Reliably Erasing Data from Flash-Based Solid State Drives

Michael Wei[*]

Laura Grupp[*], Fredrick E. Spada[†], Steven Swanson[*]

[*] **Non-Volatile Systems Laboratory**
Department of Computer Science and Engineering
University of California, San Diego

[†] **Center for Magnetic Recording Research**
University of California, San Diego

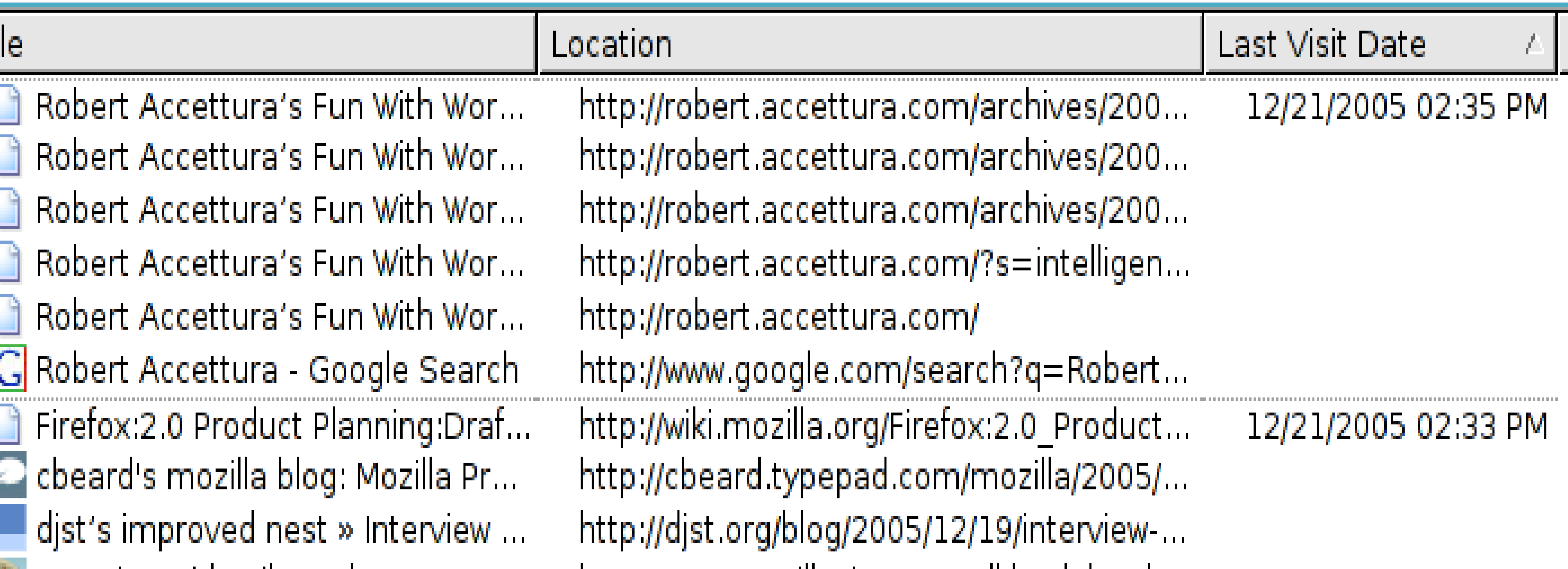
Confidential Data

sensitive information which…

- Limited to people with need
- Destroyed at end of life

YOU…

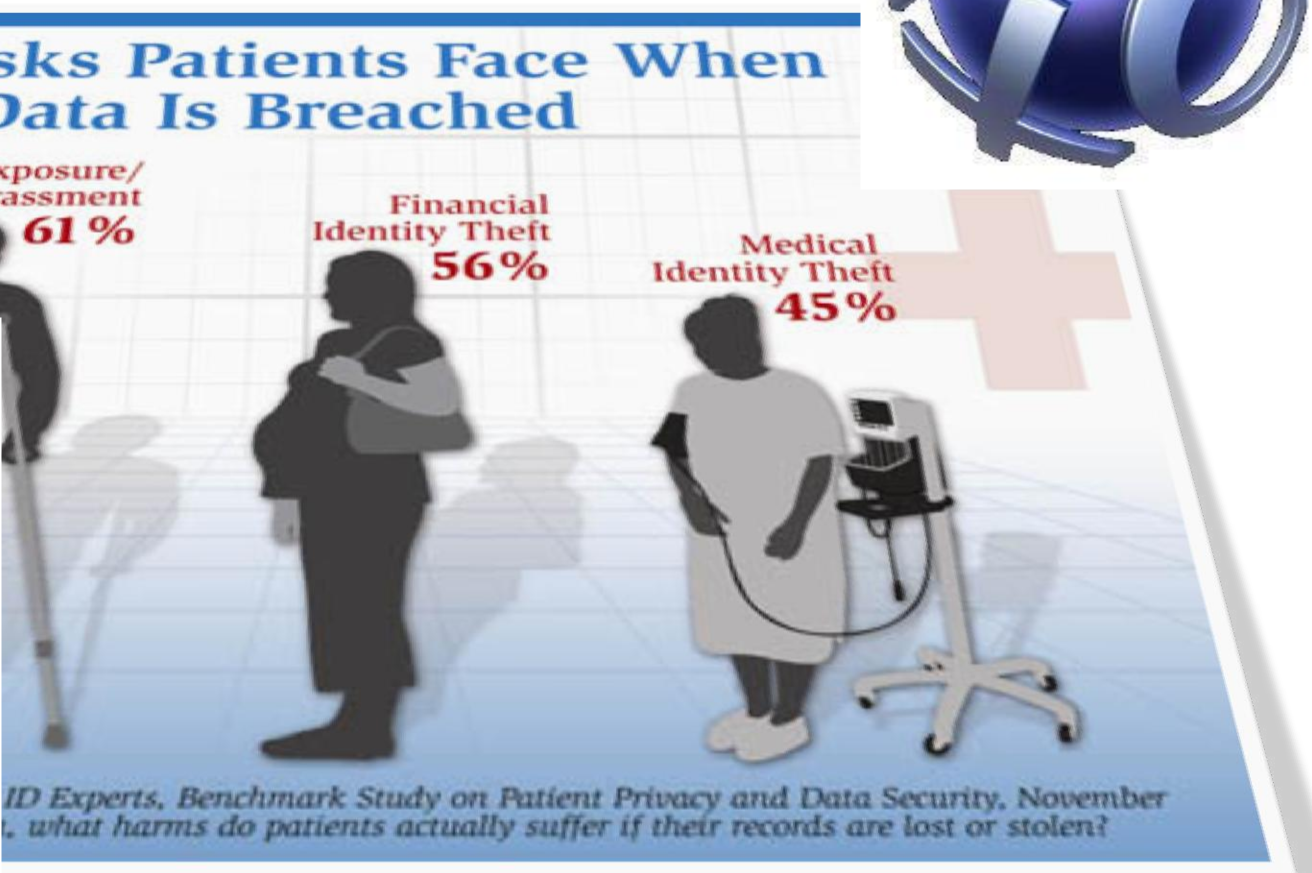
have confidential data on your computer right now!



le	Location	Last Visit Date
Robert Accettura's Fun With Wor…	http://robert.accettura.com/archives/200…	12/21/2005 02:35 PM
Robert Accettura's Fun With Wor…	http://robert.accettura.com/archives/200…	
Robert Accettura's Fun With Wor…	http://robert.accettura.com/archives/200…	
Robert Accettura's Fun With Wor…	http://robert.accettura.com/?s=intelligen…	
Robert Accettura's Fun With Wor…	http://robert.accettura.com/	
Robert Accettura - Google Search	http://www.google.com/search?q=Robert…	
Firefox:2.0 Product Planning:Draf…	http://wiki.mozilla.org/Firefox:2.0_Product…	12/21/2005 02:33 PM
cbeard's mozilla blog: Mozilla Pr…	http://cbeard.typepad.com/mozilla/2005/…	
djst's improved nest » Interview …	http://djst.org/blog/2005/12/19/interview-…	

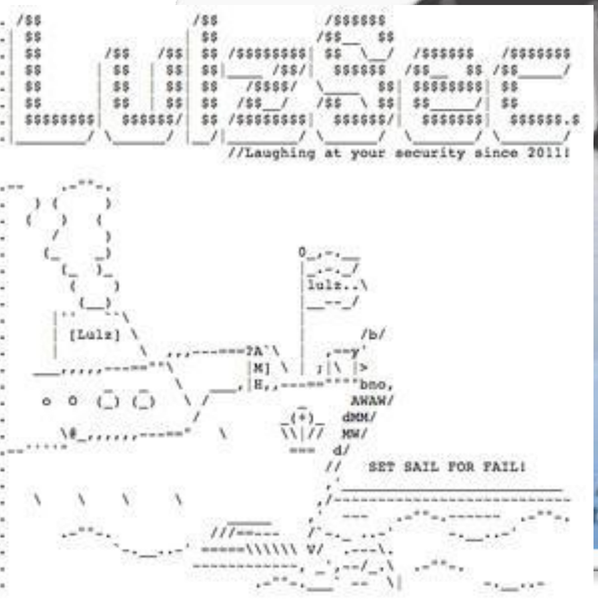
CORPORATIONS…

must protect their own data as well as client's data.



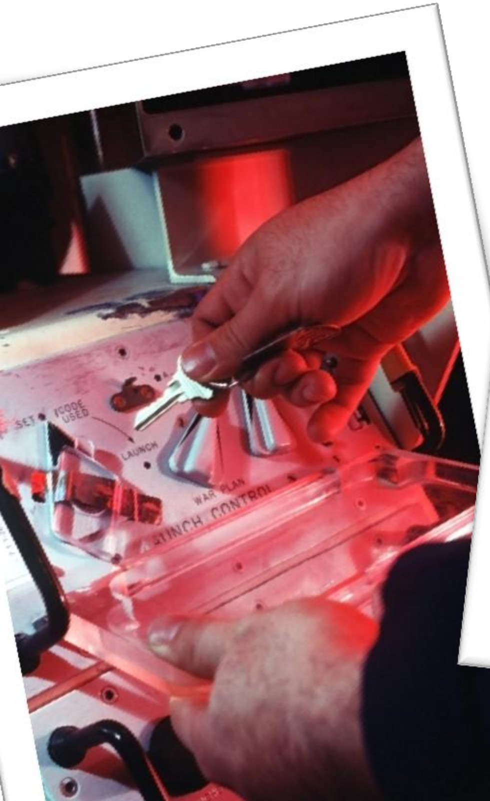
Top Risks Patients Face When Their Data Is Breached

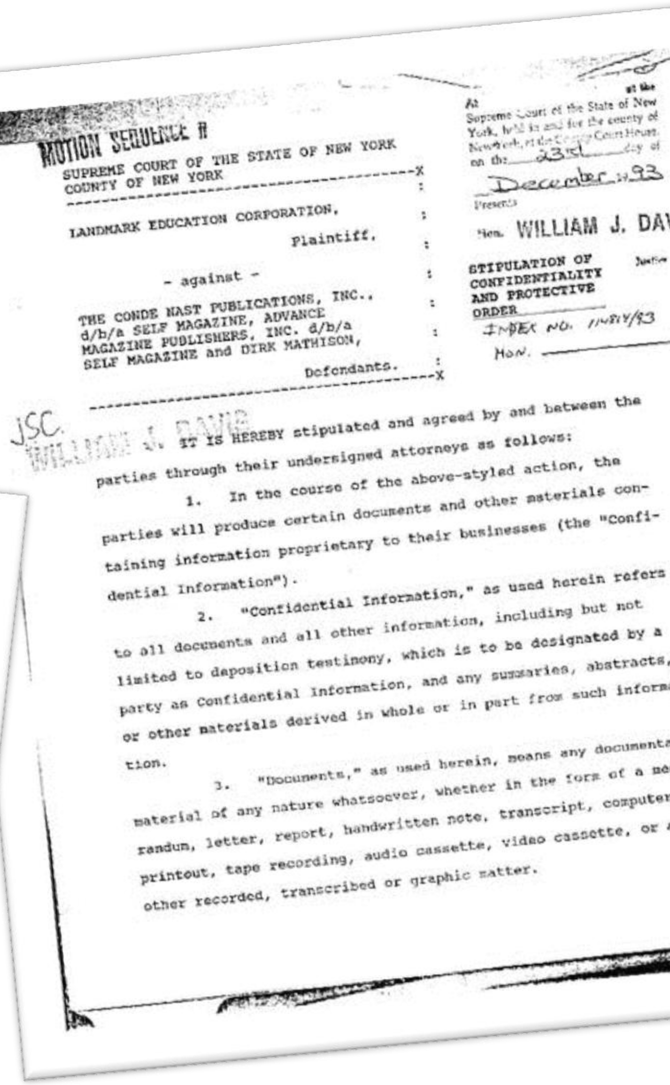
ID Experts, Benchmark Study on Patient Privacy and Data Security, November
, what harms do patients actually suffer if their records are lost or stolen?



GOVERNMENTS...

must protect information to protect the state and lives of its citizens





MOTION SEQUENCE #

SUPREME COURT OF THE STATE OF NEW YORK
COUNTY OF NEW YORK

LANDMARK EDUCATION CORPORATION,

Plaintiff,

- against -

THE CONDE NAST PUBLICATIONS, INC., d/b/a SELF MAGAZINE, ADVANCE MAGAZINE PUBLISHERS, INC. d/b/a SELF MAGAZINE and DIRK MATHISON,

Defendants.

At at the Supreme Court of the State of New York, held in and for the county of New York, at the County Court House, on the 23rd day of December 1993

Present:

Hon. WILLIAM J. DA

Justice

STIPULATION OF CONFIDENTIALITY AND PROTECTIVE ORDER

INDEX NO. 114814/93

HON.

JSC.

WILLIAM J. DAVIS

IT IS HEREBY stipulated and agreed by and between the parties through their undersigned attorneys as follows:

1. In the course of the above-styled action, the parties will produce certain documents and other materials con-taining information proprietary to their businesses (the "Confi-dential Information").

2. "Confidential Information," as used herein refers to all documents and all other information, including but not limited to deposition testimony, which is to be designated by a party as Confidential Information, and any summaries, abstracts, or other materials derived in whole or in part from such inform tion.

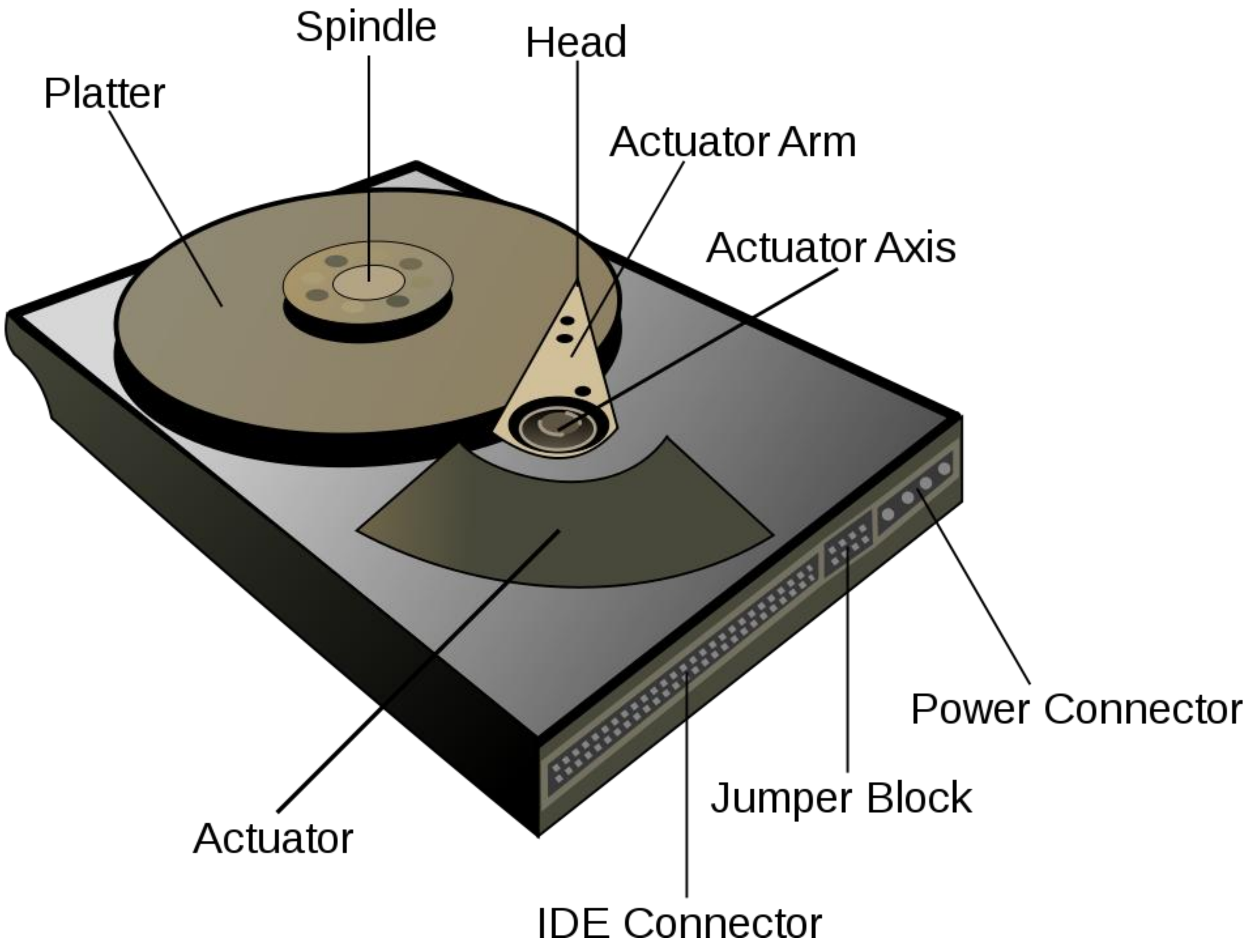
3. "Documents," as used herein, means any documenta material of any nature whatsoever, whether in the form of a me randum, letter, report, handwritten note, transcript, computer printout, tape recording, audio cassette, video cassette, or other recorded, transcribed or graphic matter.

Confidential Data

sensitive information which…

- Limited to people with need
- Destroyed at end of life

How?

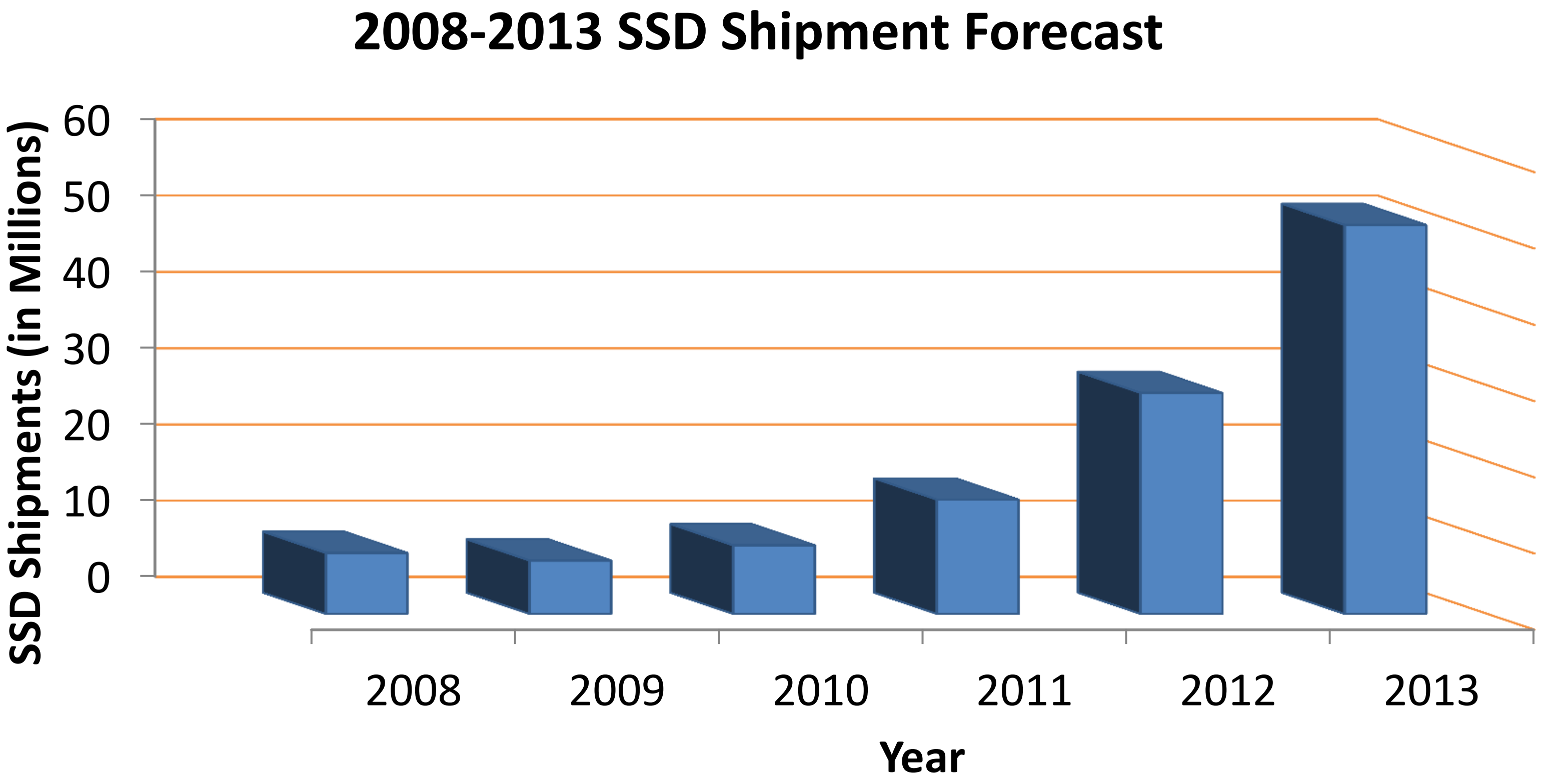


What we know comes from years of research on hard drives.

Solid State Disks (SSDs)

next generation storage…

- Flash-based
- No moving parts
- Uses a complex controller (Flash Translation Layer)

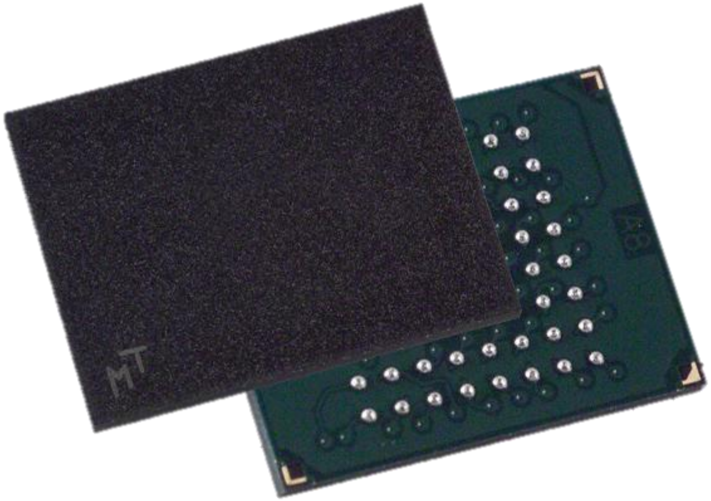


Source: DRAMeXchange

SSDs are becoming quite popular…

You might have left confidential data
and not even realized it.

Why is it hard to erase SSDs?



Current sanitization tools are designed for hard drives.

But SSDs are very different!

SSD Differences

- Recovery process is cheap
- Wide space of manufacturers for poor implementation
- Easy Disassembly / Reassembly

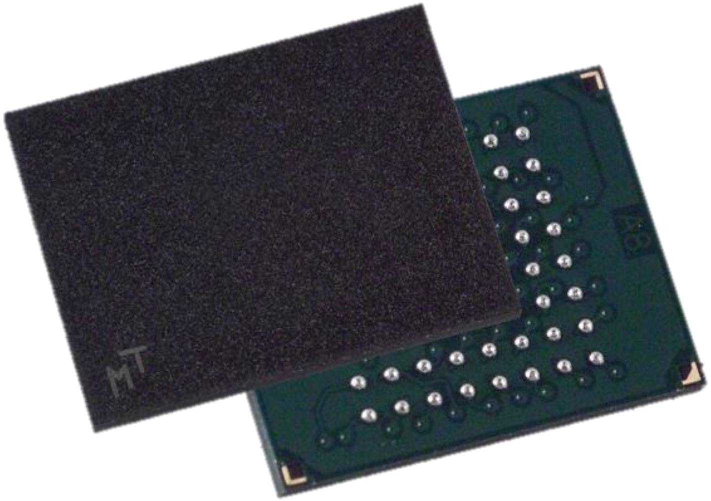
- Low cost compared to hard drives
- Someone could steal your data overnight!

Overview

- Motivation
- **Sanitization Background**
- Validating Sanitization and Results
- Single-File Sanitization Enhancement

Sanitization

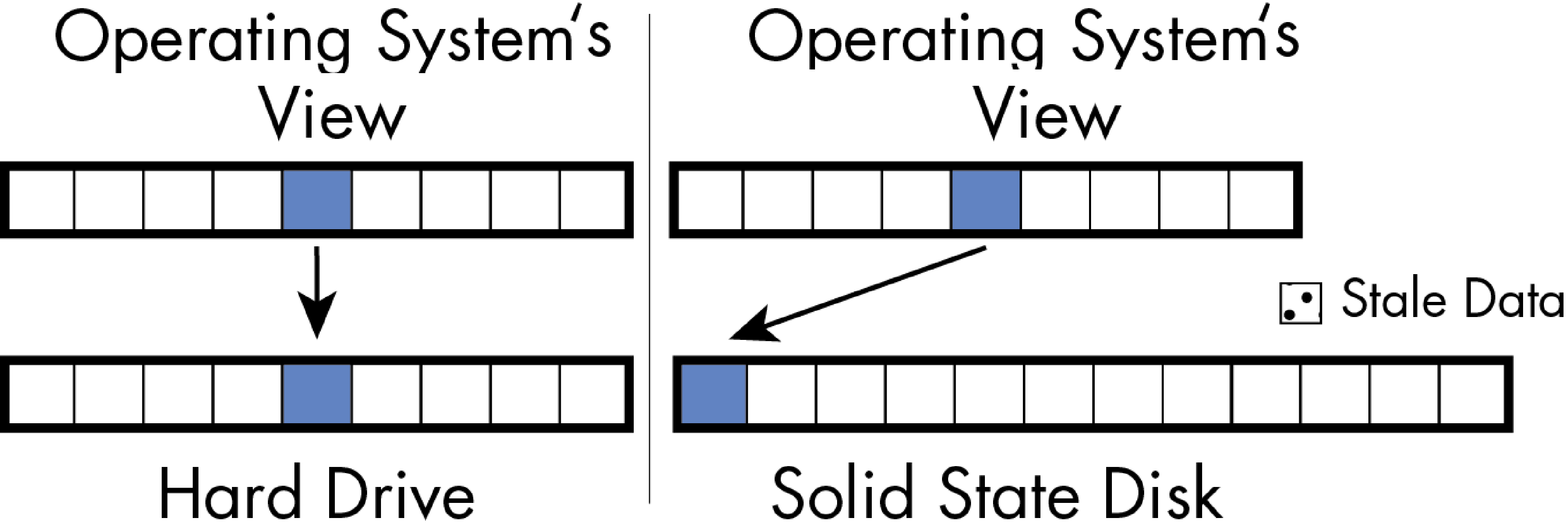
Erasing data so that it is difficult or impossible to recover



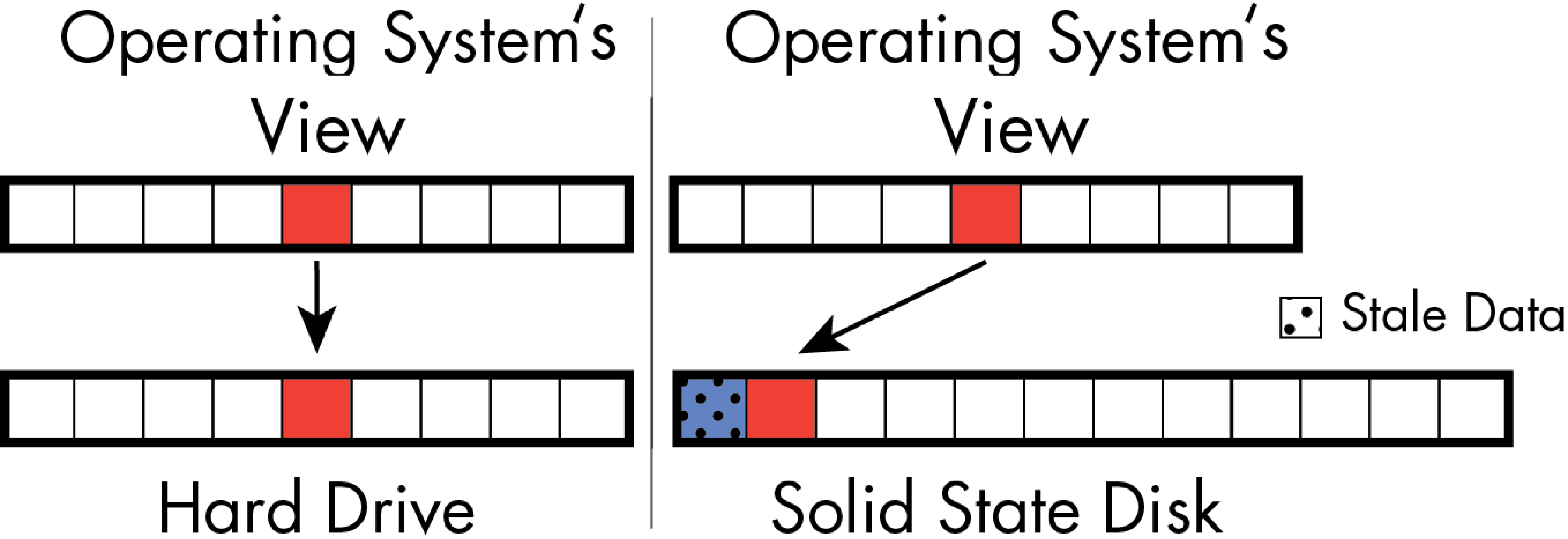
For this talk, we'll talk about the chip level.

- There's leftover data
- It's cheap
- The next level is much more complex

Writing Data



Writing more data…



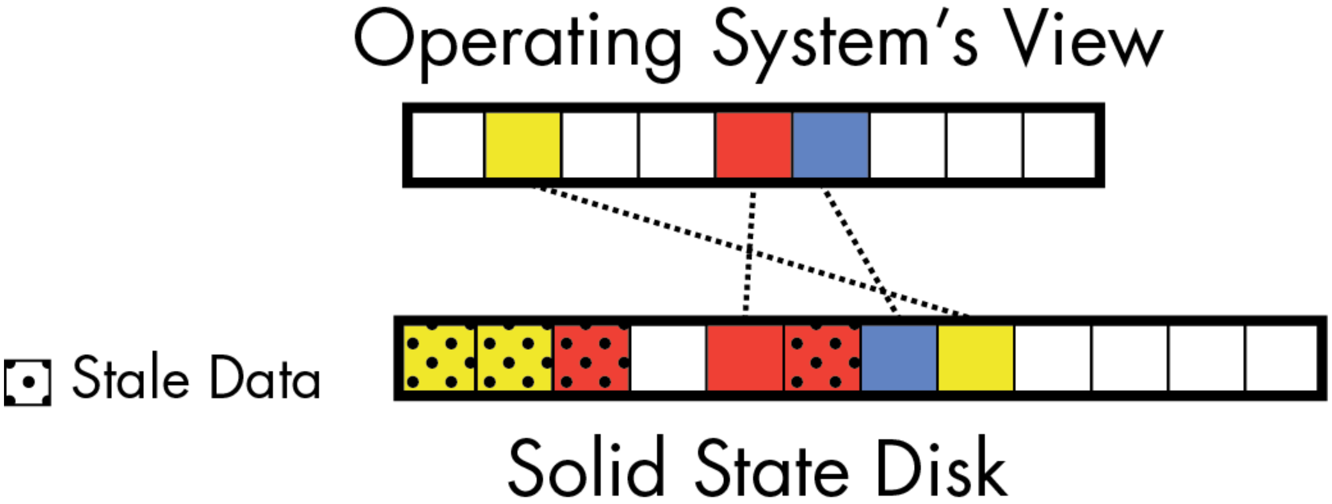
Operating System's View

Hard Drive

Operating System's View

Stale Data

Solid State Disk

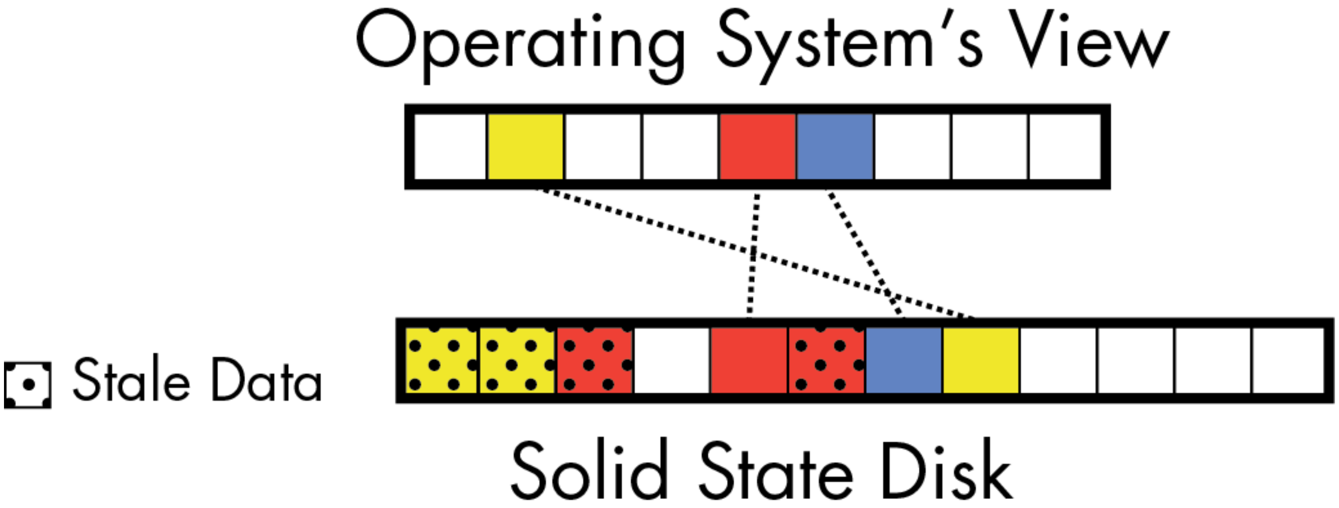


Lots of stale data can be left over on the drive…

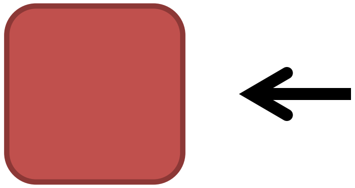
Overview

- Motivation
- Sanitization Background
- **Validating Sanitization and Results**
- Single-File Sanitization Enhancement

We now want to measure the stale data left over.

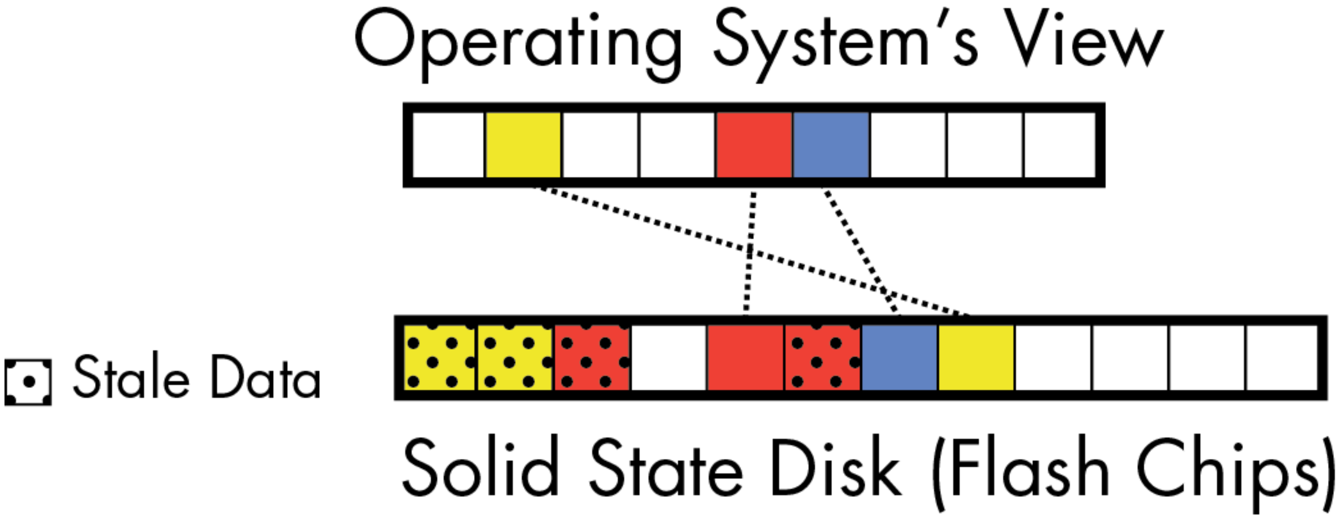


First, we constructed a “fingerprint” that was easily identifiable.

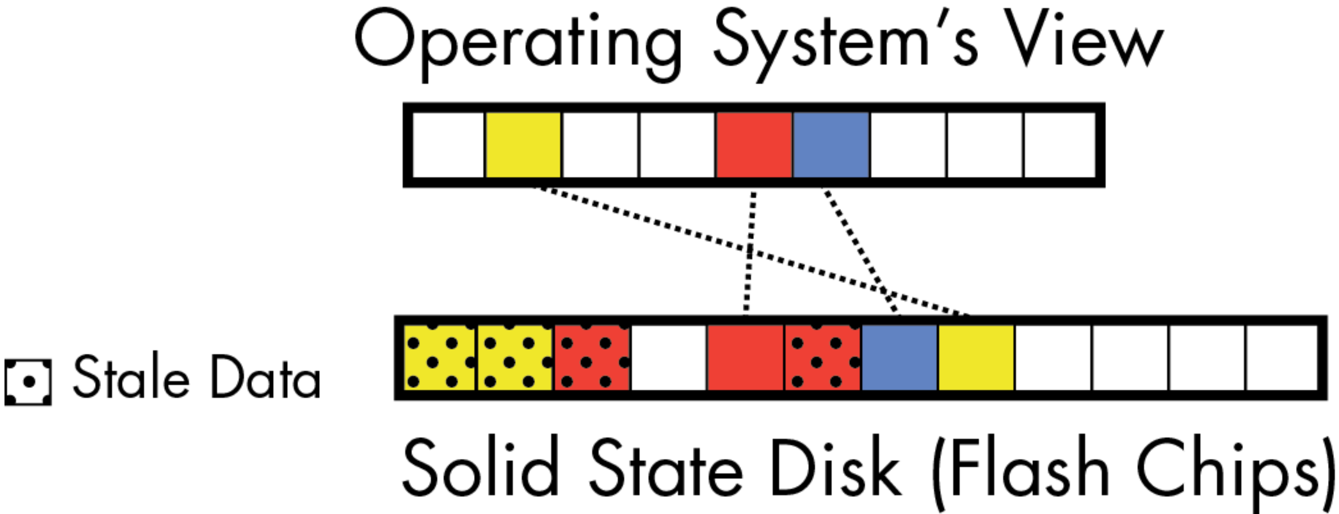


Special Identifiers
Unique Patterns
Checksum

Second, We needed a way to see more than what the operating system sees.



Second, We needed a way to see more than what the operating system sees.

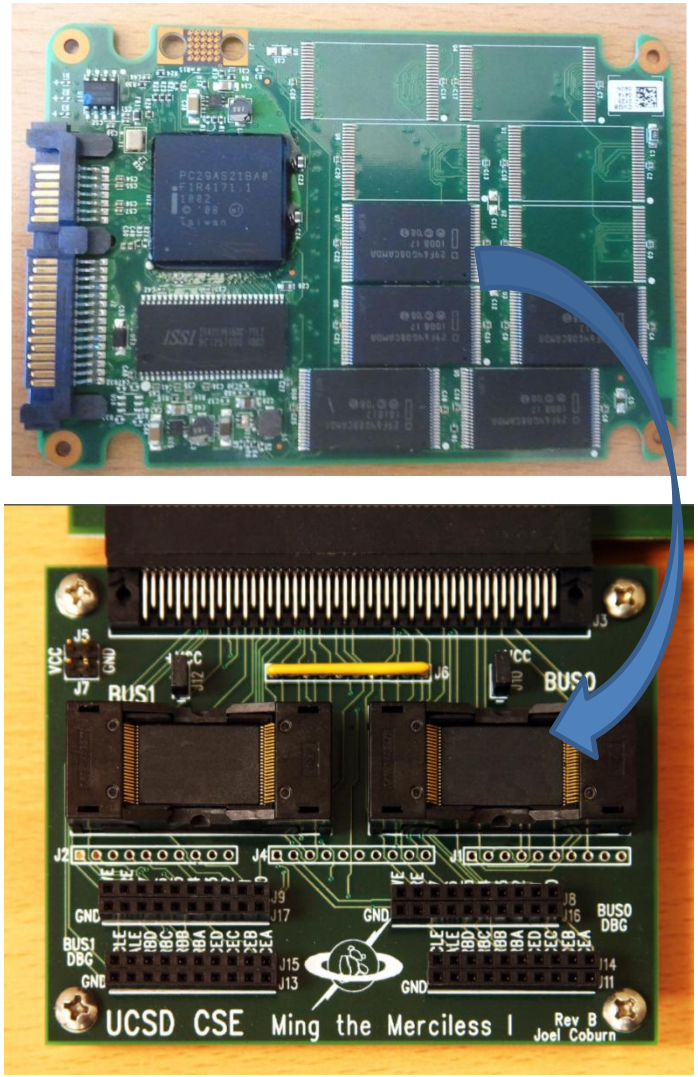


Operating System's View

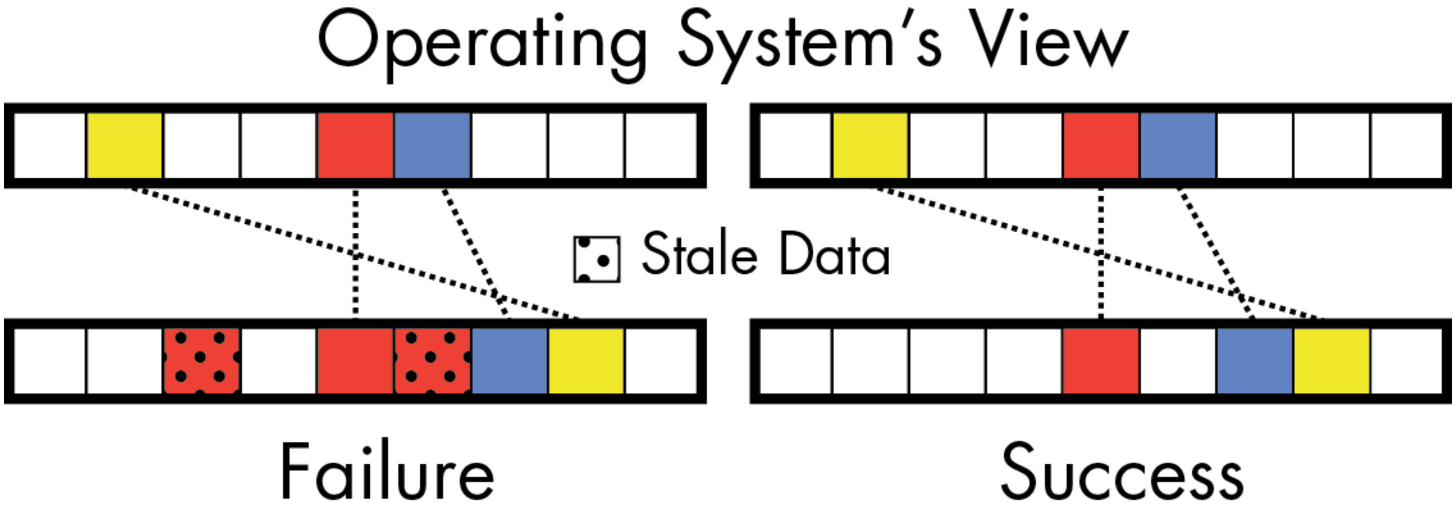
▨ Stale Data

Solid State Disk (Flash Chips)

We built a custom hardware platform to extract data off the chips.



The drive is successfully sanitized if* no stale data is left over.



Whole-disk sanitization

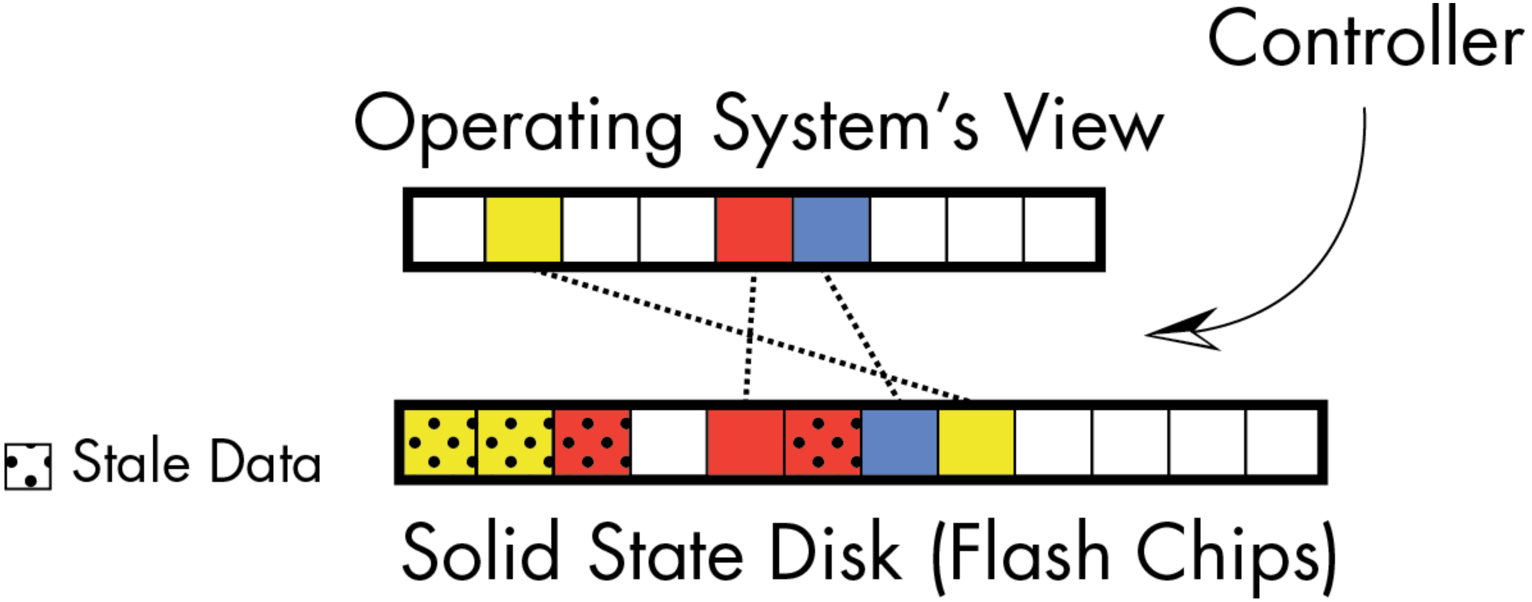
Erase the whole disk so that no old data remains.

- **Built-in Commands**
  - ATA Security “Erase Unit” (ATA-3), 1995
  - Cryptographic techniques

- **Software Overwrite**
  - Various Standards

Built-in commands

- ATA Security "Erase Unit"



ATA Security Erase Unit (1995)

- Normal: Replace the contents of LBA 0 to MAX LBA with binary zeroes or ones.

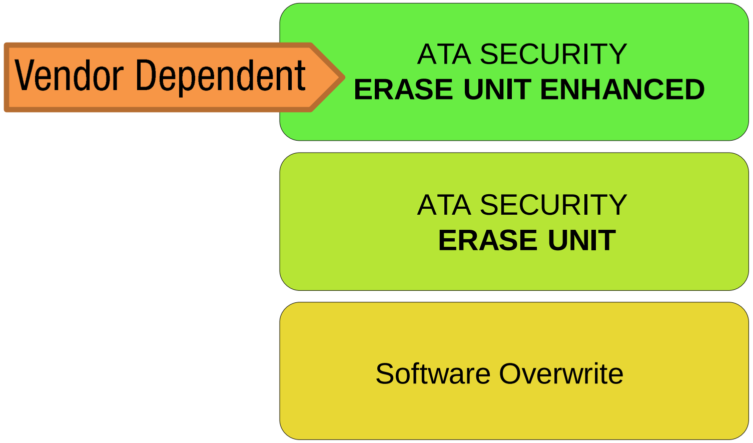
- Enhanced: All previously written user data shall be overwritten.

Predates SSDs: doesn't distinguish overwritten from erase.

ATA Security Erase Enhanced

Some drives tested supported and passed

SSD Name	Controller	SECURITY ERASE UNIT (ATA-3)	SECURITY ERASE UNIT ENHANCED (ATA-3)
A	1	No	No
B	2	No (Reports yes)	No
C	1	Partial (Bugged)	No
D	3	Partial (Bugged)	No
E	4	Crypto Scrambles	Crypto Scrambles
F	5	Yes	Yes
G	6	Yes	No
H	7	Yes	Yes
I	8	Yes	Yes



Vendor Dependent

ATA SECURITY **ERASE UNIT ENHANCED**

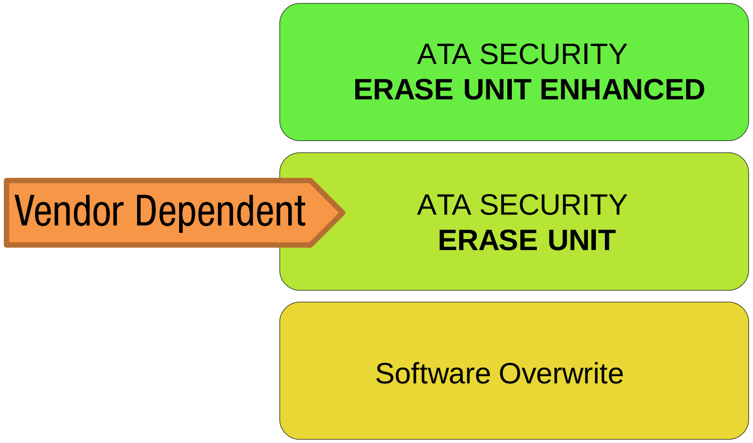
ATA SECURITY **ERASE UNIT**

Software Overwrite

ATA Security Erase Unit

One drive reported success, even though all data remained.

SSD Name	Controller	SECURITY ERASE UNIT (ATA-3)	SECURITY ERASE UNIT ENHANCED (ATA-3)
A	1	No	No
B	2	No (Reports yes)	No
C	1	Partial (Bugged)	No
D	3	Partial (Bugged)	No
E	4	Crypto Scrambles	Crypto Scrambles
F	5	Yes	Yes
G	6	Yes	No
H	7	Yes	Yes
I	8	Yes	Yes



Vendor Dependent

ATA SECURITY **ERASE UNIT ENHANCED**

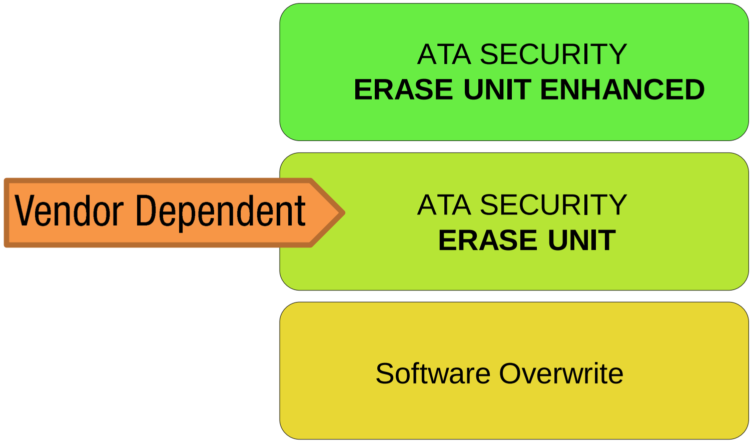
ATA SECURITY **ERASE UNIT**

Software Overwrite

ATA Security Erase Unit

- Others only worked after the drive was reset

SSD Name	Controller	SECURITY ERASE UNIT (ATA-3)	SECURITY ERASE UNIT ENHANCED (ATA-3)
A	1	No	No
B	2	No (Reports yes)	No
C	1	Partial (Bugged)	No
D	3	Partial (Bugged)	No
E	4	Crypto Scrambles	Crypto Scrambles
F	5	Yes	Yes
G	6	Yes	No
H	7	Yes	Yes
I	8	Yes	Yes


ATA SECURITY ERASE UNIT ENHANCED

Vendor Dependent

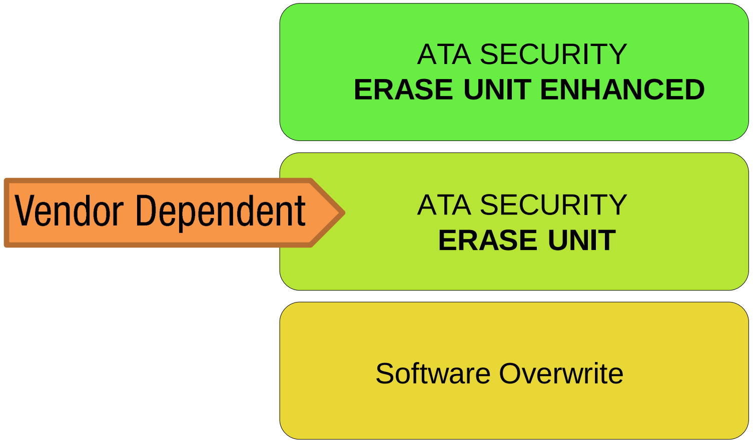
ATA SECURITY ERASE UNIT

Software Overwrite

ATA Security Erase Unit

- Some drives crypto-scrambled, so we could not verify them

SSD Name	Controller	SECURITY ERASE UNIT (ATA-3)	SECURITY ERASE UNIT ENHANCED (ATA-3)
A	1	No	No
B	2	No (Reports yes)	No
C	1	Partial (Bugged)	No
D	3	Partial (Bugged)	No
E	4	Crypto Scrambles	Crypto Scrambles
F	5	Yes	Yes
G	6	Yes	No
H	7	Yes	Yes
I	8	Yes	Yes



ATA SECURITY **ERASE UNIT ENHANCED**

Vendor Dependent

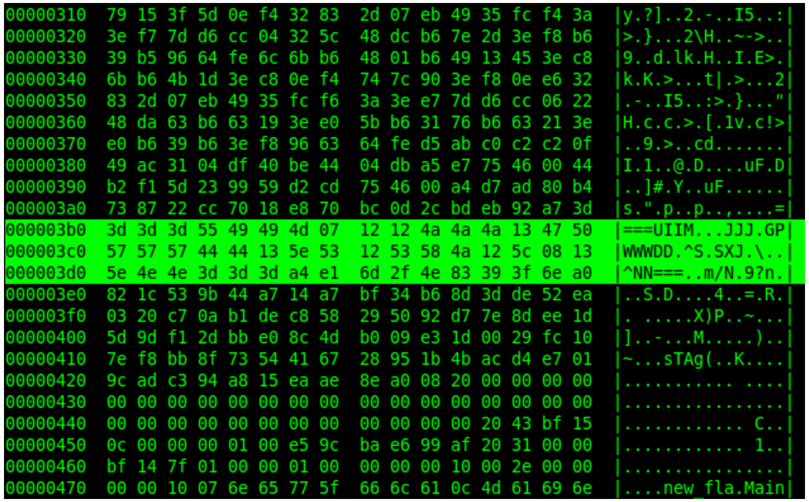
ATA SECURITY **ERASE UNIT**

Software Overwrite

Crypto-Scramble

Works by deleting key

- Fast, but…
  - Encrypted data remains
  - Implementation weakness
  - Not really sanitization
- Data isn’t erased
- Crypto scramble makes drives unverifiable



```
00000310  79 15 3f 5d 0e f4 32 83  2d 07 eb 49 35 fc f4 3a  |y.?]..2.-..I5..:|
00000320  3e f7 7d d6 cc 04 32 5c  48 dc b6 7e 2d 3e f8 b6  |>.}...2\H..~->..|
00000330  39 b5 96 64 fe 6c 6b b6  48 01 b6 49 13 45 3e c8  |9..d.lk.H..I.E>.|
00000340  6b b6 4b 1d 3e c8 0e f4  74 7c 90 3e f8 0e e6 32  |k.K.>...t|.>...2|
00000350  83 2d 07 eb 49 35 fc f6  3a 3e e7 7d d6 cc 06 22  |.-..I5..:>.}..."|
00000360  48 da 63 b6 63 19 3e e0  5b b6 31 76 b6 63 21 3e  |H.c.c.>.[.1v.c!>|
00000370  e0 b6 39 b6 3e f8 96 63  64 fe d5 ab c0 c2 c2 0f  |..9.>..cd.......|
00000380  49 ac 31 04 df 40 be 44  04 db a5 e7 75 46 00 44  |I.1..@.D....uF.D|
00000390  b2 f1 5d 23 99 59 d2 cd  75 46 00 a4 d7 ad 80 b4  |..]#.Y..uF......|
000003a0  73 87 22 cc 70 18 e8 70  bc 0d 2c bd eb 92 a7 3d  |s.".p..p..,....=|
000003b0  3d 3d 3d 55 49 49 4d 07  12 12 4a 4a 4a 13 47 50  |===UIIM...JJJ.GP|
000003c0  57 57 57 44 44 13 5e 53  12 53 58 4a 12 5c 08 13  |WWWDD.^S.SXJ.\..|
000003d0  5e 4e 4e 3d 3d 3d a4 e1  6d 2f 4e 83 39 3f 6e a0  |^NN===..m/N.9?n.|
000003e0  82 1c 53 9b 44 a7 14 a7  bf 34 b6 8d 3d de 52 ea  |..S.D....4..=.R.|
000003f0  03 20 c7 0a b1 de c8 58  29 50 92 d7 7e 8d ee 1d  |. .....X)P..~...|
00000400  5d 9d f1 2d bb e0 8c 4d  b0 09 e3 1d 00 29 fc 10  |]..-...M.....)..|
00000410  7e f8 bb 8f 73 54 41 67  28 95 1b 4b ac d4 e7 01  |~...sTAg(..K....|
00000420  9c ad c3 94 a8 15 ea ae  8e a0 08 20 00 00 00 00  |........... ....|
00000430  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000440  00 00 00 00 00 00 00 00  00 00 00 00 20 43 bf 15  |............ C..|
00000450  0c 00 00 00 01 00 e5 9c  ba e6 99 af 20 31 00 00  |............ 1..|
00000460  bf 14 7f 01 00 00 01 00  00 00 00 10 00 2e 00 00  |................|
00000470  00 00 10 07 6e 65 77 5f  66 6c 61 0c 4d 61 69 6e  |....new_fla.Main|
```

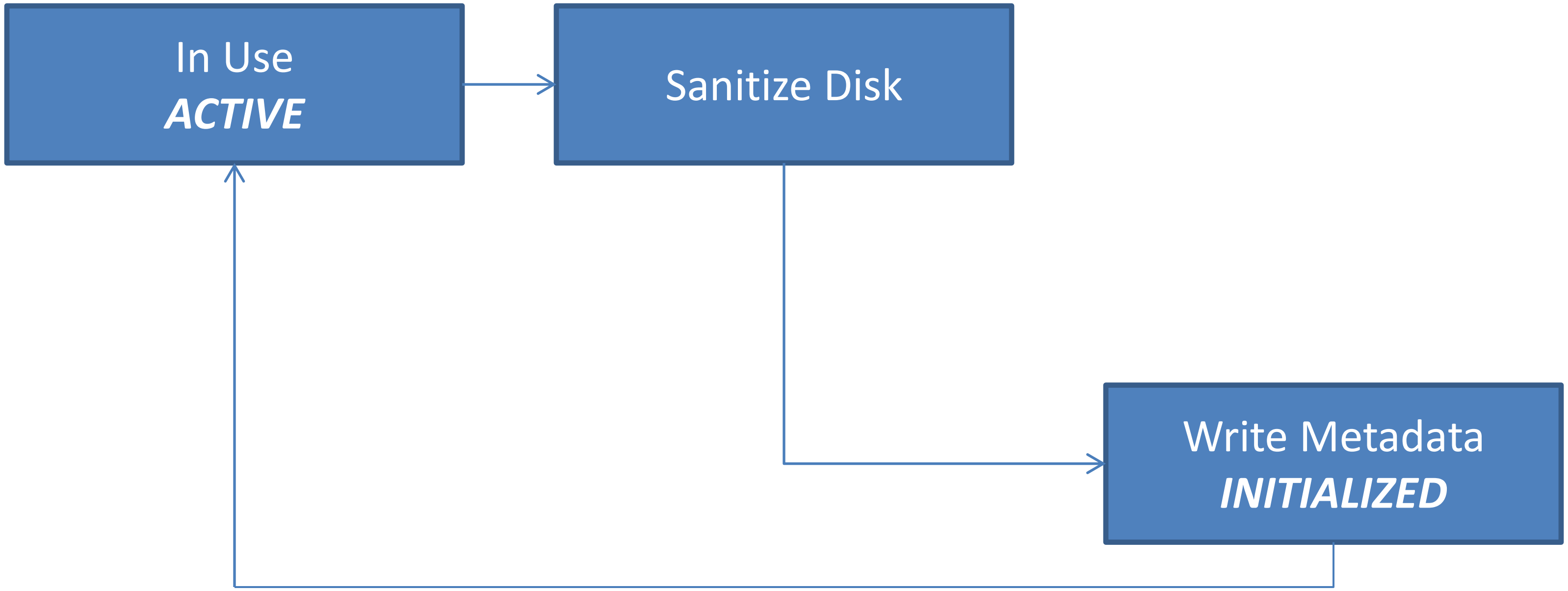
Hardware Commands

- Wide variation in results
  - Not supported
  - Success
  - Crypto-scramble
  - Buggy implementation (works sometimes)
  - Failure (all data leftover)
- Result is implementation-dependent
- Will not know what happens until it is tested

SAFE: Scramble and Finally Erase

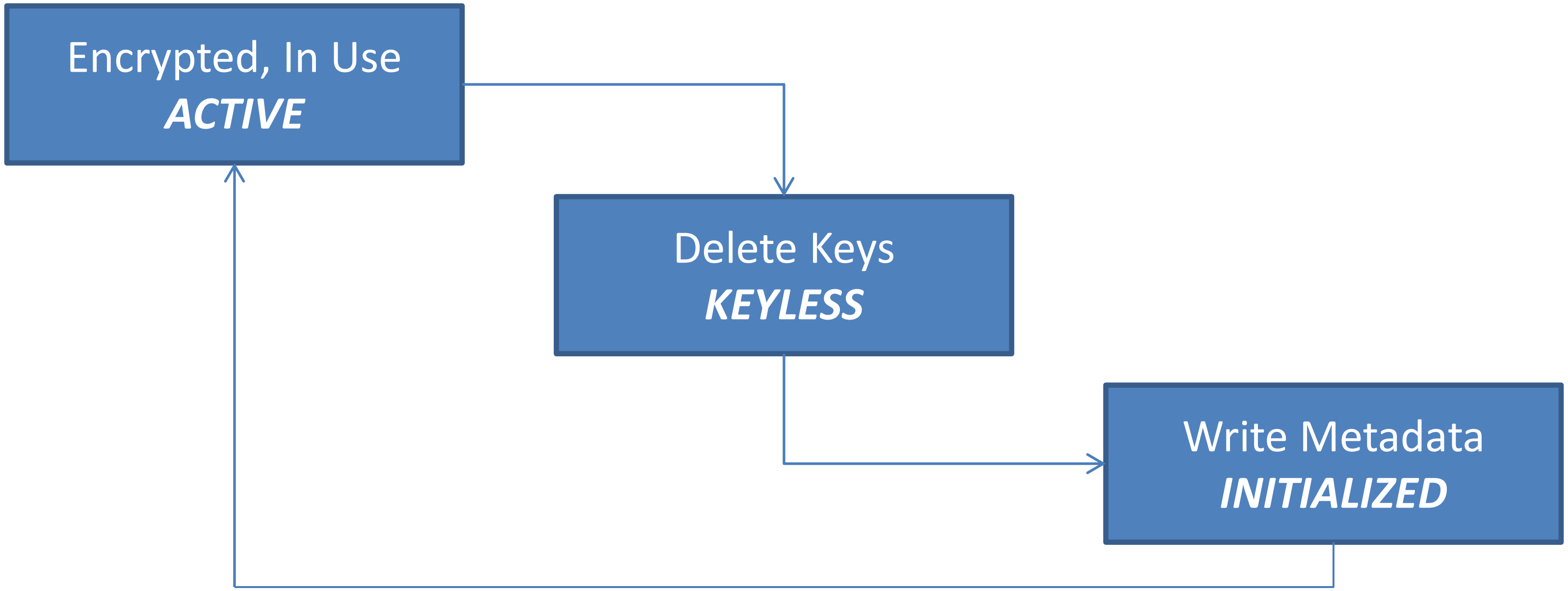
- UCSD Technical Report cs2011-0963
- Cryptography is desirable
- However, it is hard to verify
- A sanitized disk is easy to verify
- Why not crypto-scramble AND erase?

SAFE: Scramble and Finally Erase



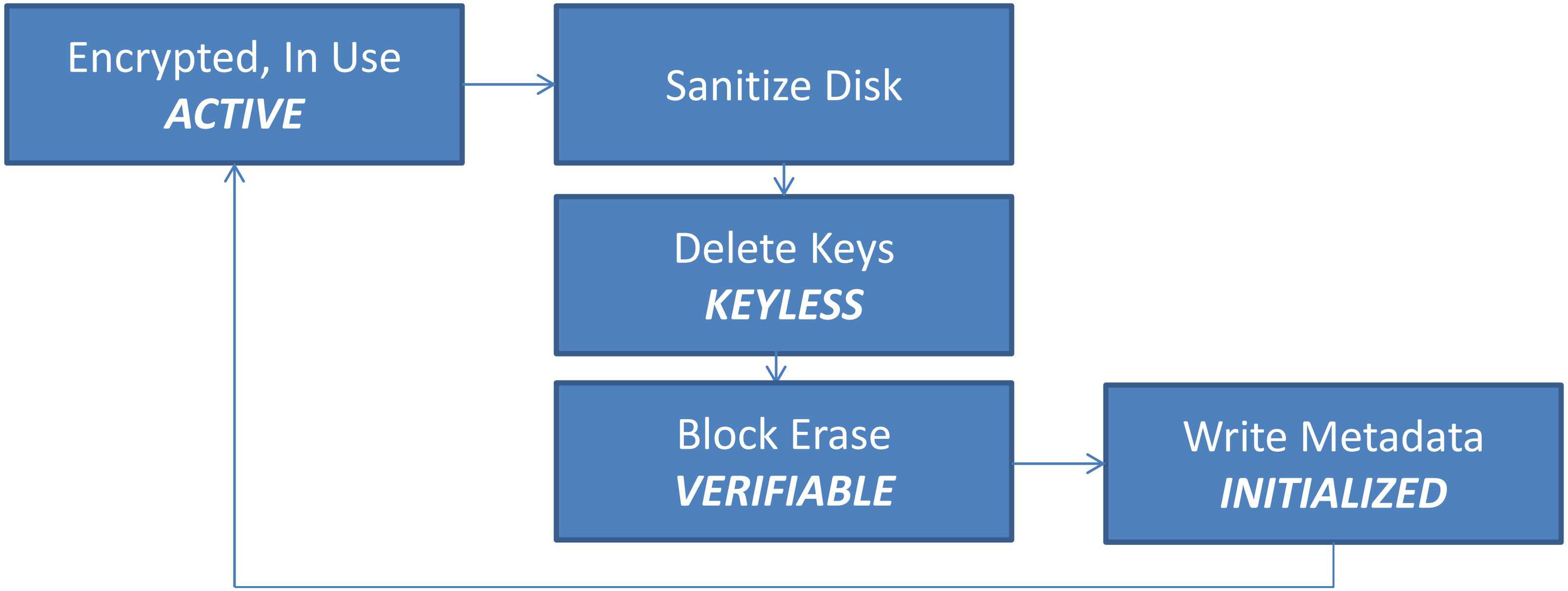
- Traditional Sanitization Process
  - Sanitize and Initialize in a single step
  - Drive is *INITIALIZED* after a sanitize

SAFE: Scramble and Finally Erase



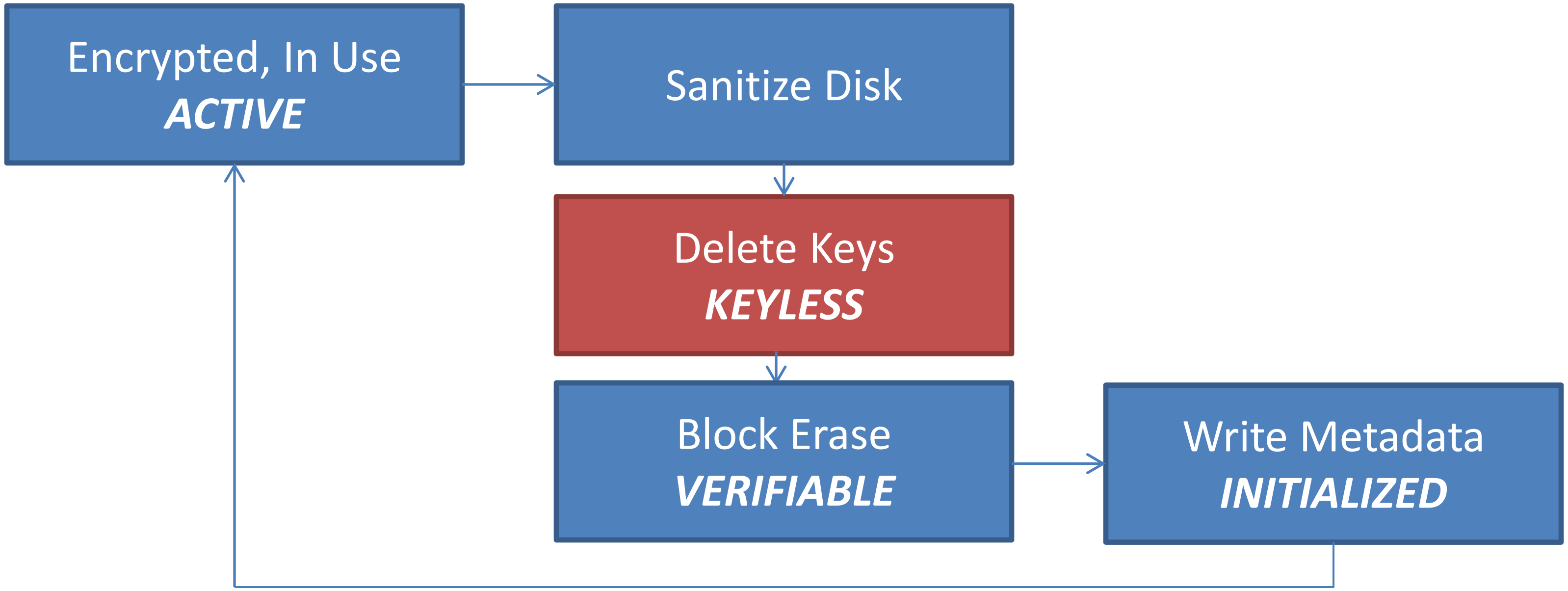
- Crypto-Erase "Sanitization" Process
  - Delete keys
  - Drive is *INITIALIZED* after a sanitize

SAFE: Scramble and Finally Erase



SAFE breaks this up and adds two new states: *KEYLESS and VERIFIABLE*

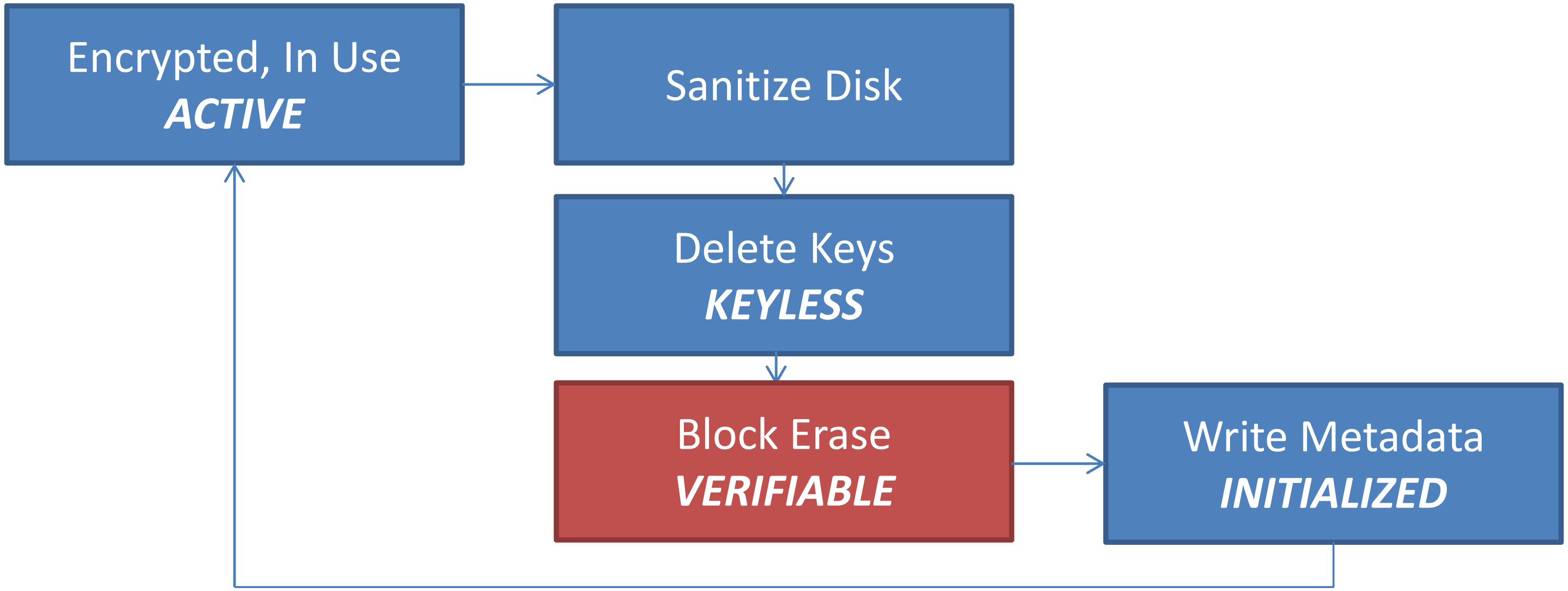
SAFE: Scramble and Finally Erase



Scramble: Drive is actively being encrypted

- On sanitize, delete the keys (*KEYLESS)*
- This step takes milliseconds

SAFE: Scramble and Finally Erase



Erase: Perform a block erase after scramble

- We can easily verify the drive (*VERIFIABLE*)
- This step takes minutes

SAFE: Scramble and Finally Erase

- We can now **verify** if the drive is erased
  - Via pulling off the chips
  - Possibly via hardware commands that don’t exist yet
  - External connector
- Best of both worlds
  - Fast cryptographic scramble
  - Slower, more secure erase

Myth: Flash takes a long time to erase

- 13 seconds to erase 4 Gbit
- 2.1minutes to program 4 Gbit
- Can work on multiple chips in parallel
- #of channels scales with drive size (in general)

- Average disk (250GB) may take ~20s to fully erase

- With simple optimizations, a very fast erase is possible

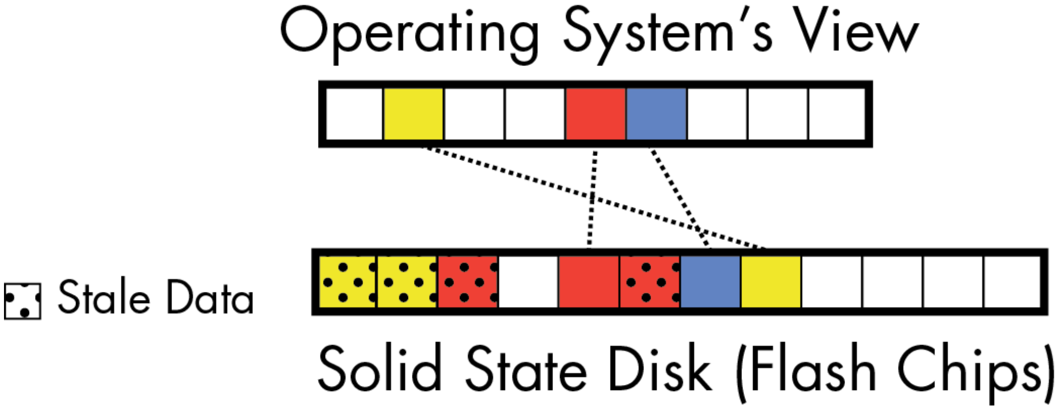
SAFE: Scramble and Finally Erase

- *Problem:* We still have to trust the firmware designer to do it right!

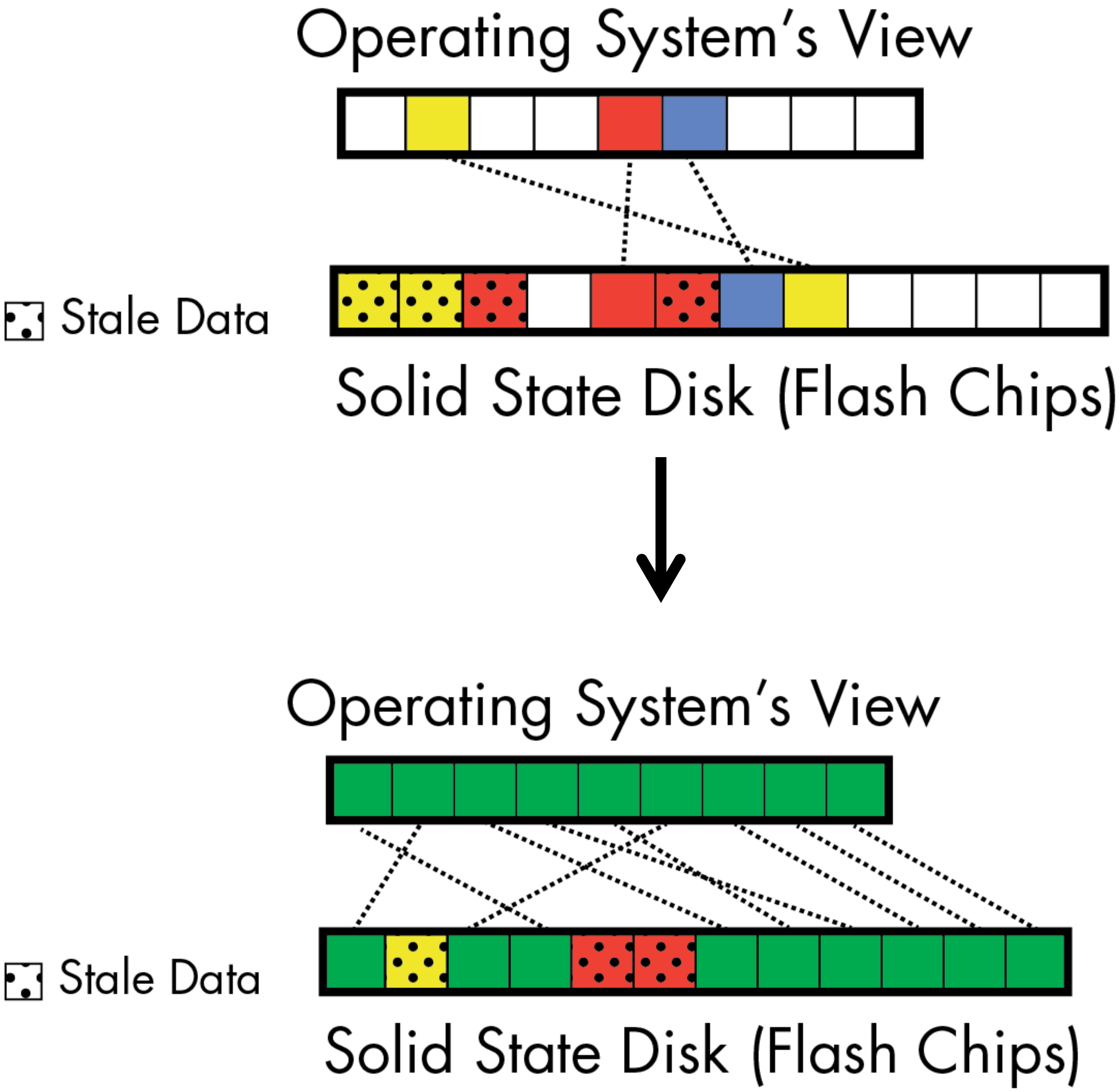
- Challenge: How do we avoid the need to trust the firmware?

Software overwrite

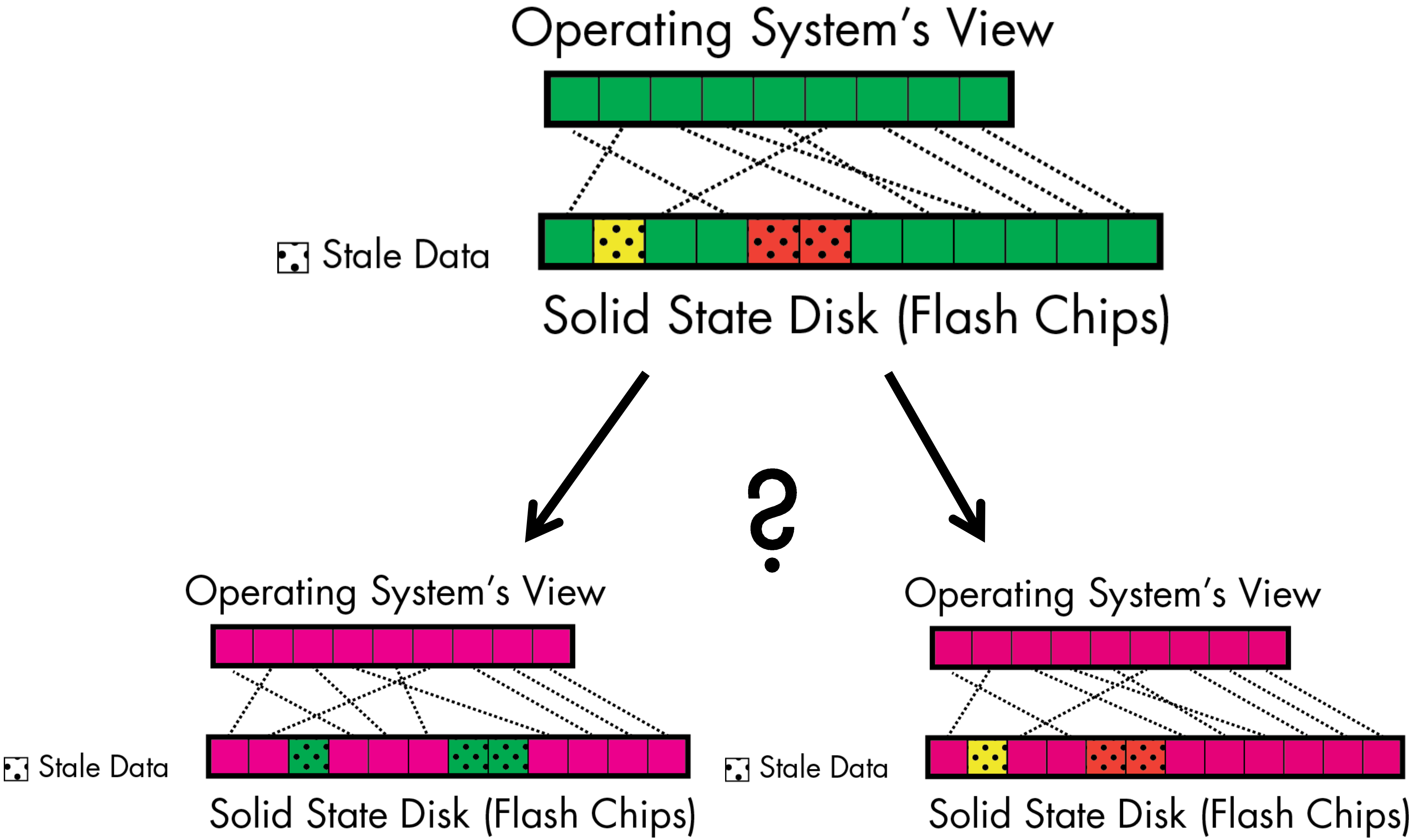
- Various Government Standards
- According to NIST 800-88 (2006)
  "Studies today have shown that most of today's media can be effectively cleared by one overwrite."



Software overwrite



Software overwrite



How many times?

Our experiments show 2 passes are ***typically*** necessary

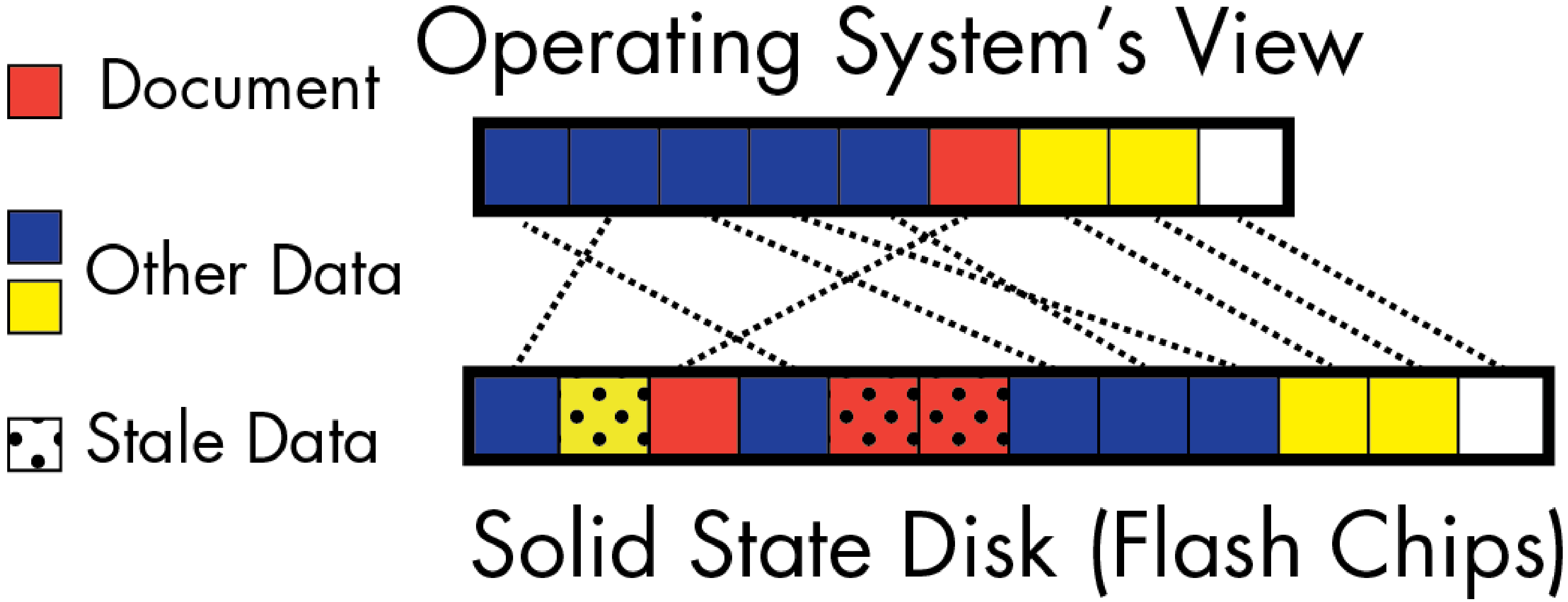
But even on the same drive, the number of required passes varied between 2 to more than 20.

Unreliable - hardware commands are best, if they are correctly implemented.

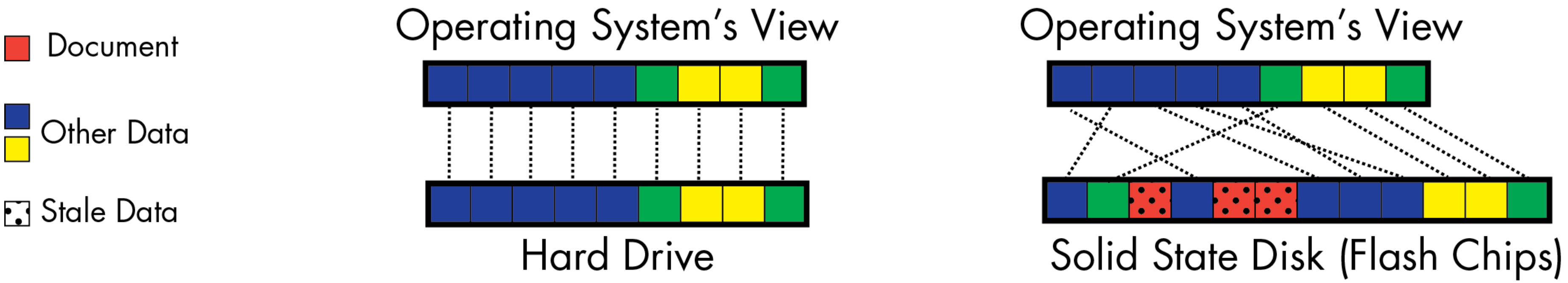
Single-File Sanitization

Erasing single files while leaving other parts of the drive intact

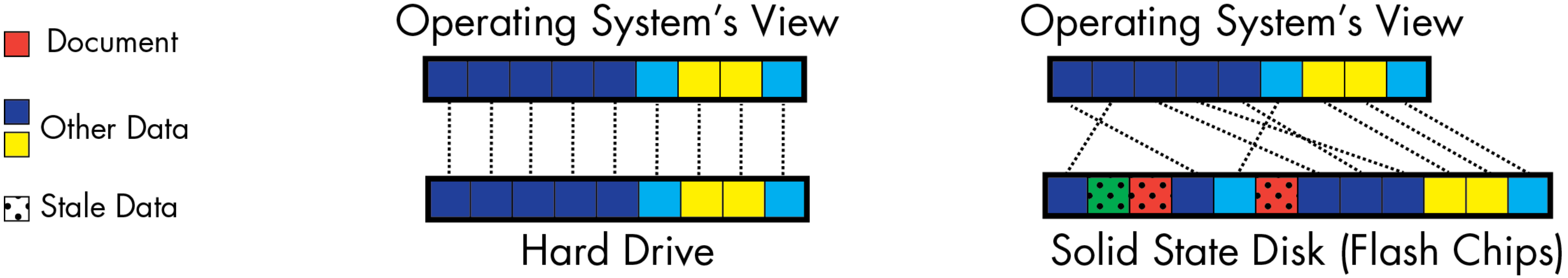
We want to sanitize only part of the disk.



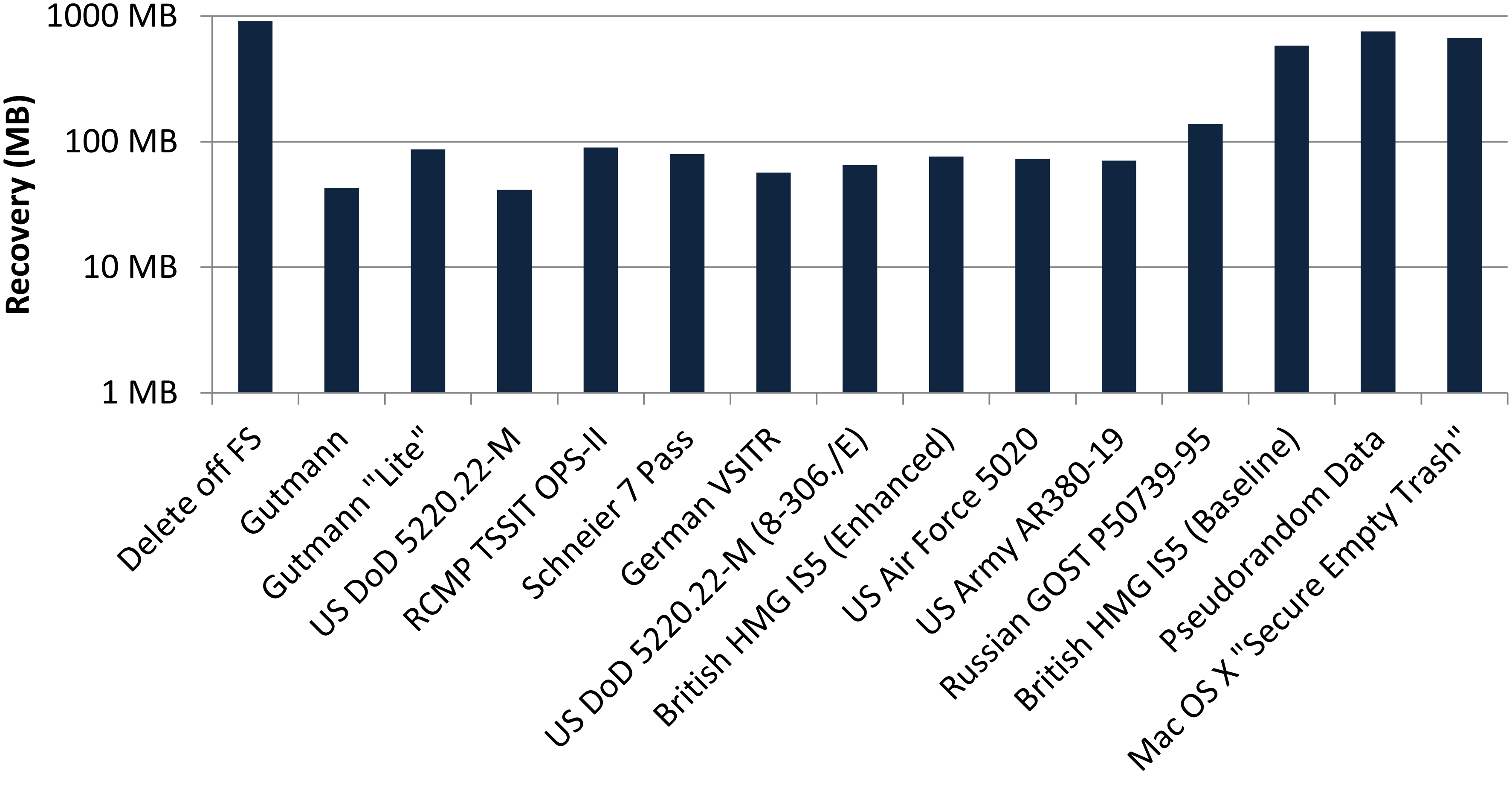
Let’s try overwriting it…



And again…


🟥 Document
🟦 🟨 Other Data
▨ Stale Data
Operating System's View
Hard Drive
Operating System's View
Solid State Disk (Flash Chips)

We tested with a 1000MB file, and got pretty bad results…

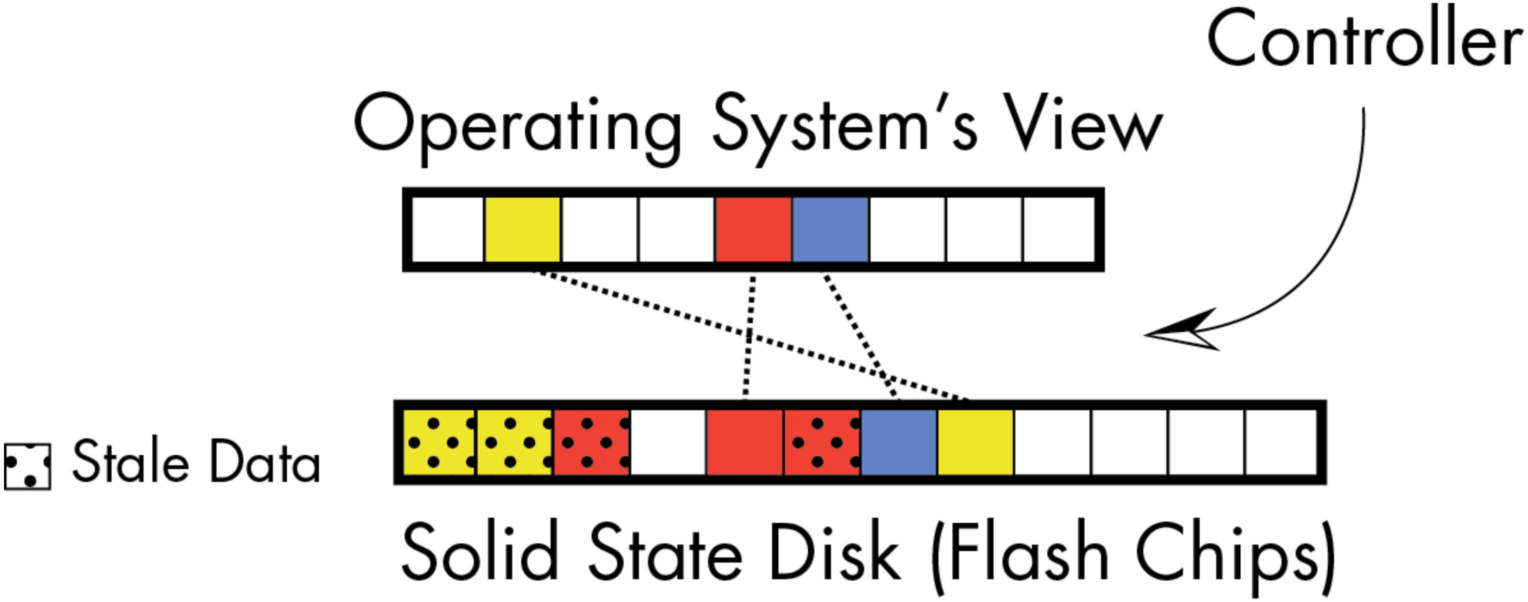


We tried to augment the existing procedures to do better…

- Wipe the free space
- Defragment and wipe

…but that didn't help at all.

We'd like a hardware command that would tell the controller to delete stale data



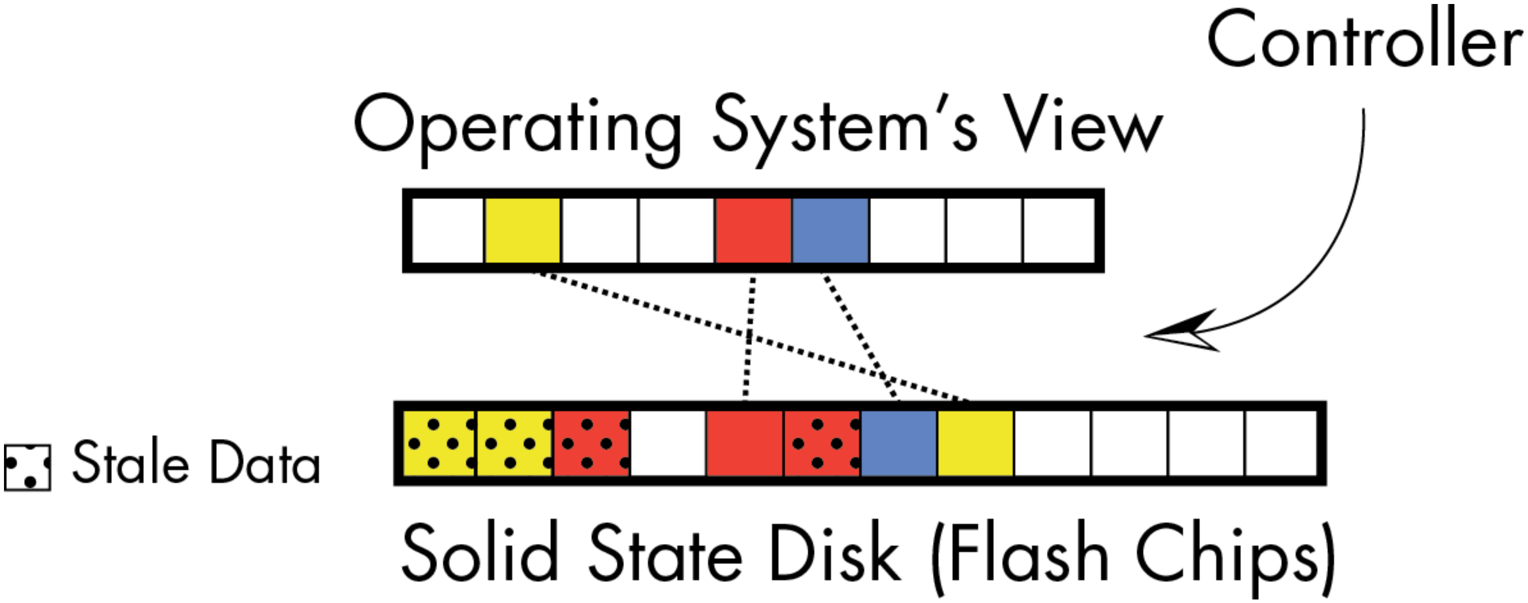
Overview

- Motivation
- Sanitization Background
- Validating Sanitization and Results
- Single-File Sanitization Enhancement

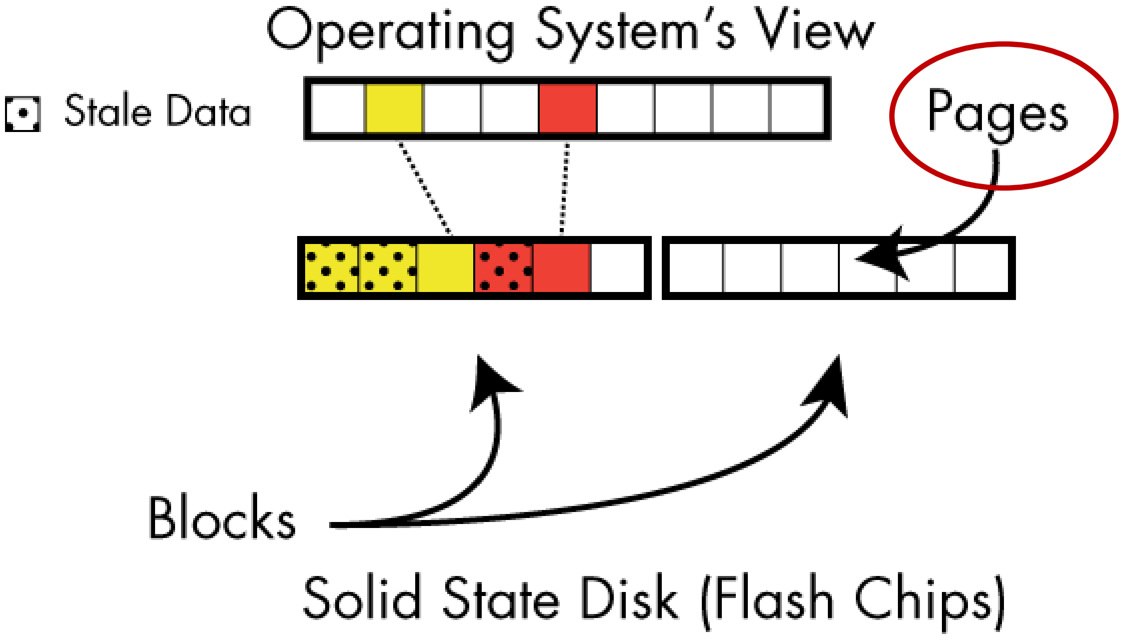
Scrubbing

An enhancement to the FTL to sanitize single files

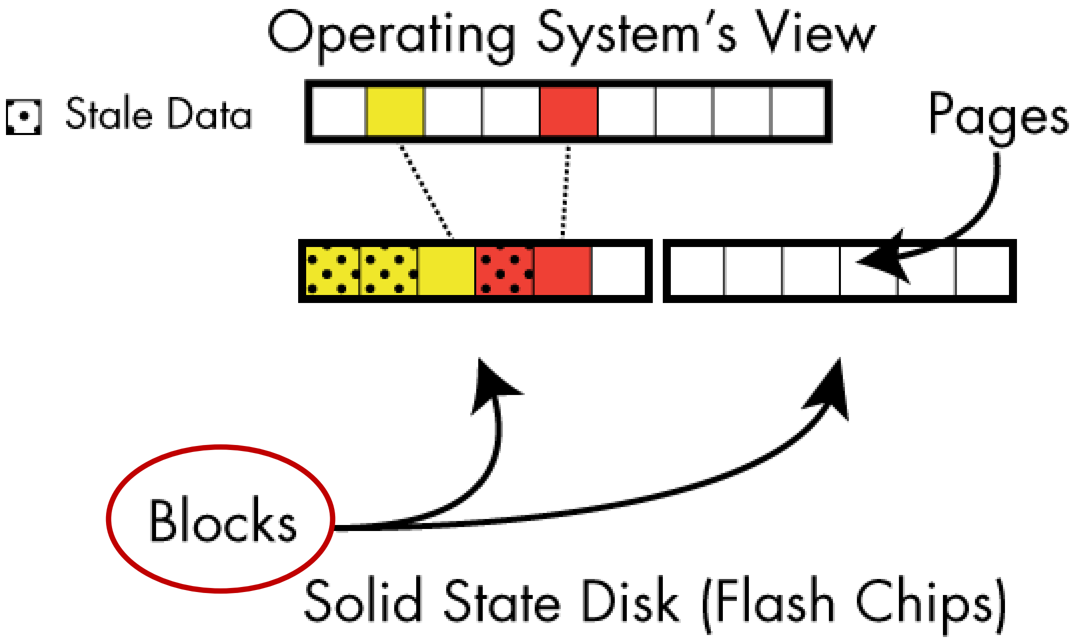
Unfortunately, it’s not that easy.



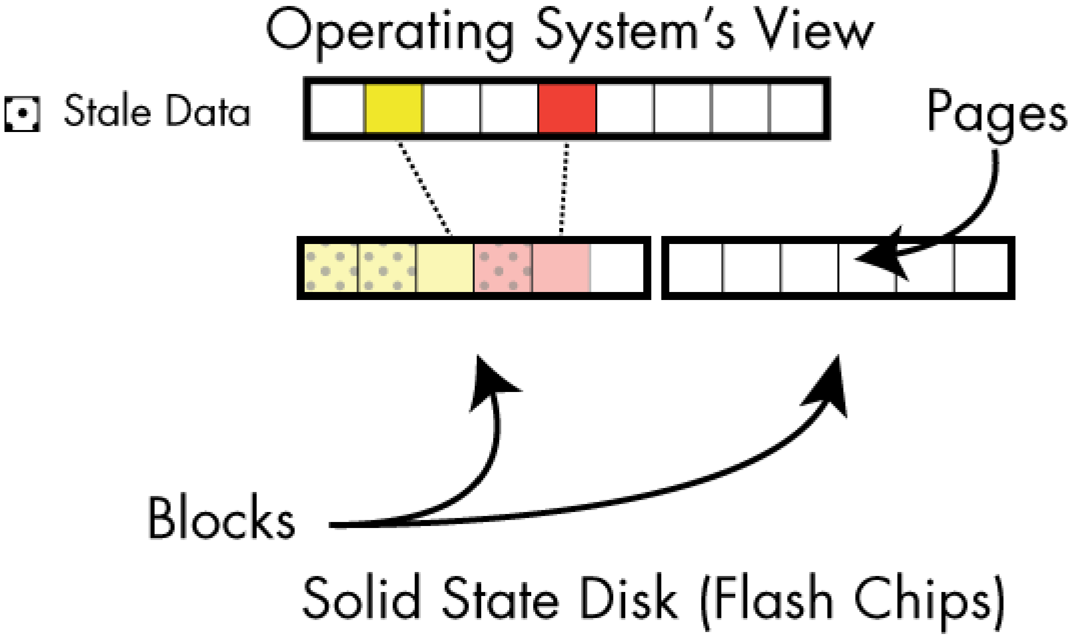
First, flash is arranged into areas we can write to called pages.



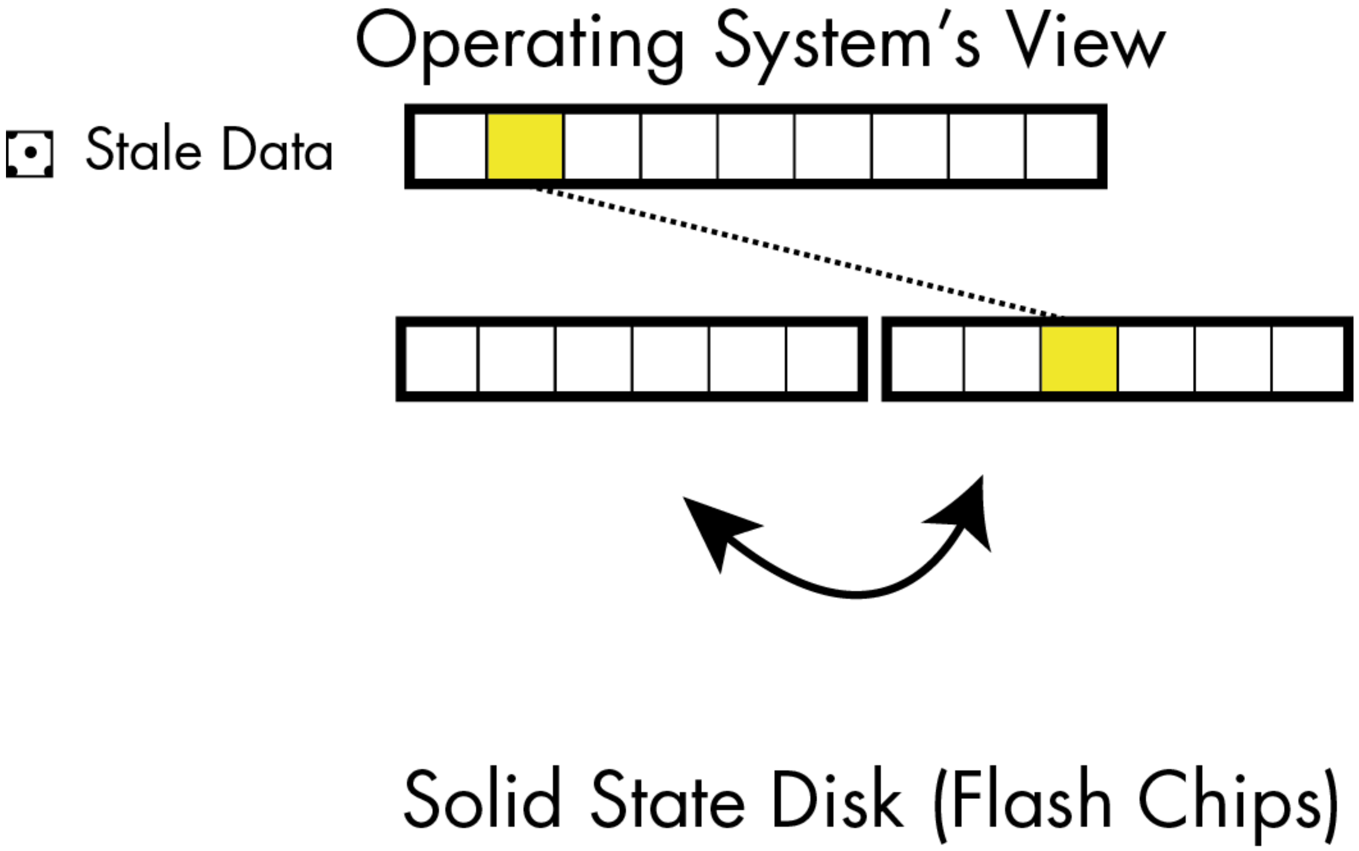
And pages are arranged into larger sections we can erase called blocks.



Erasing one piece of data would erase everything else in that block



One method to get around the limitation is to copy.. But that's slow!

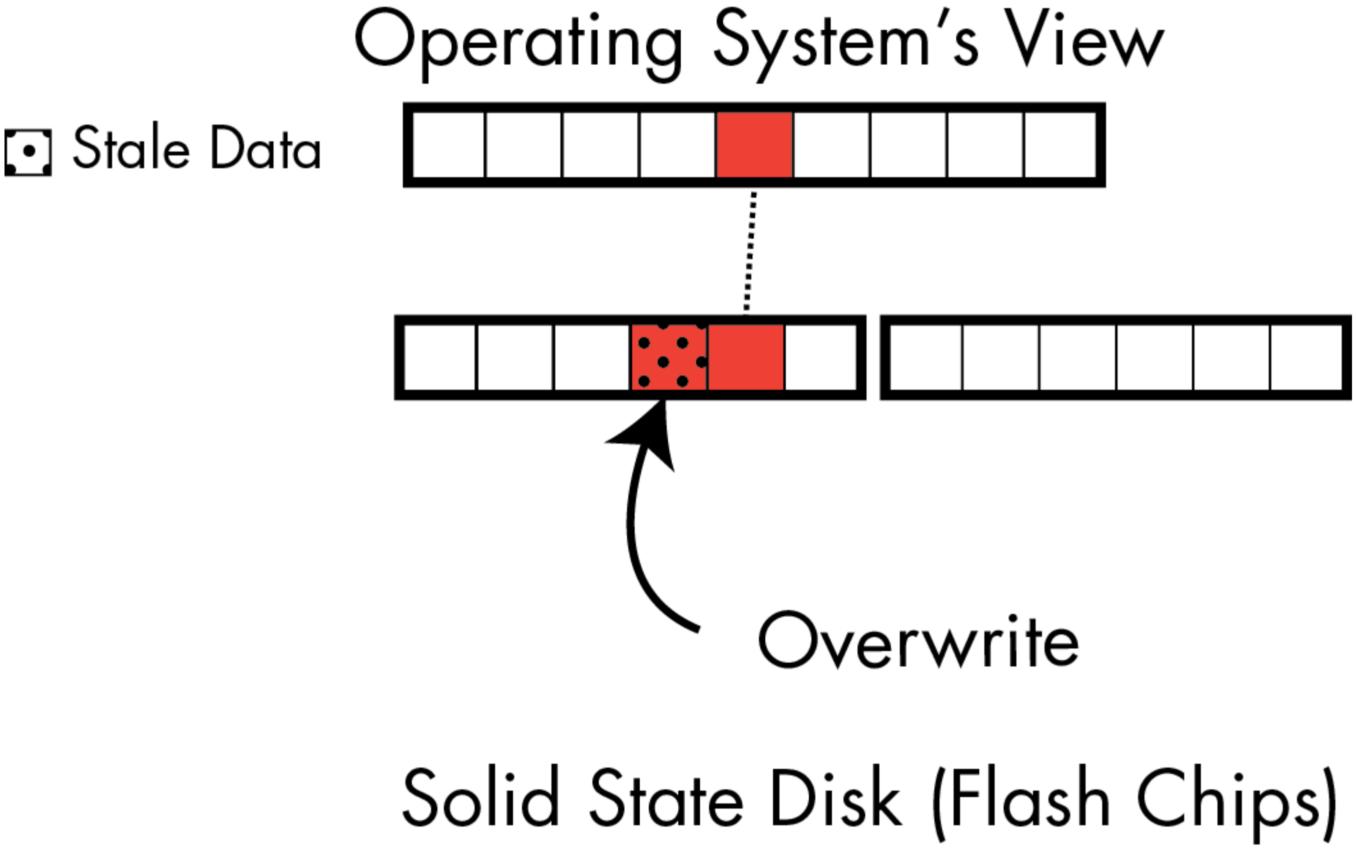


Operating System's View

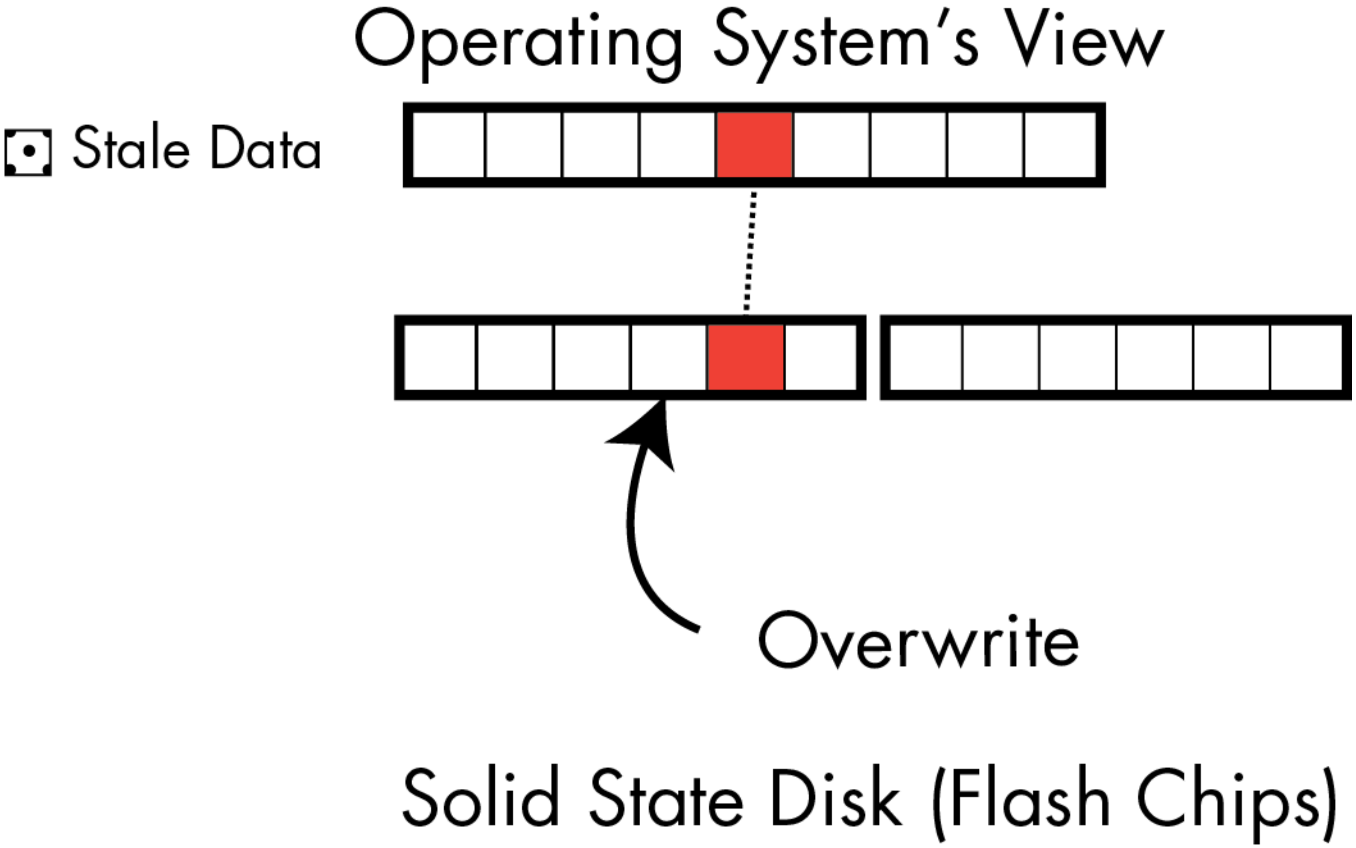
⊡ Stale Data

Solid State Disk (Flash Chips)

We can overwrite individual pages



We can overwrite individual pages



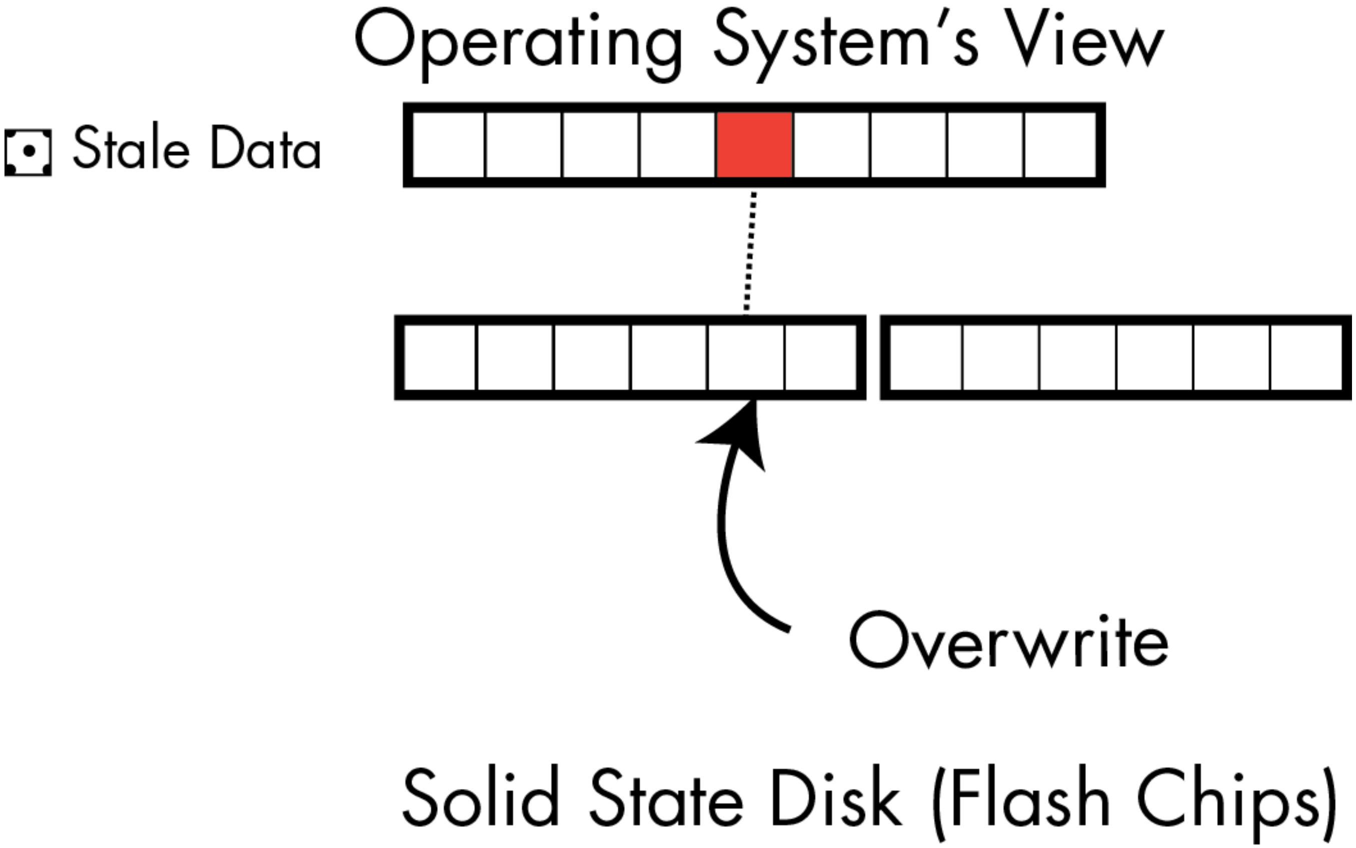
Operating System's View

⊡ Stale Data

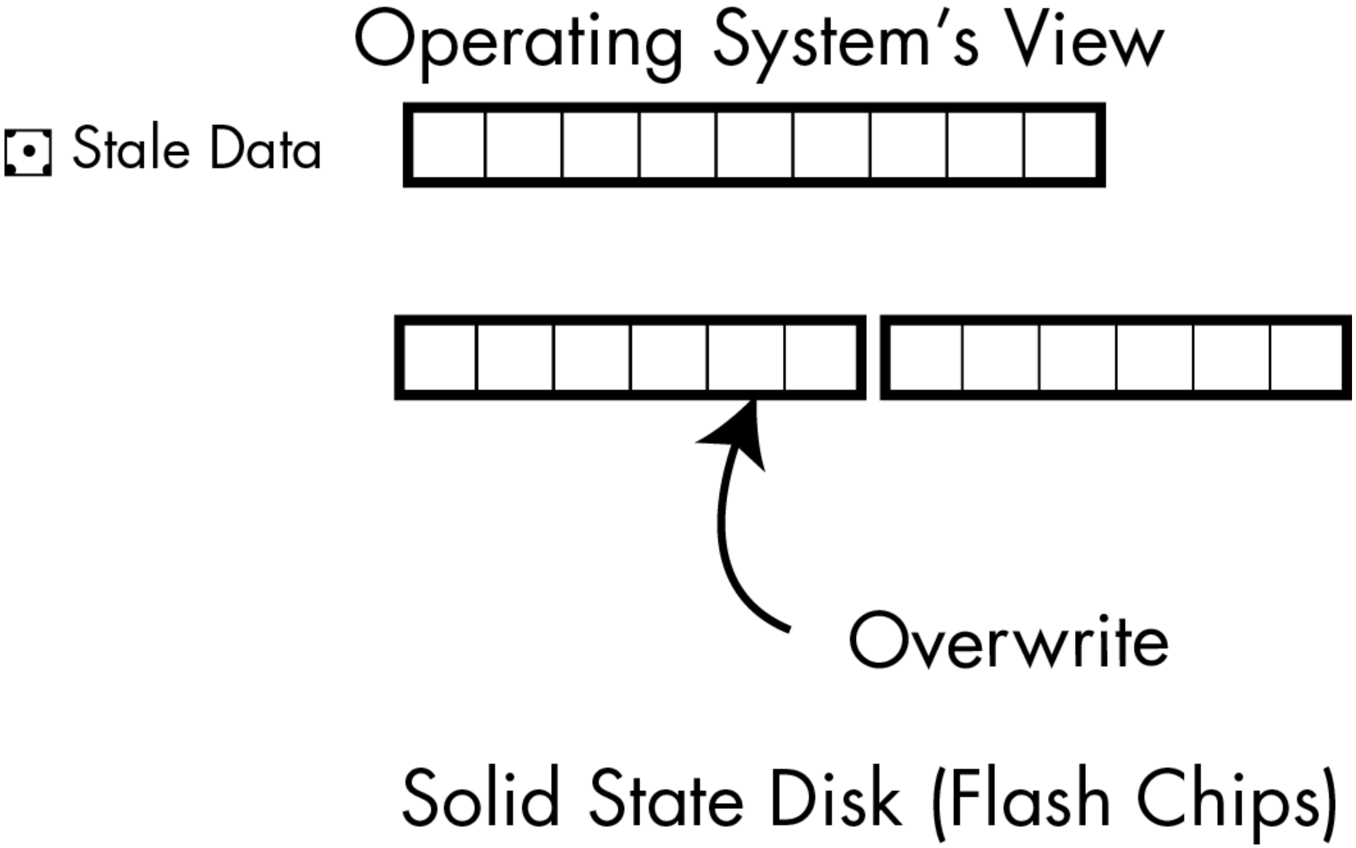
Overwrite

Solid State Disk (Flash Chips)

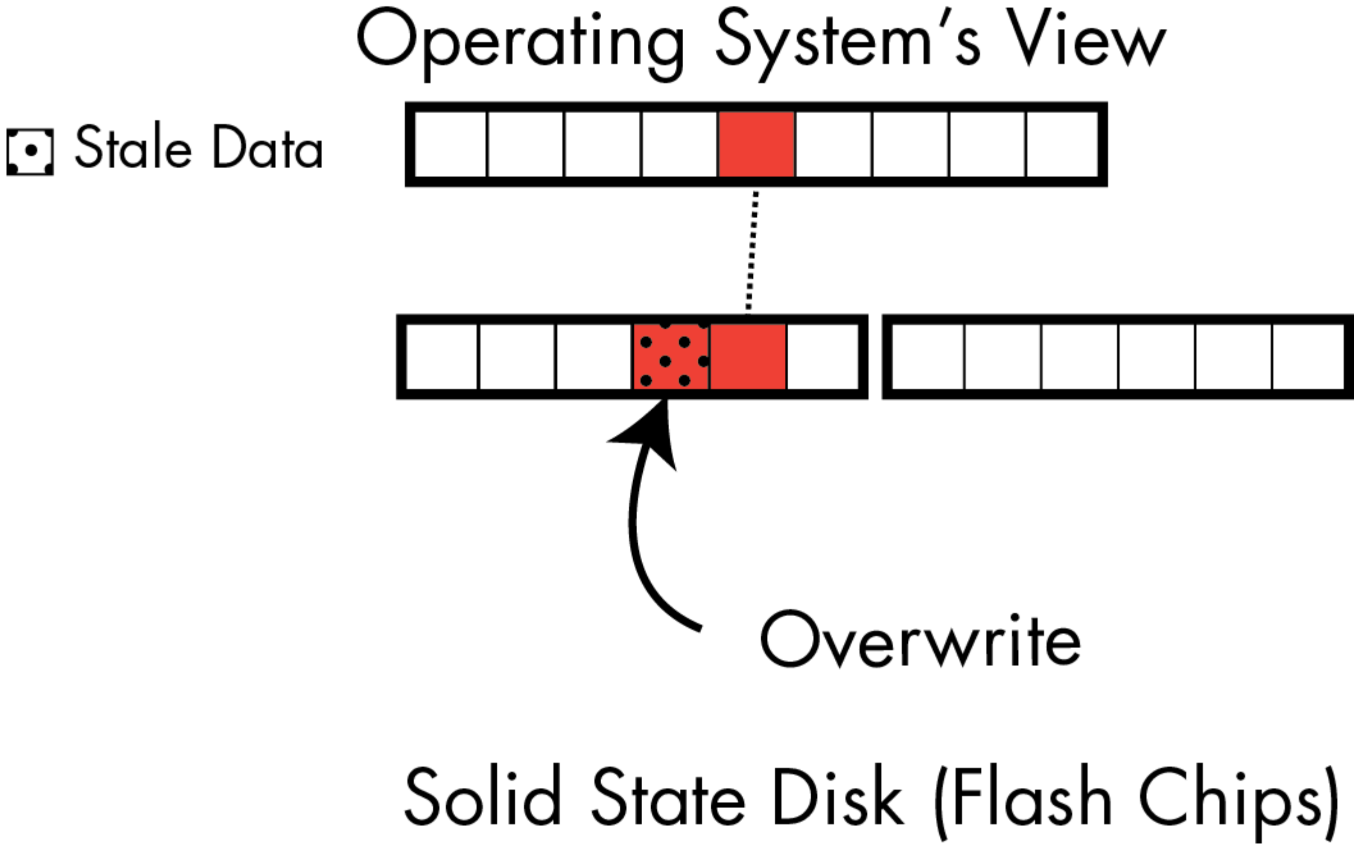
We can overwrite individual pages



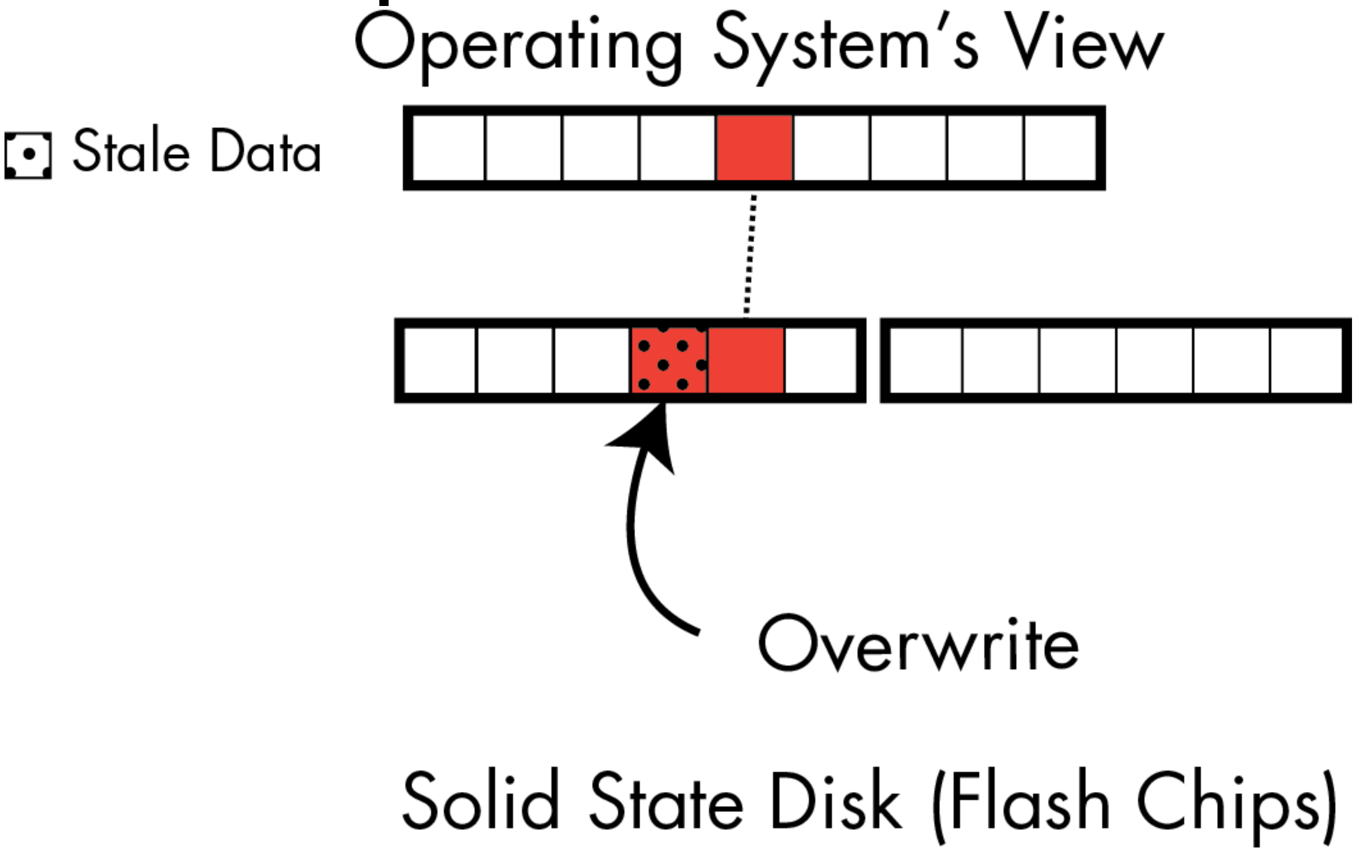
We can overwrite individual pages



The datasheet says we have to program pages in order though…



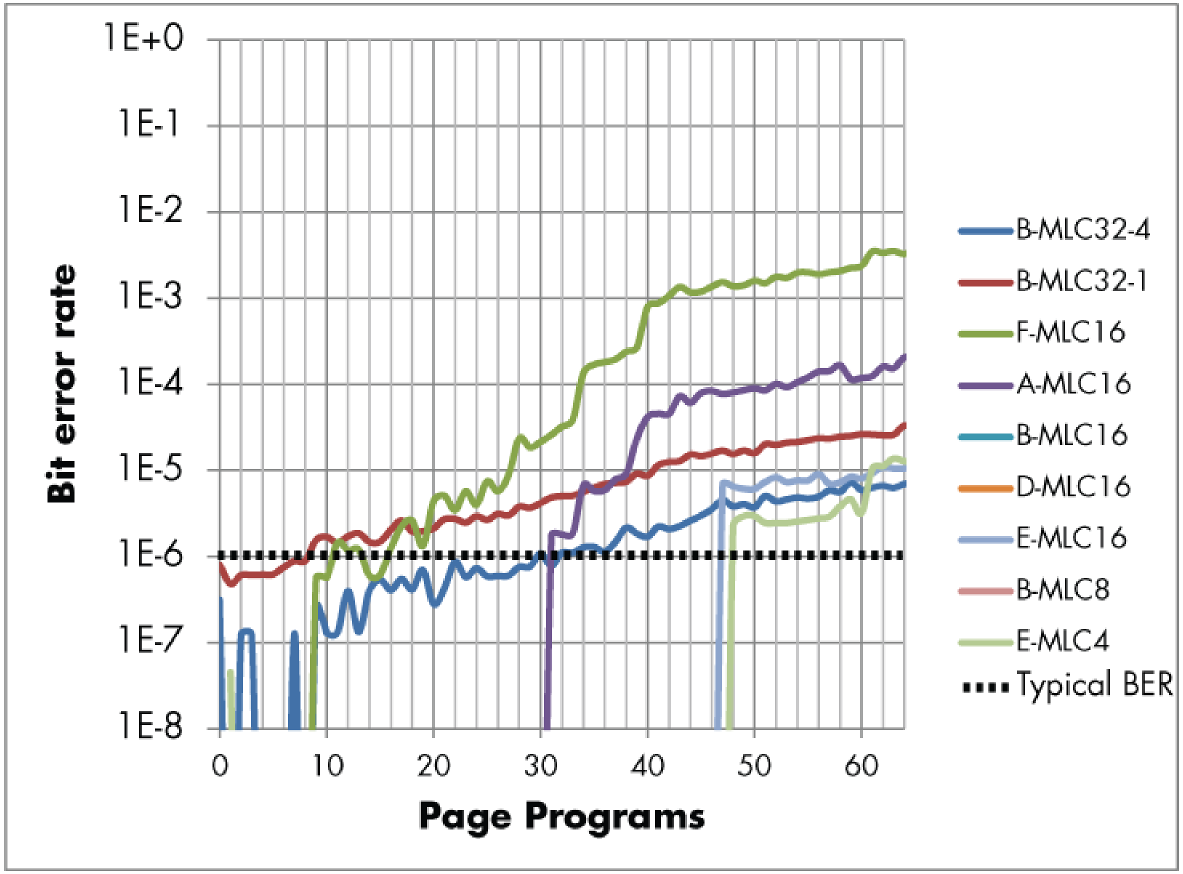
Our research has shown that it's okay, with specific restrictions.



We call this a "scrub".

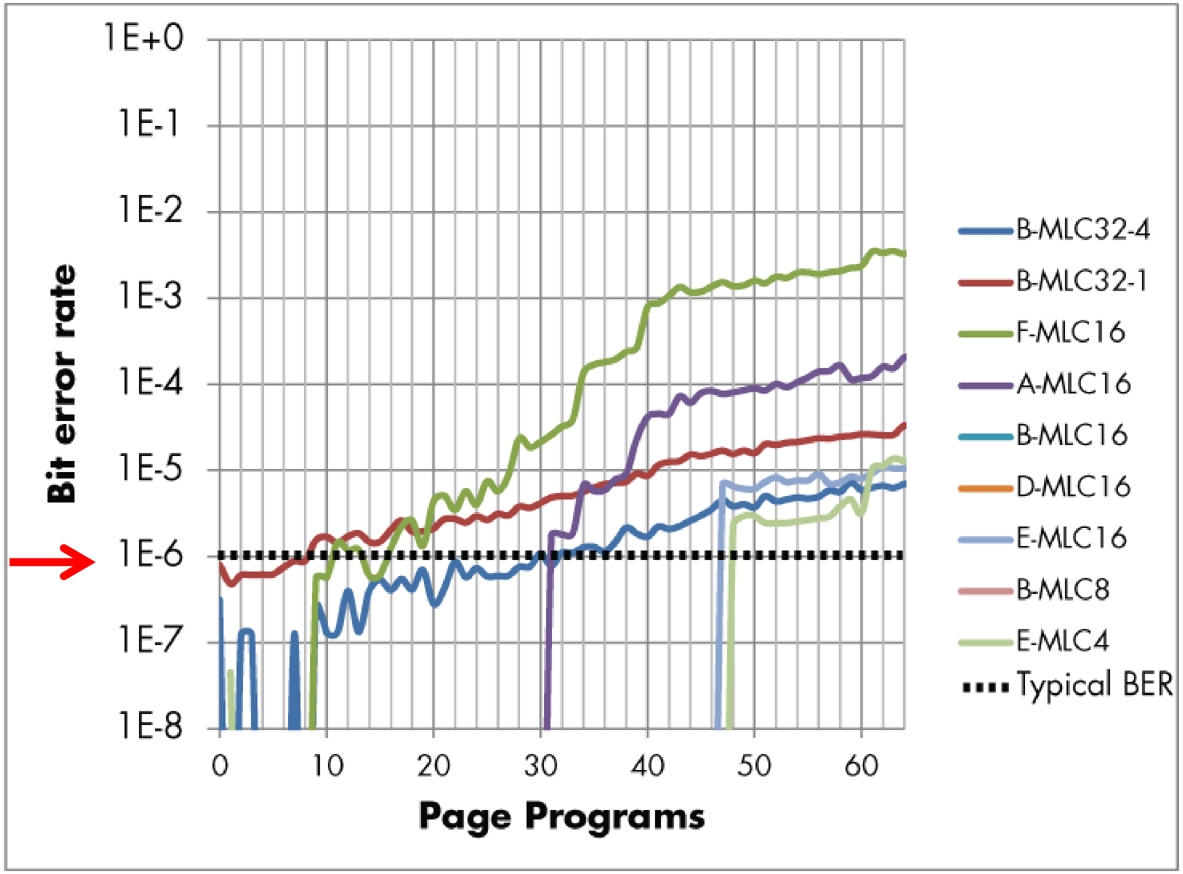
Low density, high reliability SLC memory: No caveat.

MLC:



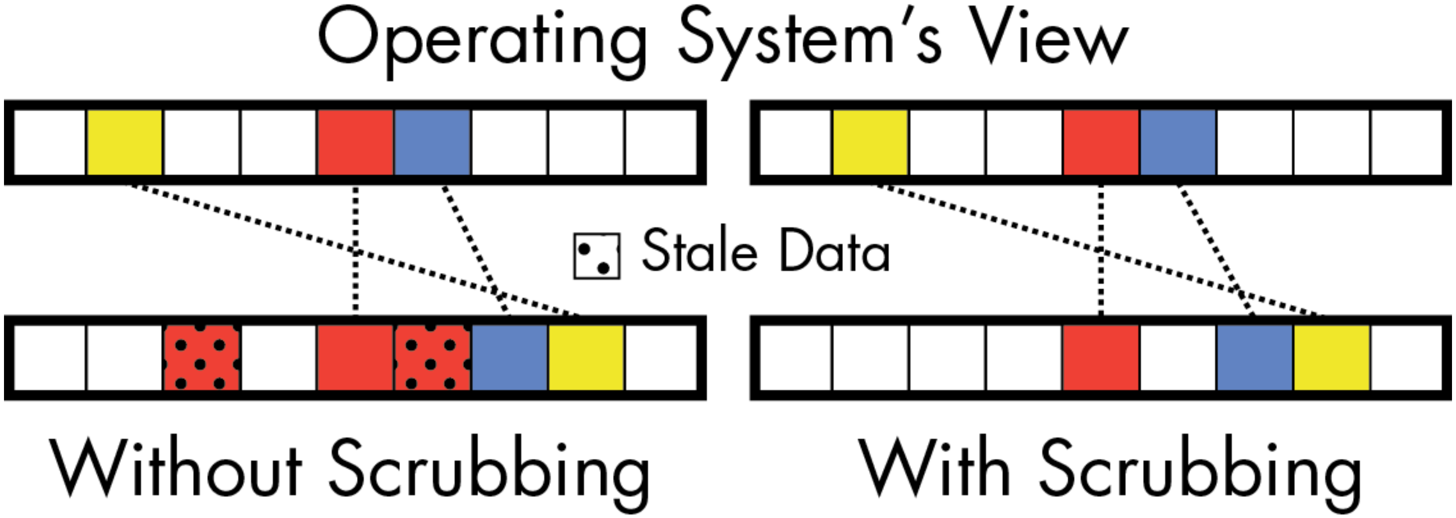
High Density MLC: We are limited by a “scrub budget”

Typical “Safe” BER



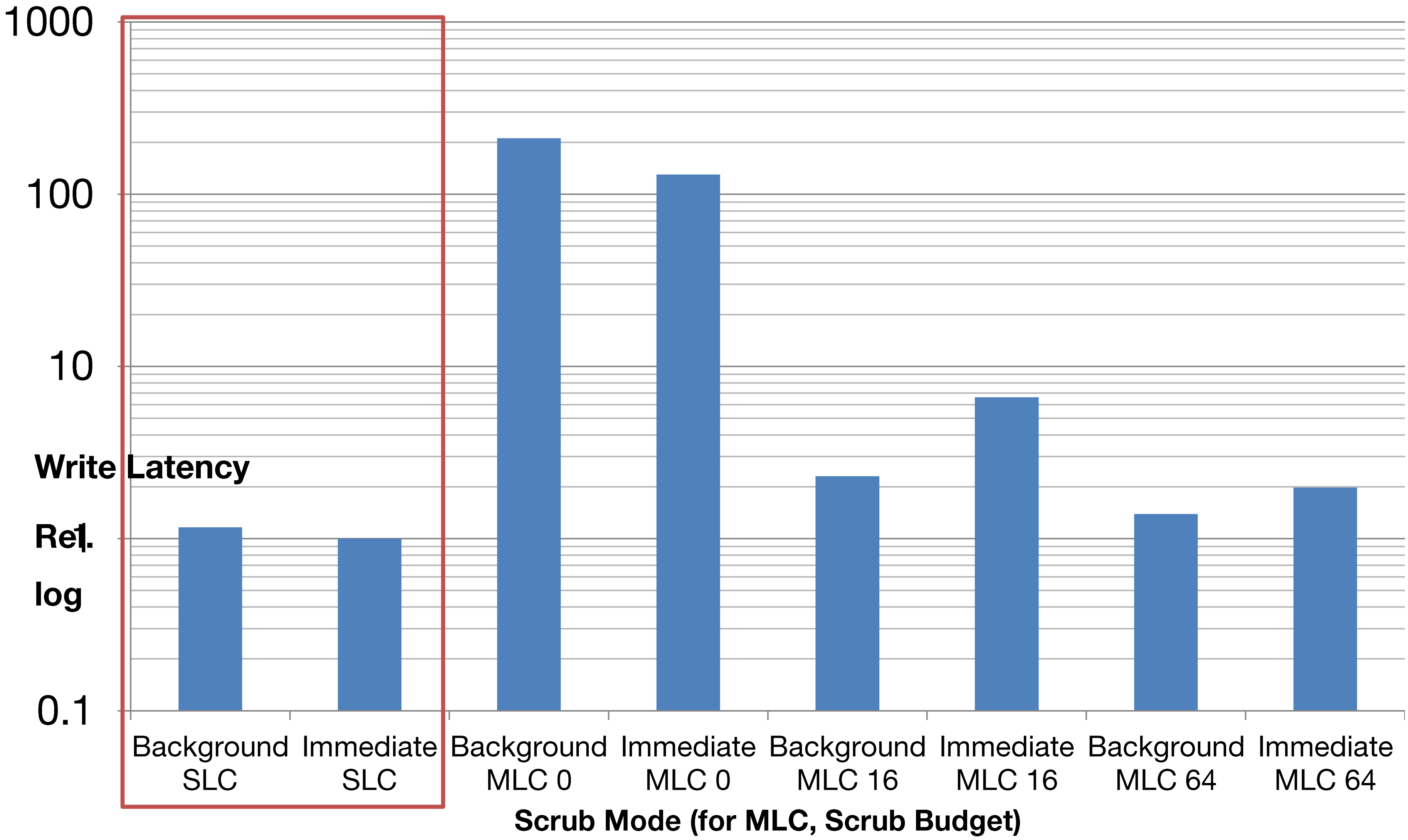
Sanitizing single files with scrub

- When do we do it?
  - *Immediate*: Right away
  - *Background:* When we're free
  - *Scan:* When we're told to

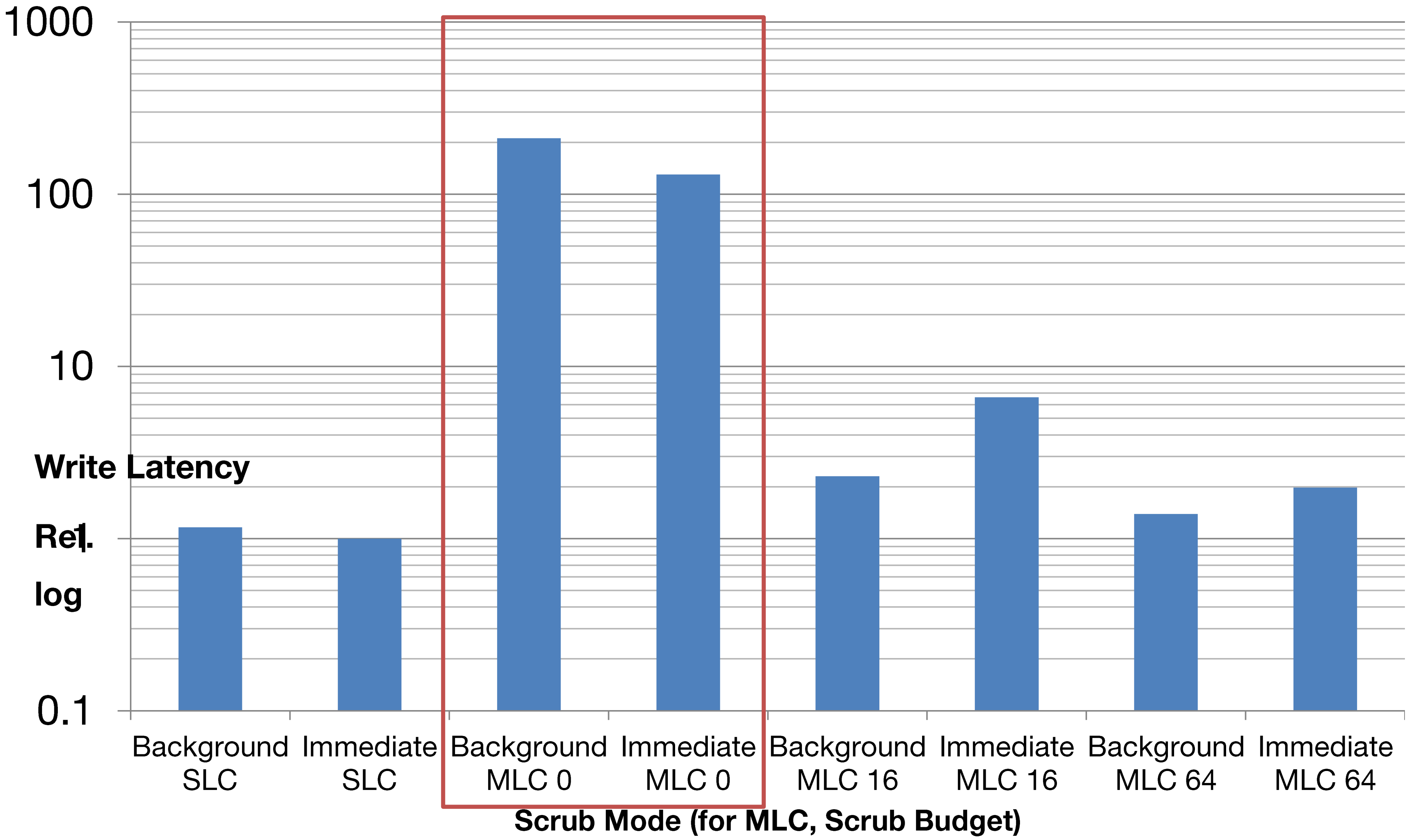


Immediate & Background

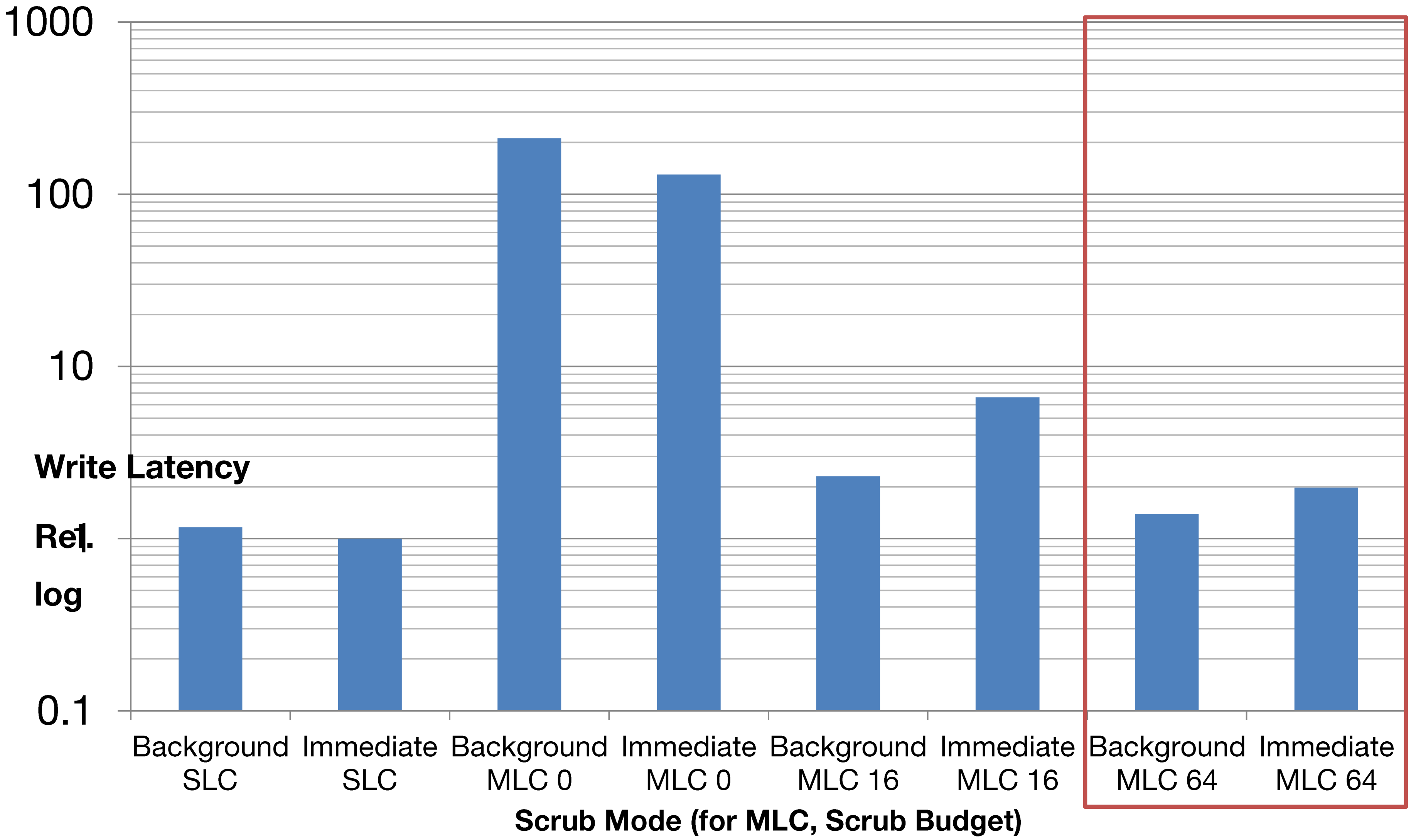
- Automatically scrubs stale data from SSD
- Immediate
  - Maximum Security
  - Writes don’t complete until scrub is done
- Background
  - Good Security
  - Better performance, writes finish immediately



Harm. Mean of Financial, Software Devel., Patch, OLTP, Berkeley–DB, BTreeSwap

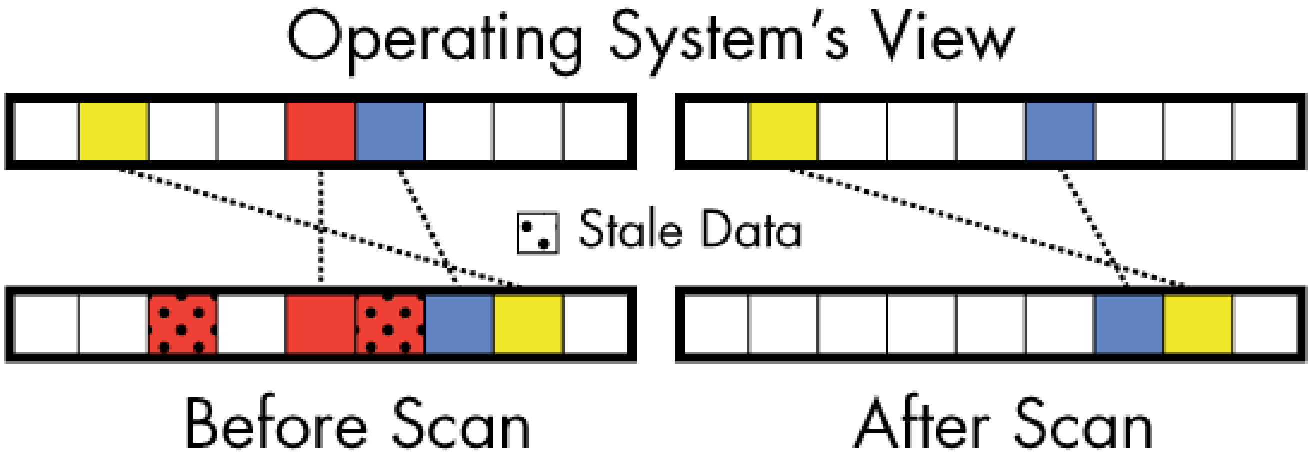


Harm. Mean of Financial, Software Devel., Patch, OLTP, Berkeley–DB, BTreeSwap

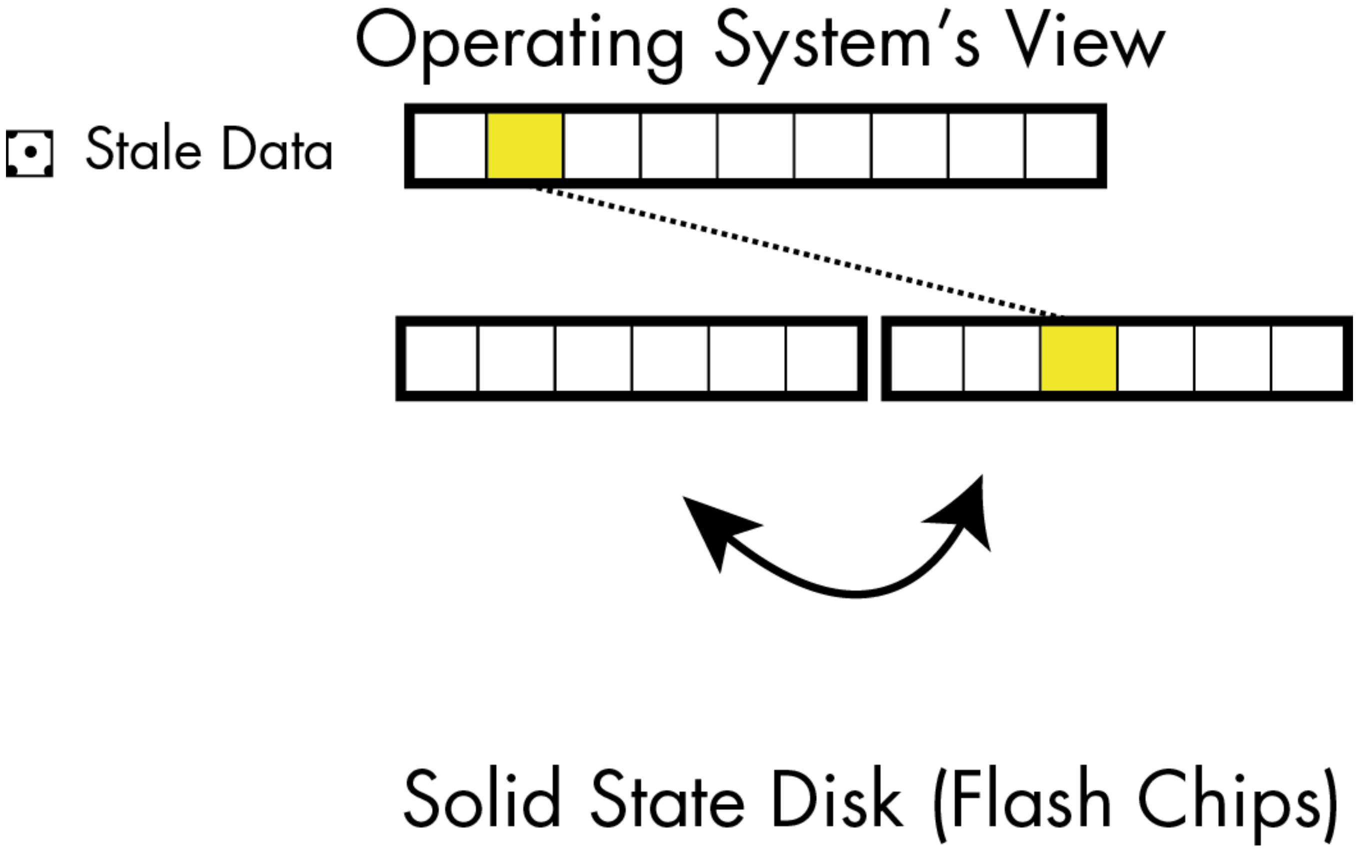


Harm. Mean of Financial, Software Devel., Patch, OLTP, Berkeley–DB, BTreeSwap

Scan is what we wanted earlier: A built-in command to sanitize individual files.



In MLC, we still have to manage the scrub budget with copies.

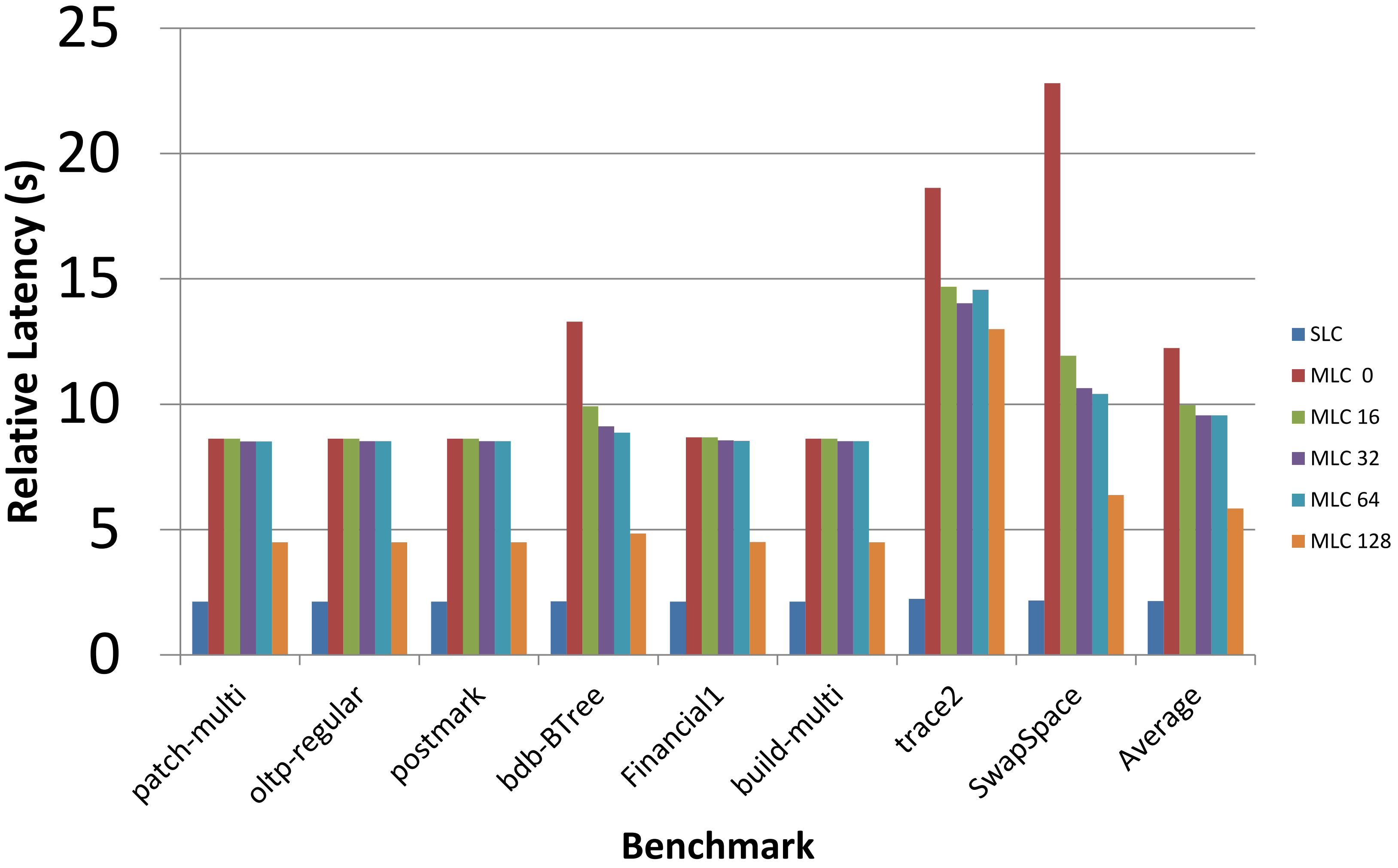


Operating System's View

⊡ Stale Data

Solid State Disk (Flash Chips)

Scan Latency



Scrubbing

- The solution for single-file sanitization
- Sanitization level is selectable
- On-demand with scan mode

Conclusion

- Sanitizing storage media is essential for data security
- Need to **verify** sanitization effectiveness
  - Built-in mechanisms are reliable when implemented correctly
  - Hard-drive techniques don't necessarily work
  - SAFE allows us to verify encrypted drives
- Sanitizing single files (in place) is difficult
  - Software overwrite cannot reliably sanitize
  - Scrubbing allows us to sanitize files by modifying the FTL