize *LIFE*. Note that similar analyses are conducted by existing work, but they seek to determine expected life using normal data rates and neglect worst-case conditions. The naive expectation, assuming effective wear-leveling techniques and other flash controller algorithms, is that $A \approx 1$.

GANGRENE Implementation. To explore the efficacy of GANGRENE, we developed an application in C that generates attacks by varying the duty cycle, file size, and file contents. We implement GANGRENE using the considerations detailed in Section 3. While there is no basis for arriving at the exact value, we chose maximum file size N naively to be 100 MB. Additionally, GANGRENE maintains detailed logs with timestamps and write rates for each operation. Errors are reported through the host OS.

Test setup. We purchased a lot of commercially available flash drives that are representative of commonly used devices. We selected a model with no a priori knowledge of the flash IC, flash controller capabilities, or performance. The model chosen has a reported capacity of 2 GB and E_{max} of 3,000 cycles. Out of these drives a random selection of five 2 GB drives were used for endurance testing using the aforementioned approach. Additionally, another five of these 2 GB drives were chosen for a baseline test where they were written at high rates with pattern data to ensure these drives performed adequately under normal loads. Three other models from different manufacturers were evaluated during exploratory testing, but were not subjected to a formal endurance test. The collection of drives under evaluation included varied flash device integrators, raw flash vendors, and flash controllers.

[2]Our *LIFE* metric bears similarity to the JEDEC terabytes written (*TBW*) metric, `http://www.jedec.org/sites/default/files/docs/JESD218.pdf`, but we emphasize the relationship between variables and ultimately the wear acceleration, A.

Table 1: Results of GANGRENE group endurance testing. All five drives were rendered unable to write with significant rates of degradation.

Drive	*LIFE* (GB)	C (GB)	E_{max}	A
1	204.6	2	3,000	29.33
2	414.9	2	3,000	14.46
3	450.2	2	3,000	13.33
4	627.1	2	3,000	9.57
5	461.5	2	3,000	13.00

The test system is a Dell PC running the Microsoft Windows 7 OS. The same Windows Portable Devices (WPD) FileSystem Volume Driver is used by default for all tested drives. The default file system protocol and AU unit sizes are used. Performance monitoring is limited to logging successful operations and trapping errors. Errors are explicitly observed through system logs and implicitly noted through anomalous system behavior correlated with test operations.

All drives were in new condition at the start of the test to ensure that premature failure is attributed to our operations and not pedigree. All test drives underwent a baseline procedure to erase all data and check reported capacity.

Empirical results. All five drives subjected to GANGRENE accelerated wear testing experienced failures during testing such that the drives became permanently read-only. Our five 2 GB drives experienced failure after 204.6 to 627.1 GB of data were written. There was no degradation of performance and no predictor of failure before it occurred. There was no corruption of existing data files. Using vendor-provided E_{max} and C values, we can see that the acceleration A ranged from 9.57 to 29.33, which represents significant reduction in *LIFE*. The mean acceleration was 15.94 while the mean *LIFE* was 431.6 GB. Table 1 displays the results.

A total of ten drives shared the same bus on the test machine during evaluation. The average write data rate for the drives subjected to GANGRENE was 0.277 MB/s where little variation of the rate was observed during testing. The effective time for drive failure under test was 14-29 days for the five failed drives.

The set of like drives from the same lot that were subjected to normal, high-rate patterned writes experienced no anomalies, errors, or reduction in capacity after five months of constant use. Further, GANGRENE ran on a machine that attacked additional drives on an ad hoc basis. While not affecting permanent damage to the devices, these drives repeatedly endured corruption that destroyed on-board files and necessitated frequent reformatting. While not part of our formal test, we include this as anecdotal information where failure was mani-

fest, albeit in a different mode. We intend to broaden the scope of our research to include formal evaluation of more products.

5 Discussion

The results demonstrate that flash device endurance warrants additional research. While our observed *LIFE* and *A* values may not extend in the general case, employment of GANGRENE malware poses a risk to flash systems.

Flash is mortal. At the onset of the project, we hypothesized that flash devices possess a vulnerability to accelerated wear attacks. Our assumption was that damage would be affected in a localized manner resulting in graceful degradation of the device evidenced by decreasing capacity (hence our descriptive project title).

Observed failure modes range from permanent loss of write ability (with the integrity of the existing data maintained) to destruction of on-board data with the ability to maintain use of the device after reformatting. In all instances the failure arises with no warning.

In any case a degradation of utility is realized and GANGRENE accelerates wear significantly. This result contrasts with the perceived reliability of flash that is held by many. The test drive average life of 431.6 GB can be related to a duration in time given the write rate. A write rate of 250 KB/s would tax a host system minimally and would equate to failure of a drive in 21 days for our test devices. Note that this write rate is conservative and higher write rates are inevitable, being limited only by the capability of the flash device.

Defending flash in the OS. Current popular operating systems do not have a means to detect or thwart accelerated wear on flash. Flash endurance may be compromised by friendly promiscuous use in addition to malicious attacks. Familiar with the effectively infinite endurance of magnetic media, developers may misuse flash devices [13]. In both cases of malicious GANGRENE applications and promiscuously-acting friendly applications, users would benefit from controls that seek to identify high rates of wear on flash devices. Simply flagging such activity would be sufficient for even a normal user to investigate and shut suspect applications down. Such functions can be performed by the host OS by characterizing write activity to flash devices. A limitation is that the OS does not know the condition of the drive held by the controller. Thus, the OS may be unable to differentiate GANGRENE activity from intended writes of high-entropy data, e.g. encrypted or compressed files.

Flash self-defense. An intuitive approach would be to add functionality within the flash device to defend against attacks on endurance. Select solid state drives (SSDs), a subset of flash products, provide indication of wear to the user, but the accuracy of these measures and the performance of these products against GANGRENE is unknown. While the SMART protocol[3] for SAS and SATA devices offers some degree of monitoring and feedback, this does not provide the capability to thwart GANGRENE attacks. One existing secure flash scheme, Kells [3, 4], has the capability to control who can write to a flash device and could possibly be extended to defend against attacks on endurance.

Future work. Our work represents an initial exploration into the mortality of flash devices. Early failure of flash devices was observed as hypothesized, albeit in failure modes that were unexpected. Especially notable was the success of GANGRENE with no a priori knowledge of the target device and no tuning of our GANGRENE implementation. Future work would seek to study increases in acceleration and optimal attacks that can be realized by varying parameters in the implementation.

Future work would also include characterizing the behavior of more complex flash devices. Both SSDs and flash-integrated devices use flash as primary mass storage and as such are more critical to the overall system. In the unique case of flash-integrated devices, a successful attack would effect expensive and irreparable damage.

6 Related Work

There is a large body of published work that addresses flash memory endurance. The physics of flash oxide degradation are detailed by Park [12]. Analysis and simulation of wear-leveling algorithms have been researched by Ben-Aroya [1], Chang [5], Gal [7], and Toledo [1, 7] among others. Data deduplication [11] methods and caching [15] are more sophisticated flash controller techniques that have been published. Hu has addressed nicely the write amplification phenomenon and its impact on flash devices [10].

The notion of depletable storage systems squarely addresses the problem of finite write endurance [13]. Publicly available data on flash performance under worst-case conditions does not exist and only few papers even hint at the existence of a vulnerability [2, 13]. Several papers have sought to understand better flash write endurance [2, 6, 8], but these have only done so under the auspices of normal use.

Magnetic media-based hard disk drives are well-studied and benefit from a robust corpus of reliability data unlike their flash brethren. Hard drive vendors have historically touted optimistic reliability data, which research has shown to be not realistic [14]. Such research is not available for flash products and conclusions have yet to be drawn about the effective reliability of flash.

There is limited related work that focuses on the security properties of flash. Kells is a secure USB flash

[3] Technical Committee T13. `http://www.t13.org`

framework where the controller has unique control over the writes that are performed and the hosts that are validated [3, 4]. While novel, the Kells system does not address the write endurance vulnerability. The most popular thread of security-related flash research seeks to solve the problem of validation of data erasure imposed by the black box nature of flash devices [9, 16].

The existing work shows that worst-case flash endurance exploited by malicious actors is under-explored.

7 Conclusion

Our exploration of accelerated wear attacks on flash drives demonstrates that such attacks are not only possible, but that device failure can be quickly realized while using naive and unoptimized approaches. While we evaluated our technique on a sampling of test devices, we have developed a basic framework for this research and hope to further characterize the vulnerability by testing other devices and performing more complex analyses.

We hope our initial demonstration of the vulnerability of flash devices to accelerated wear will spur further research on other products such as SSDs and flash-integrated devices. Attacks on flash storage devices and non-malicious promiscuous use of flash are under-explored research topics that we hope to advance. We hope that end users will better understand flash mortality and that researchers and vendors will work to reduce or eliminate the threat of such GANGRENE attacks.

Acknowledgment

We would like to thank Shirin Nilizadeh and the anonymous reviewers for their helpful comments. We also thank John McCurley for his editorial help.

References

[1] A. Ben-Aroya and S. Toledo. Competitive analysis of flash memory algorithms. *ACM Trans. Algorithms*, 7(2):23:1–23:37, Mar. 2011.

[2] S. Boboila and P. Desnoyers. Write endurance in flash drives: measurements and analysis. In *Proceedings of the 8th USENIX Conference on File and Storage Technologies*. USENIX Association, 2010.

[3] K. R. B. Butler, S. E. McLaughlin, and P. D. McDaniel. Kells: a protection framework for portable data. In *Annual Computer Security Applications Conference*, pages 231–240, 2010.

[4] K. R. B. Butler, S. E. McLaughlin, and P. D. McDaniel. Protecting portable storage with host validation. In *ACM Conference on Computer and Communications Security*, pages 651–653, 2010.

[5] L.-P. Chang. On efficient wear leveling for large-scale flash-memory storage systems. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, pages 1126–1130. ACM, 2007.

[6] P. Desnoyers. Empirical evaluation of NAND flash memory performance. *SIGOPS Oper. Syst. Rev.*, 44:50–54, Mar. 2010.

[7] E. Gal and S. Toledo. Algorithms and data structures for flash memories. *ACM Comput. Surv.*, 37:138–163, June 2005.

[8] L. M. Grupp, A. M. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. H. Siegel, and J. K. Wolf. Characterizing flash memory: anomalies, observations, and applications. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 24–33, 2009.

[9] P. Gutmann. Data remanence in semiconductor devices. In *Proceedings of the 10th conference on USENIX Security Symposium*, Aug. 2001.

[10] X.-Y. Hu, E. Eleftheriou, R. Haas, I. Iliadis, and R. Pletka. Write amplification analysis in flash-based solid state drives. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, pages 10:1–10:9, 2009.

[11] D. Meyer and W. Bolosky. A study of practical deduplication. *Proceedings of the 9th USENIX Conference on File and Storage Technology*, 2011.

[12] Y. Park and D. Schroder. Degradation of thin tunnel gate oxide under constant Fowler-Nordheim current stress for a flash EEPROM. *Electron Devices, IEEE Transactions on*, 45(6):1361–1368, June 1998.

[13] V. Prabhakaran, M. Balakrishnan, J. D. Davis, and T. Wobber. Depletable storage systems. In *Proceedings of the 2nd USENIX conference on Hot topics in storage and file systems*, 2010.

[14] B. Schroeder and G. A. Gibson. Disk failures in the real world: what does an MTTF of 1,000,000 hours mean to you? In *Proceedings of the 5th USENIX conference on File and Storage Technologies*. USENIX Association, Feb. 2007.

[15] G. Soundararajan, V. Prabhakaran, M. Balakrishnan, and T. Wobber. Extending SSD lifetimes with disk-based write caches. In *Proceedings of the 8th USENIX Conference on File and Storage Technologies*, 2010.

[16] O. Tsur. Enabling data security with COTS solid-state flash disks. In *Non-Volatile Memory Technology Symposium, 2004*, pages 131–134, Nov. 2004.