SAFE: Fast, Verifiable Sanitization for SSDs
Or: Why encryption alone is not a solution for sanitizing SSDs

Rev v0.6.1– 13 October 2010

Steven Swanson Michael Wei

Non-Volatile Systems Laboratory
Computer Science & Engineering
University of California, San Diego
{swanson,mwei}@cs.ucsd.edu

1 Introduction

As users, corporations, and government agencies store more data in digital media, managing that data and access to it becomes increasingly important. Reliably removing data from persistent storage (i.e., *sanitizing* the storage) is an essential aspect of this management process, and several techniques that reliably delete data from hard disks are available as built-in ATA or SCSI commands, software tools, and government standards.

Recently, there have been two disruptive developments in storage sanitization. The first is the emergence of flash-based solid-state drives (SSDs) that use silicon chips rather than spinning disks to store data. The second, is the rise of encryption as a means to protect data on the drive and as a means to quickly render it irrecoverable.

Reliably erasing data from SSDs is challenging both because of the complex data management schemes they employ and because the built-in facilities for sanitization are sometimes buggy. We have evaluated built-in sanitization facilities by applying a sanitization technique, dismantling the drives, extracting the raw bits from the discrete flash devices inside, and searching for remnant data. The technique takes a few hours, is inexpensive, requires only moderate technical skill, and works independently of the controller. We have used this technique to show that some drives claim to successfully erased the drive when the data remains intact, leaving us with a strong conviction that firmware-based sanitization techniques must be verifiable to be trustworthy.

An alternative to overwriting or erasing data is to store the data in encrypted form. When the user wishes to destroy the data, the drive destroys the cryptographic key. In theory this should render the data irrecoverable. An advantage of this technique is that it is fast. It takes a fraction of a second to destroy a cryptographic key while conventional sanitization operations on an SSD may take many seconds. In emergency situations, speed is of the essence and erasure-based techniques may be too slow.

However, reliably destroying the cryptographic keys is a challenging problem. Side-channel attacks based on semiconductor memory data remanence may allow an attacker to recover the key or related data. Or, more prosaically, the implementation of key destruction may be faulty. The long history of incorrectly or insecurely implemented cryptographic systems, makes it likely that these weakness will exist in at least some SSDs.

The lack of verifiability compounds the dangers of cryptographic sanitization. Determining whether a drive correctly implements key destruction requires detailed information about drive's firmware and the silicon technology used to manufacture the controller (in order to understand memory remanence issues). Drive manufactures are hesitant to provide this kind of information, and even if they shared it freely, each SSD model would require independent verification – a time consuming and expensive process.

We propose a hybrid approach to sanitizing SSDs that combines speed of cryptographic key destruction with the verifiability of explicitly erasing the storage media. The technique, called *Scramble and Finally Erase (SAFE)*, stores encrypted data in the drive and uses a two step process for sanitization. First, it destroys the key. Then, SAFE erases every physical page in the SSD. After this step, verification is a simple matter of dismantling the drive and verifying that the flash chips are actually erased.

This report makes the detailed case for SAFE as an alternative to purely cryptographic erasure techniques. We show that SAFE can provide fast sanitization without sacrificing reliability. Given the demonstrated history of buggy sanitize operations and the long history of vulnerabilities in the application of cryptography, it would be unwise to rely solely on those techniques, especially when a viable alternative, like SAFE, is available.

The rest of the report is organized as follows. Section 2 summarizes our work in verifying sanitization commands. Section 3 describes the dangers of relying on cryptographic techniques and difficulties in verifying them. Section 4 describes our proposal for the SAFE protocol, and Section 5 concludes.

2 Verifying erasure-based methods

The most reliable way to sanitize a device is to erase and/or overwrite the data it contains. Modern drives have built-in sanitize commands that should do just that. We have found that these commands are effective, if they are implemented correctly. Since this is not always the case, it is essential to have the means to verify their efficacy.

Below we describe the method we used to validate the built-in sanitization commands and results for several SSDs.

2.1 Test methods

We tested the built-in sanitization operations of 12 different SSDs. For each drive, we verified effectiveness by writing a series of recognizable patterns to the entire drive and then applying the sanitize command. Then we dismantled the drive and searched the individual chips for any unerased data.

We tested commands from the ATA security command set. That command set specifies an “ERASE UNIT” command that erases all user-accessible areas on the drive [2]. The new ACS-2 specification [3], which is still in draft at the time of this writing, specifies a “BLOCK ERASE” command that is part of its SANITIZE feature set, which instructs a drive to perform a block erase on all memory blocks containing user data.

Our method for verifying the success of a sanitization operation uses the lowest-level digital interface to the data in an SSD: the pins of the individual flash chips. We access the pins using the flash testing system in Figure 1. The testing system uses an FPGA running a Linux software stack to provide direct access to the flash chips.

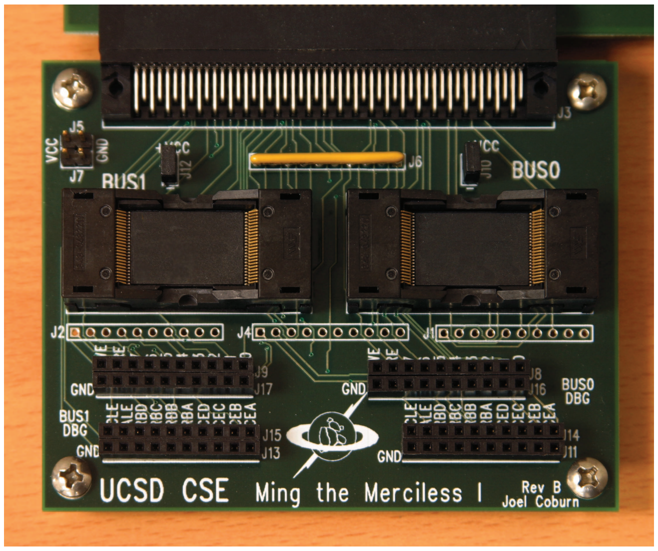

Figure 1: **Ming the Merciless** Our custom FPGA-based flash characterization hardware showing flash chips inside prototyping sockets.

2.2 Findings

We found that support and implementation of the built in commands varied across vendors and firmware revisions (Table 1). Of the 12 drives we tested, none supported the ACS-2 “SANITIZE BLOCK ERASE” command. This is not surprising, since the standard is not yet final. Eight of the drives reported that they supported the ATA SECURITY feature set. One of these encrypts data, so we could not verify if the sanitization was successful. Of the remaining seven, only four executed the “ERASE UNIT” command reliably.

Drive B’s behavior is the most disturbing: it reported that sanitization was successful, but *all* the data remained intact. In fact, the filesystem was still mountable. Two more drives suffered a bug that prevented the ERASE UNIT command from working unless the drive firmware had recently been reset. Otherwise, only the command would only erase the first LBA. In these cases, though, the drives accurately reported that the command failed.

2.3 Discussion

Our results demonstrate three important facts about sanitizing SSDs.

1. Built in erasure-based sanitization techniques can effectively sanitize flash-based SSDs.

SSD #	Ctlr # & Type	Capacity (GB)	SECURITY ERASE UNIT	time (s)	SECURITY ERASE UNIT ENH	time (s)
A	1-MLC	120GB	Not Supported	N/A	Not Supported	N/A
B	2-SLC	32GB	Failed∗	4.5s	Not Supported	N/A
C	1-MLC	30GB	Failed†	7.6s	Not Supported	N/A
D	3-MLC	120GB	Failed†	7.4s	Not Supported	N/A
E	4-MLC	60GB	Encrypted‡	0.8s	Encrypted‡	0.7s
F	5-MLC	40GB	Success	39s	Success	42s
G	6-MLC	50GB	Success	7.6s	Success	7.5s
H	7-MLC	64GB	Success	56s	Success	59s
I	8-MLC	120GB	Success	39s	Success	42s

∗Drive reported success but all data remained on drive
†Sanitization only successful under certain conditions
‡Drive encrypted, unable to verify if keys were deleted

Table 1: **Built-in ATA sanitize commands** Support for built-in ATA security commands varied among drives, and three of the drives tested did not properly execute a sanitize command it reported to support.

2. In some cases, the implementations of these techniques have serious errors that can prevent successful sanitization, even when the drive reports that it was successful.

3. Encrypting data on the drive makes it impossible to verify whether sanitization was successful.

The first two points argue strongly for the use of erasure-based techniques, but also show that verifiability is absolutely essential. Otherwise, users may be lulled into a false sense of security – although the drive says sanitization was a success, their data may still remain on the drive. The final point highlight the fact that it is very difficult to tell whether the drive is actually sanitized. The next section addresses this point in more detail.

3 The shortcomings of cryptography

Encrypting data on a drive and then destroying the keys to render the data irrecoverable is an alluring prospect. Destroying a key takes a fraction of a second, making it ideal for emergency situations where fast destruction is essential.

Crytographic erasure depends on the manufacturer implementing key destruction correctly. This means they must take into account detailed information about how the key is used, stored, and managed in order prevent side-channel attacks that would allow an attacker to recover the key. The history of secure computing is littered with buggy implementation of cryptographic systems that were secure in theory but not in practice. Furthermore, the results in Section 2 demonstrate that bugs exist even in the comparatively simple erasure-based techniques.

Since it is unreasonable to expect that SSD manufacturers will always implement key destruction correctly, some form of verification is required. That verification will be expensive.

Determining whether a cryptographic key has been rendered irrecoverable is very difficult, since it requires three difficult, time-consuming, and expensive steps. First, we must determine which on-chip memories the controller needs to sanitize in order to ensure that keys are irrecoverable. These include memories that held cryptographically important intermediate values as well as the key itself. This information will include detailed plans for the controller chip and access to the controller firmware.

Second, we would need to know the details of how the controller chip was manufactured. The data remanence [6, 5] characteristics of the registers, SRAMs, and embedded DRAMS that a controller might use will vary with manufacturing technology. Data remanence is a serious concern since practical attacks on computer systems have exploited this type of remanence [7].

Assembling this information is just the first step of verification. The next step is a complete and thorough design review of each of the components and, potentially, experimental validation of the technique. This

	Verification	Speed
Erasure-based techniques	Simple Inexpensive General	10s of seconds
Cryptography-based techniques	Complex Expensive Per-drive	< 1 second

Figure 2: **Speed and verification cost trade-offs** Erasure-based schemes are slower, but easy to verify. Cryptographic schemes are faster, but verification is impractical in general. SAFE provides the best of both worlds.

process would need to be repeated for each drive model from each manufacturer. With well over 100 SSD makers competing the market today, verifying even a significant fraction of them not feasible.

4 The SAFE protocol

Erasure- and cryptography-based sanitization techniques represent different trade-offs (Figure 2), but neither of them provides an ideal solution: A fast sanitization scheme that is reliable because it is easy to verify.

This section describes our proposed sanitization procedure: Scramble and Finally Erase (SAFE). SAFE combines encryption-based and erasure-based techniques to provide almost instant erasure along with verifiability. We describe the SAFE algorithm and discuss its performance and security properties. Then we compare it to degaussing, a state-of-the-art mechanism for hard disk sanitization.

4.1 SAFE-ly sanitizing an SSD

SAFE assumes that data in the SSD is stored in encrypted form and that the SSDs implements best practices with respect to key management (e.g., that the key should never leave the controller).

The SAFE algorithm is as follows:

1. Upon receiving a sanitize command, sanitize the memory containing the cryptographic keys.
2. Mark the drive as being KEYLESS.
3. Erase every block in the device, overwrite all pages with a known pattern, and erase them again.
4. Mark the drive as being VERIFIABLE.
5. Upon receiving a re-initialization command, perform a low-level format the drive.

The algorithm introduces two new states that a drive can be in: KEYLESS and VERIFIABLE. A KEYLESS drive is incapable of encrypting or decrypting data, and as long as the key destruction process work properly, no data can be recovered from the drive. A VERIFIABLE drive provides the guarantee that all data in all blocks were erased. In this state, it is easy to verify erasure was complete, by applying the technique described in Section 2. A VERIFIABLE drive would probably not be usable, since all the metadata the controller stores in the flash is missing. The final step restores the drive to a usable state.

Step 3 in the process includes an alternating erase/program/erase sequence to remove analog remanence from the flash storage. Our conversations with industry [1] suggest that a single erasure maybe be sufficient for MLC devices and that erasing, programming, and erasing is sufficient for SLC devices.

4.2 Performance

The main disadvantage of erasure-based techniques relative to key destruction is that erasure takes longer. For hard drives, the time to erase a large disk is measured in hours, but for SSDs it is much shorter. It takes about 13 seconds to completely erase a 4Gbit flash chip and about 2.1 minutes to program an entire 4Gbit SLC chip. An SSD can perform operations in parallel across many chips simultaneously, so the total time to erase a drive need not be much longer. Indeed, as Table 1 demonstrates, drives can erase their contents in between 7 and 59 seconds. Since an SSD can erase all the chips in parallel, latencies for larger drives need not be much longer. The data in Table 1 bears this out: Erasing Drive I takes less time than erasing Drive H even though it is nearly twice the size.

Using the numbers calculated above, SAFE would take between 10s of seconds to complete on an MLC drive, and a few minutes on an SLC drive. In both cases, the drive enters the KEYLESS state almost instantaneously.

We have also heard that the concern that simultaneous erases on all the flash chips in an SSDs in parallel would take a large amount of power. Our experiments [4] show that erase operations take at most 100 mW. Even on a drive with 16 flash chips, this amounts to less than 2 W.

4.3 Security

Since SAFE includes complete erasure of the storage media, it eliminates risks due to incorrect implementation of the key destruction operation, side-channel attacks on controller to recover information about the destroyed key, and future attacks on the cryptosystem (e.g., AES) that protects the data.

Since we can check that drives have implemented the erasure portion of SAFE correctly, the protection SAFE provides is verifiable. Furthermore, verification is highly reliable, fast (less than a day), and does not require access to any proprietary information about the drive. All that is required is the ability to read bits off flash chips and information from the manufacture about data remanence so that the drive can provide adequate erase/overwrite in Step 3 of the process. Manufactures have already demonstrated willingness to provide this information [1].

SAFE provides the best known assurances of sanitization at multiple time scales. If instant sanitization is necessary, the key destruction portion of SAFE can provide it with a moderate level of confidence. If more time is available, the erasure stage provides highly reliable erasure-based protection.

4.4 Comparison to hard drive degaussing

SAFE provides a level of protection and verifiability similar to those that degaussing provides for magnetic media. In particular, they both provide a means for a third party verify sanitization without relying on the drive manufactures hardware or software.

In the case of degaussing verification takes the form of examining degaussed disk platters with a magnetic force microscope and verifying that only noise remains. Our colleague, Fred Spada, at UCSD has used just this approach to demonstrate that modern degaussers are reliable enough to handle the most demanding sanitization scenarios.

SAFE sets the same standard for SSDs, by providing a well-defined means to examine the contents of the SSD without interference from the controller.

5 Conclusion

SAFE sanitization combines two best-in-class techniques for destroying data in encrypted SSDs – physical erasure and key destruction. The combination provide a verifiable sanitization solution that will allow governments, businesses, and individuals to be confident that their data is irretrievable once they choose to destroy it. While the performance advantages of key destruction alone are attractive, it would be unwise to ignore its inherent vulnerabilities. By combining both techniques, SAFE provides an equivalent level of assurance when time is of the essence, and much stronger guarantees when more time is available.

References

[1] M. Abraham. Nand flash security. In *Special Pre-Conference Workshop on Flash Security*, August 2010.

[2] American National Standard of Accredited Standards Committee X3T13. *Information Technology - AT Attachment-3 Interface*, January 1997.

[3] American National Standard of Accredited Standards INCITS T13. *Information Technology - ATA/ATAPI Command Set - 2*, June 2010.

[4] 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 *MICRO'09: Proceedings of ...*, New York, NY, USA, 2009. ACM, IEEE.

[5] P. Gutmann. Secure deletion of data from magnetic and solid-state memory. In *SSYM'96: Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography*, pages 8–8, Berkeley, CA, USA, 1996. USENIX Association.

[6] P. Gutmann. Data remanence in semiconductor devices. In *SSYM'01: Proceedings of the 10th conference on USENIX Security Symposium*, pages 4–4, Berkeley, CA, USA, 2001. USENIX Association.

[7] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: cold-boot attacks on encryption keys. *Commun. ACM*, 52(5):91–98, 2009.