Enabling Data Recovery from Malicious Attacks in Mobile Devices

MTU CS Cybersecurity Reading Group Spring 2020

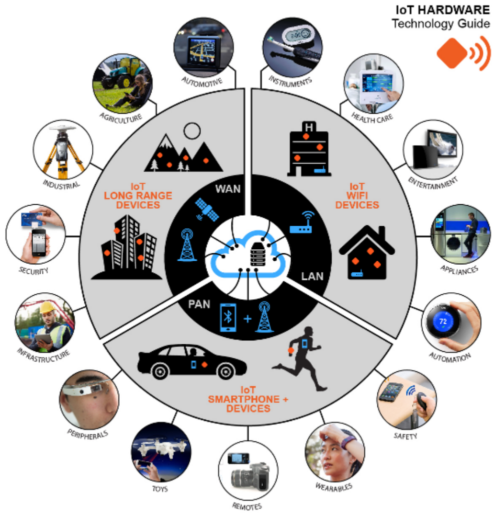
Bo Chen, PhD

Assistant Professor

Department of Computer Science, Michigan Technological University

Flash Memory Storage is Here and There

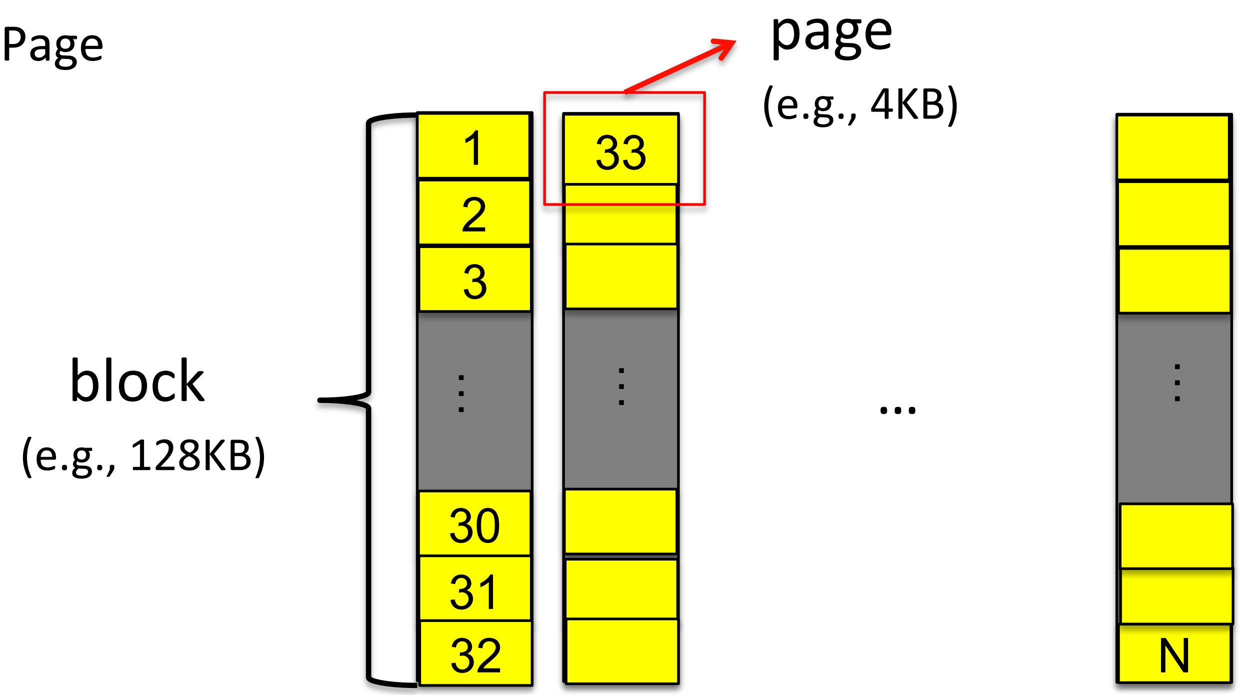
- People are increasingly turning to flash memory for data storage due to its high throughput and decreasing price
  - Solid state drives (SSDs) in laptops and desktops
  - eMMC cards, SD/miniSD/microSD cards in mobile devices
  - USB drives
  - Storage of IoT devices



NAND Flash Memory

- Flash memory
  - NAND flash (broadly used for mass-storage of mobile devices)
  - NOR flash (used for storing program code that rarely needs to be updated, e.g., a computer's BOIS)

- NAND flash organization
  - Block
  - Page

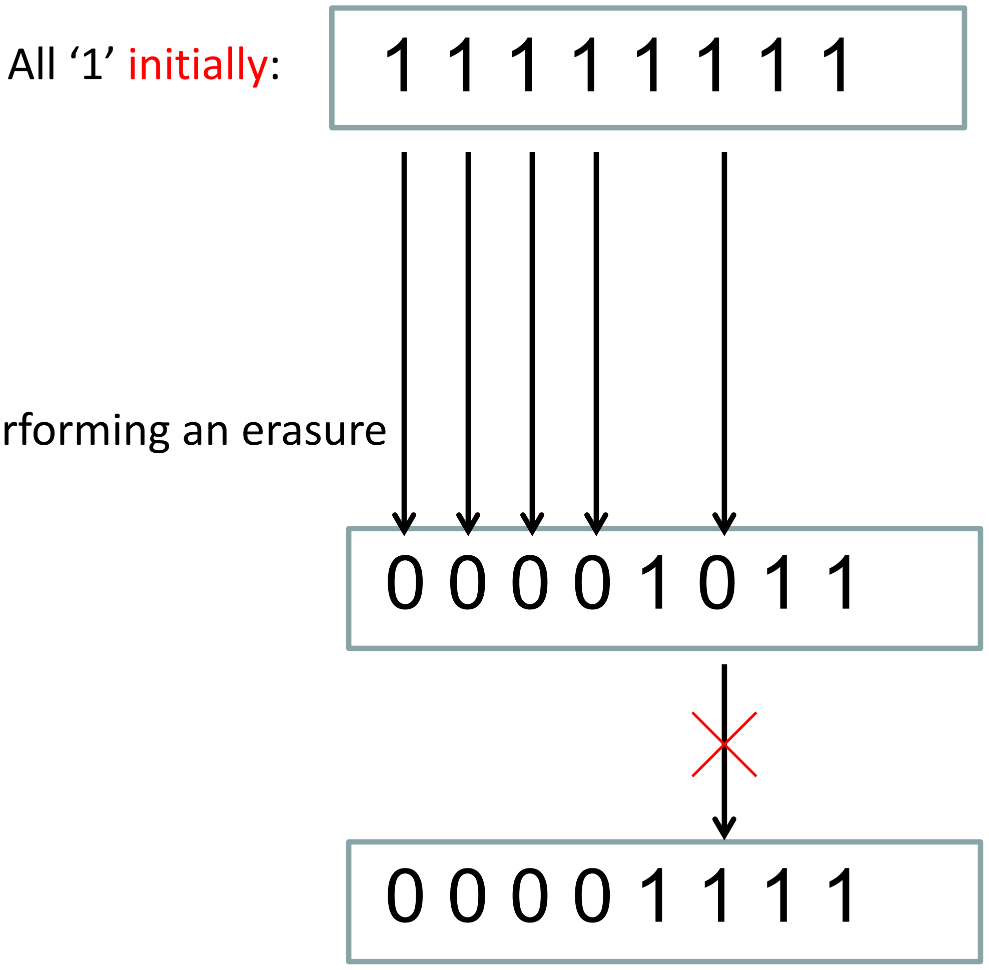


How to Program Flash Memory?

Write **0x0b**:
Rule:
1) 1 can be programed to 0
2) 0 cannot be programed to 1 except performing an erasure

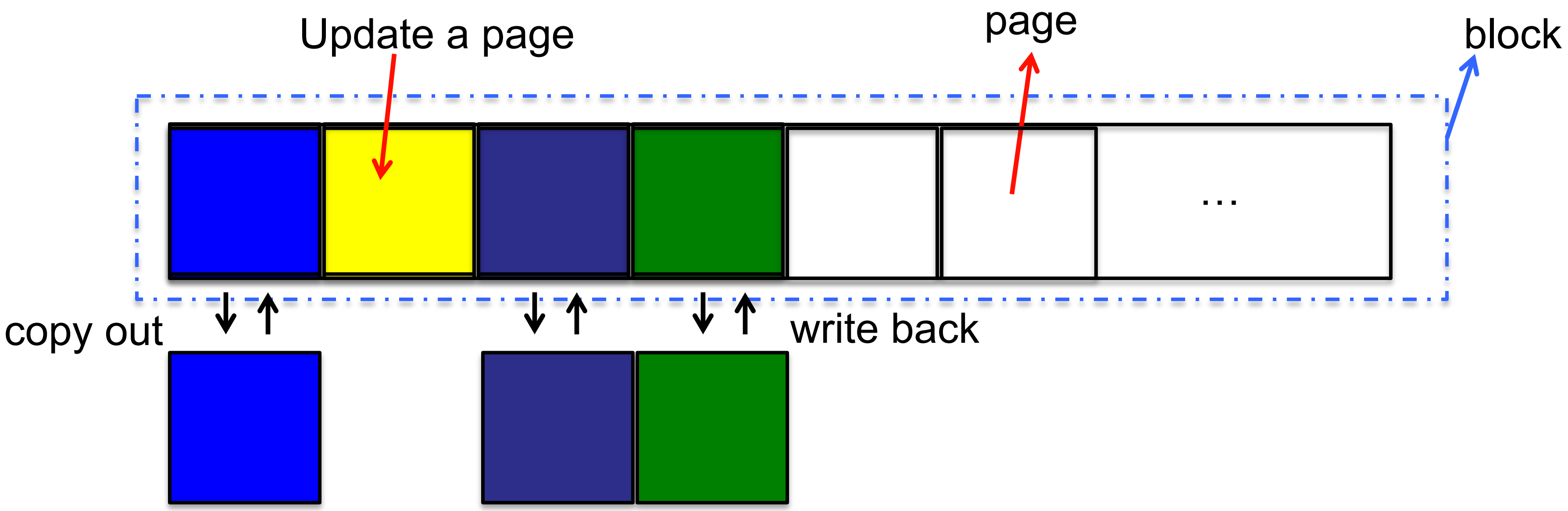
Modify 0x0b to **0x0f**?

Need to erase to all '1' first



Special Characteristics of Flash Memory

- Update unfriendly
  - Over-writing a page requires first erasing the entire block
  - Write is performed in pages (e.g., 4KB), but erase is performed in blocks (e.g., 128KB)



  - Over-write may cause significant write amplification
  - Usually prefer out-of-place update instead of in-place update

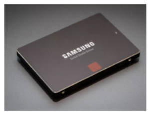
Special Characteristics of Flash Memory (cont.)

- Support a finite number of program-erase (P/E) cycles
  - Each flash block can only be programmed/erased for a limited number of times (e.g., 10K)
  - Data should be placed evenly across flash (wear leveling)

How to Manage Flash Memory?

- Flash-specific file systems, which can handle the special characteristics of NAND flash
  - YAFFS/YAFFS2, UBIFS, F2FS, JFFS/JFFS2

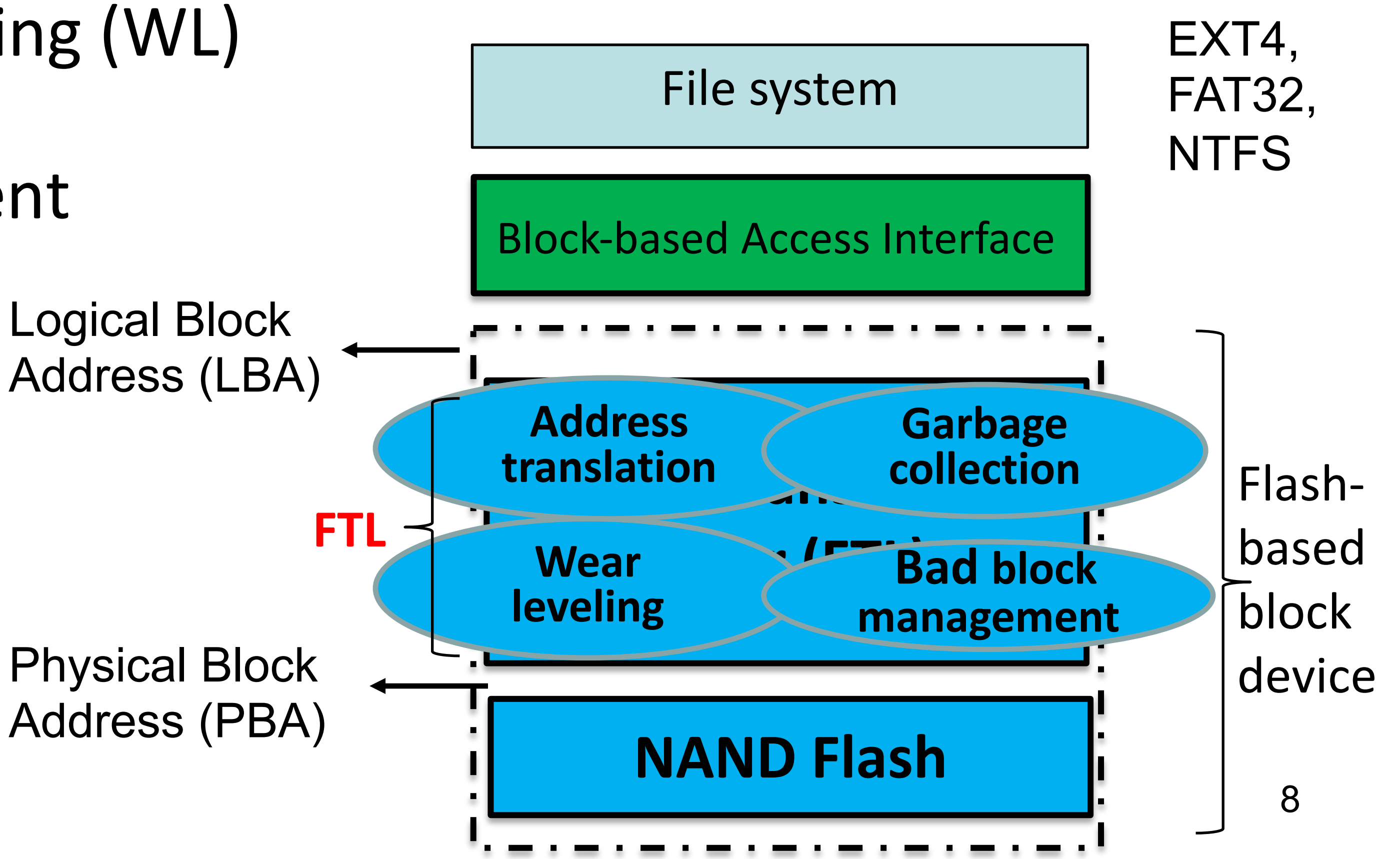
- Flash translation layer (FTL) – a flash firmware embedded into the flash storage device, which can handle the special characteristics of NAND flash and emulate the flash storage as a regular block device (most popular)
  - SSD
  - USB
  - SD/miniSD/MicroSD
  - MMC cards



Flash Translation Layer (FTL)

FTL usually provides the following functionality:

- ✓ Address translation
- ✓ Garbage collection (GC)
- ✓ Wear leveling (WL)
- ✓ Bad block management



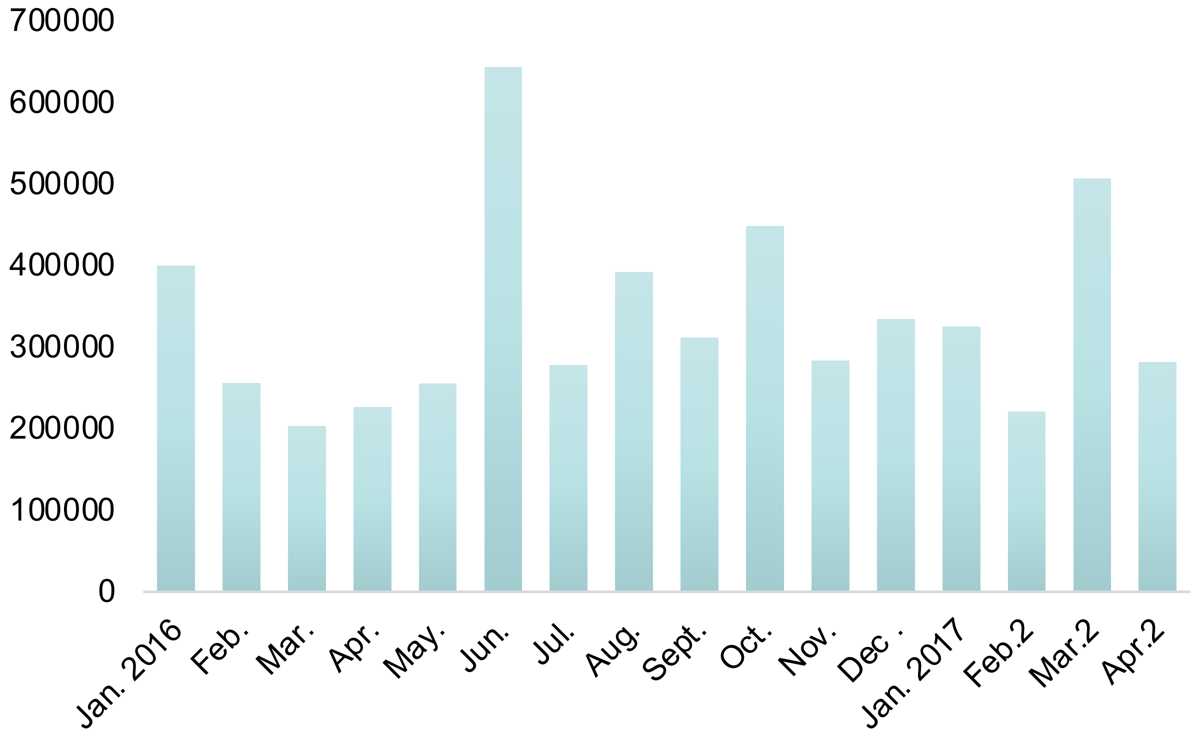
Outline

- Fast system recovery after bare-metal malware analysis
- Data recovery from ransomware without paying ransom
- Other Research

Outline

- Fast system recovery after bare-metal malware analysis
- Data recovery from ransomware without paying ransom
- Other Research

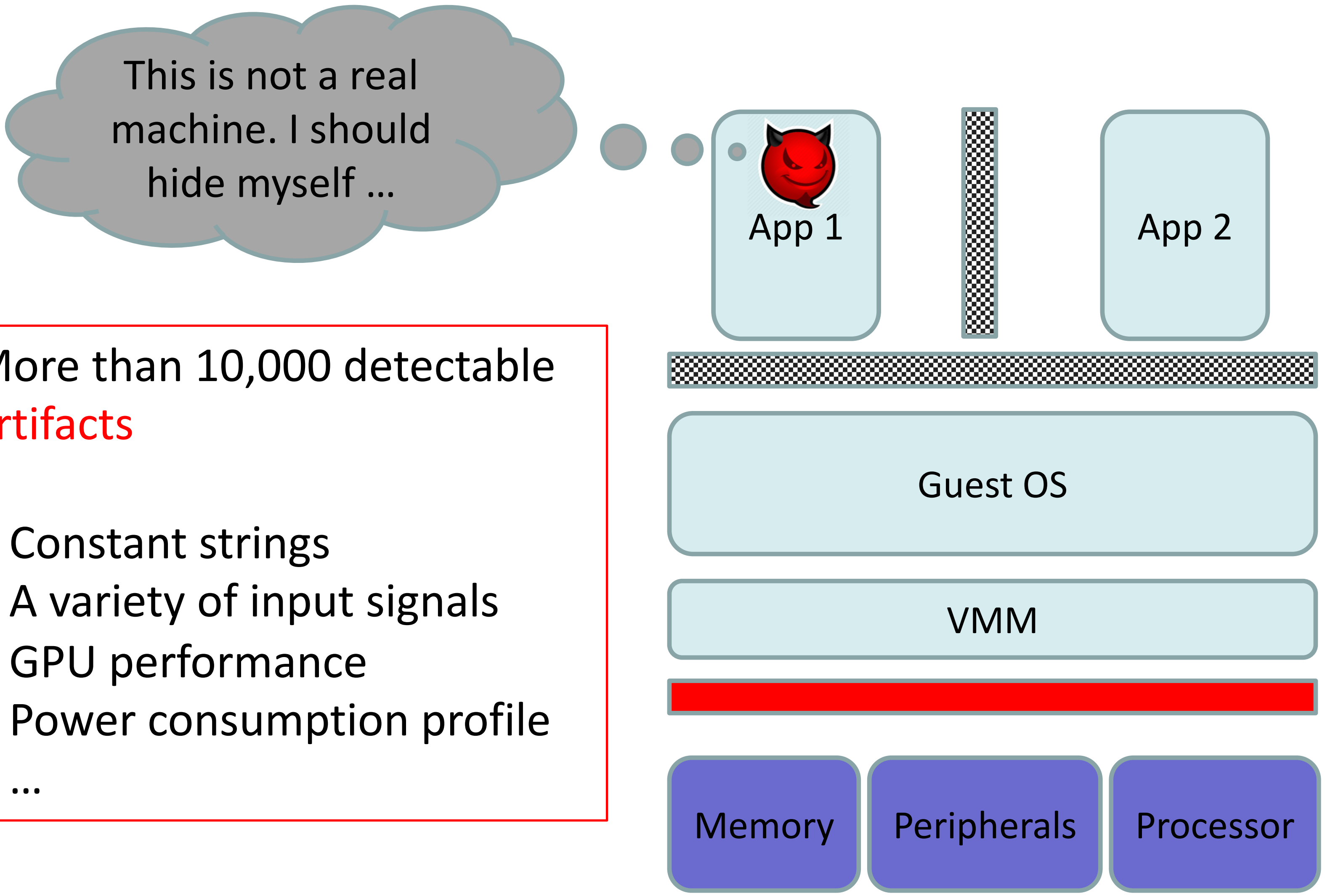
Android Malware Proliferates Recent Years



Data source: AV-TEST Security Report 2016-2017

Malware analysis is necessary for malware mitigation!

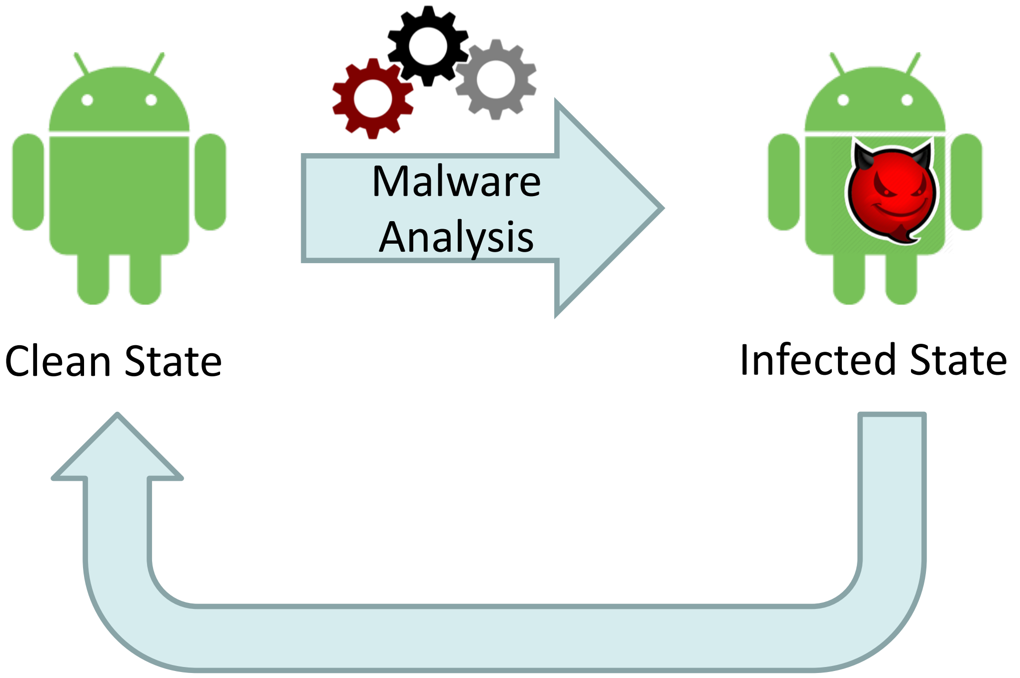
Sandboxed Malware Analysis in Mobile Devices is Not Good



More than 10,000 detectable artifacts

- Constant strings
- A variety of input signals
- GPU performance
- Power consumption profile
- …

Bare-metal Malware Analysis in Mobile Devices is Useful



how to efficiently restore?

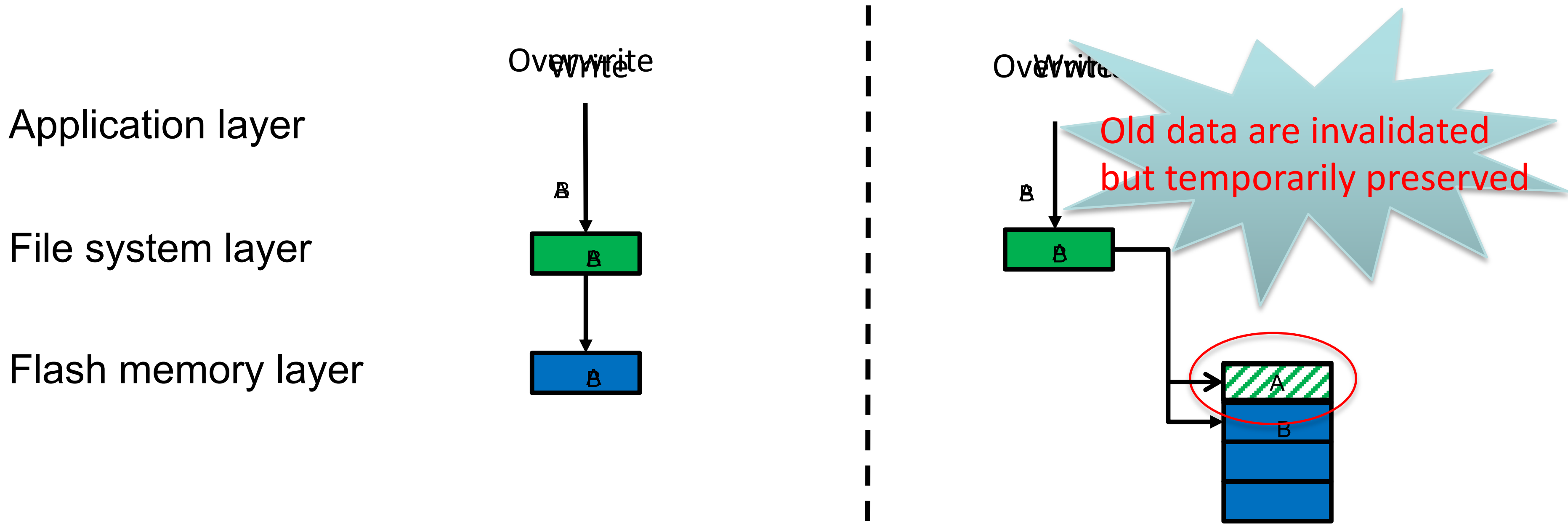
How to Efficiently Restore The Infected State to The Clean State [ACSAC ’17]?

- Data in memory
  - Before malware analysis, back up the memory data to a special memory region
  - The special memory region is isolated by ARM trustzone
    - Malware is not able to corrupted the data in the isolated memory region
  - After malware analysis, the data in the isolated memory region will be copied back to restore the memory
- Data in external flash storage (focus in this talk)
  - How to recover data stored in the external storage back to the clean state?

Le Guan, Shijie Jia, **Bo Chen**, et al. Supporting Transparent Snapshot for Bare-metal Malware Analysis on Mobile Devices. 2017 Annual Computer Security Applications Conference (ACSAC '17), 2017 (Acceptance rate: 19.7%) (Distinguished Paper Award)

A Review of An Interesting Feature of Flash Storage: Out-of-place Update

🟩 Logic Page 🟦 Physical Page ▨ Invalid Physical Page



(a) In-place update on mechanical disk drives

(b) Out-of-place update on flash memory

When Malware Corrupts Data in The Upper Layer

- During malware analysis, the malware will try to corrupt the data stored in the external storage by over-writing them using garbage information
  - Malware usually runs in the upper layer (e.g., application layer / file system layer)

- The data being over-written by the malware from the application layer will be invalidated but still preserved ("old data") in the flash memory
  - Thanks to the out-of-place update feature of flash memory

Can The Malware Directly Corrupt The "Old Data" Preserved in Flash Memory?

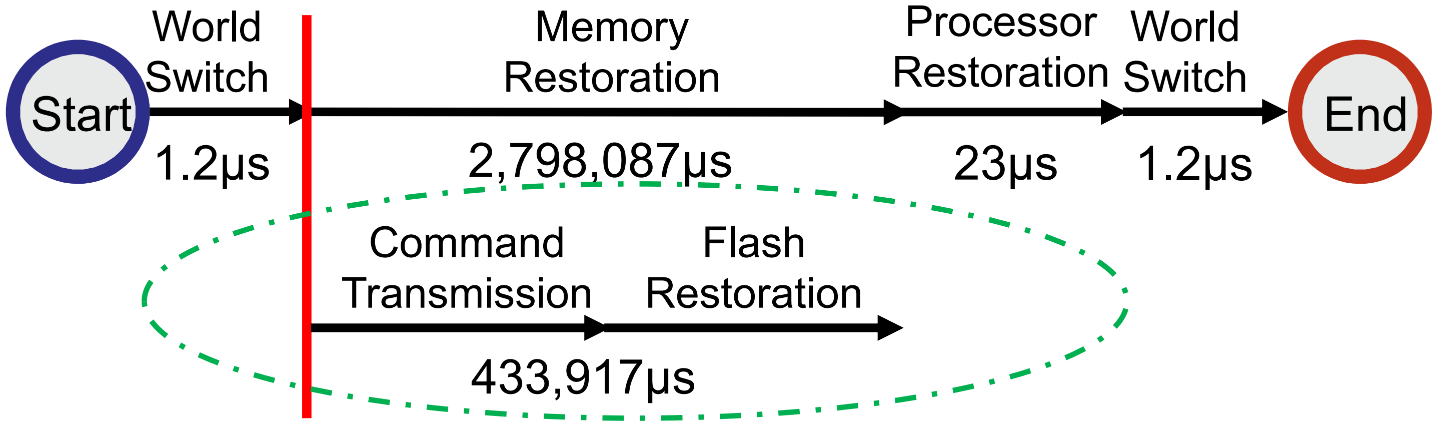
- The answer is No
  - The flash memory is usually completely isolated from the host computer system (e.g., SD cards, USB)
    - Independent hardware (processor, RAM)
    - Independent software (flash firmware)
    - Interface: SCSI, ATA, etc.
  - The malware cannot bypass the interface to corrupt the "old" data
    - Even if the malware can compromise the operating system

How to Prevent "Old Data" from Being Deleted by Garbage Collection?

- The flash memory usually implements garbage collection
  - Periodically reclaim space occupied by invalid "old data"
- The preserved "old data" will be eventually deleted by garbage collection
- The solution is to temporarily disable garbage collection during malware analysis
  - In other words, there is no data deletion in the flash memory during malware analysis
  - Right before malware analysis, we can perform a regular garbage collection
  - After malware analysis, the old data will be recovered, and the garbage collection will return to norm

Evaluation – Recovery Performance

The time required to restore an Android system with 512 MB physical memory, and 512 MB flash storage.



Outline

- Fast system recovery after bare-mental malware analysis
- Data recovery from ransomware without paying ransom
- Other research

Ransomware

- A piece of special malware that infects a computer and restricts access to the computer and/or its files
  - A ransom needs to be paid in order for the restriction to be removed

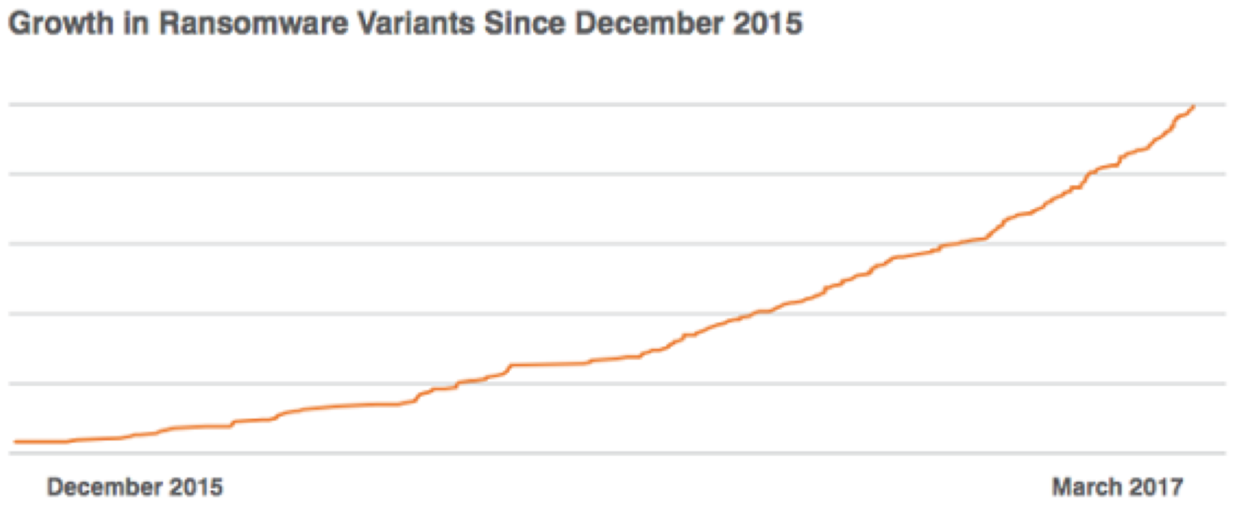
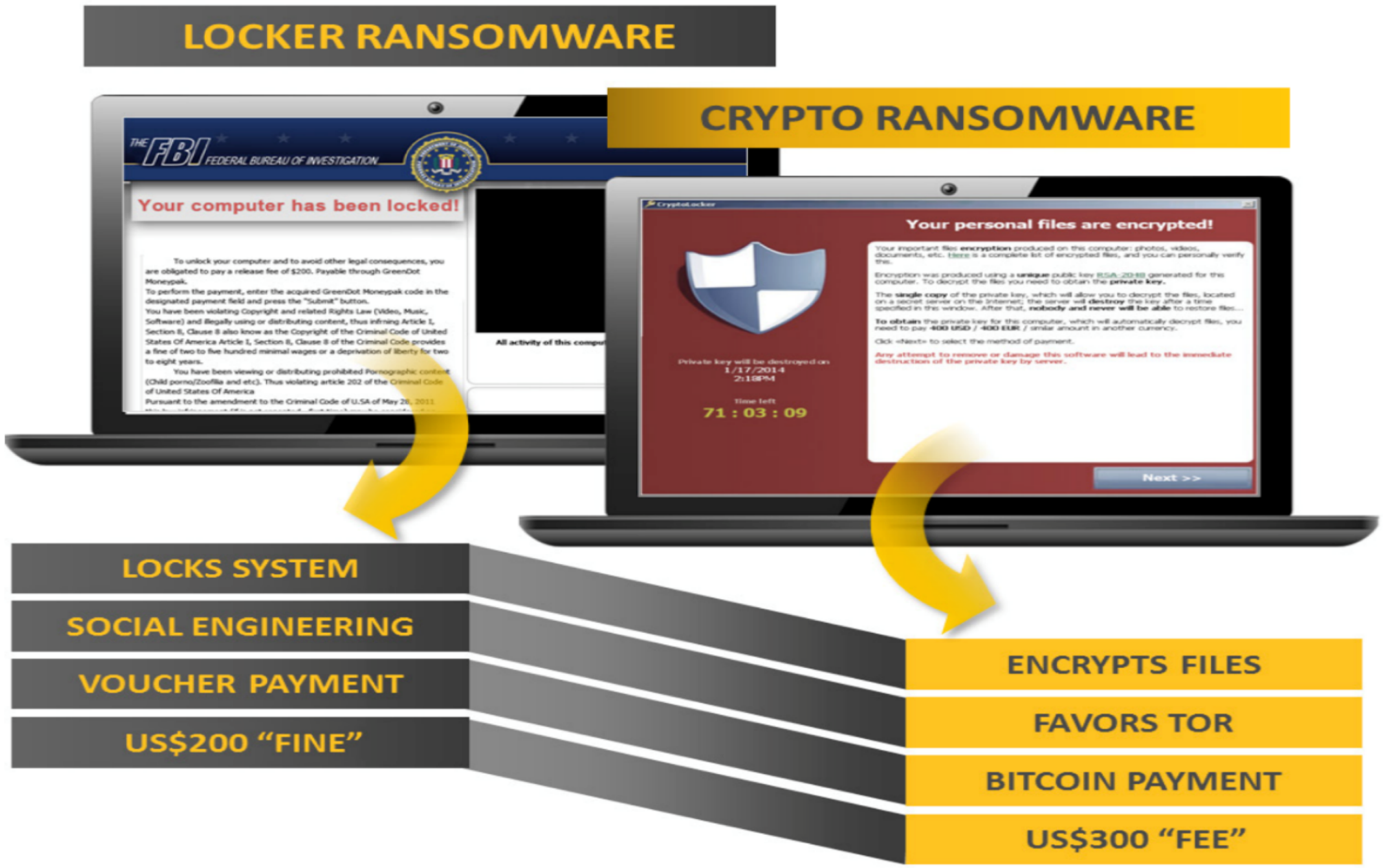


Figure 6: Indexed growth in total number of observed ransomware strains, December 2015-March 2017

Source: Proofpoint Q1 2017 Quarterly Threat Report

Main Types of Ransomware

- Locker ransomware
- Crypto-ransomware



Your computer has been locked!

To unlock your computer and to avoid other legal consequences, you are obligated to pay a release fee of $200. Payable through GreenDot Moneypak.
To perform the payment, enter the acquired GreenDot Moneypak code in the designated payment field and press the "Submit" button.
You have been violating Copyright and related Rights Law (Video, Music, Software) and illegally using or distributing content, thus infrning Article I, Section 8, Clause 8 also know as the Copyright of the Criminal Code of United States Of America Article I, Section 8, Clause 8 of the Criminal Code provides a fine of two to five hundred minimal wages or a deprivation of liberty for two to eight years.
You have been viewing or distributing prohibited Pornographic content (Child porno/Zoofilia and etc). Thus violating article 202 of the Criminal Code of United States Of America
Pursuant to the amendment to the Criminal Code of U.S.A of May 28, 2011

All activity of this compu

Your personal files are encrypted!

Your important files **encryption** produced on this computer: photos, videos, documents, etc. Here is a complete list of encrypted files, and you can personally verify this.

Encryption was produced using a **unique** public key RSA-2048 generated for this computer. To decrypt the files you need to obtain the **private key.**

The **single copy** of the private key, which will allow you to decrypt the files, located on a secret server on the Internet; the server will **destroy** the key after a time specified in this window. After that, **nobody and never will be able** to restore files...

To obtain the private key for this computer, which will automatically decrypt files, you need to pay **400 USD / 400 EUR** / similar amount in another currency.

Click «Next» to select the method of payment.

Any attempt to remove or damage this software will lead to the immediate destruction of the private key by server.

Private key will be destroyed on
1/17/2014
2:18PM

Time left
71 : 03 : 09

How to Combat Locker Ransomware?

- Observation: only the system is locked by the ransomware, but the data are stored intact

- Unplug the storage medium (e.g., SSD drives, microSD cards), plug the storage medium to a new computing device, and copy out the data

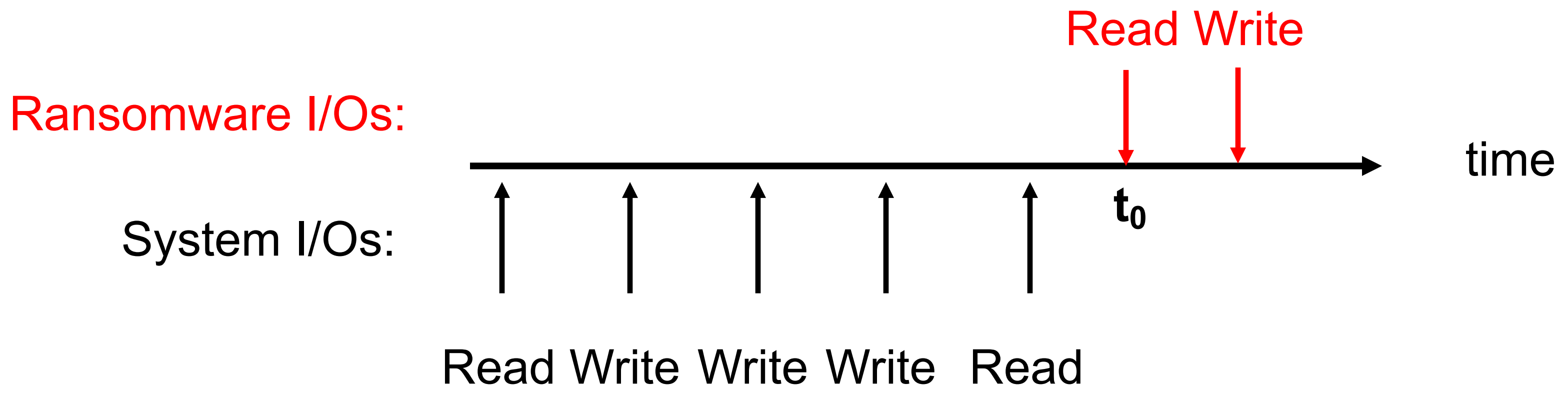
- Plug the storage device back to the device which has been locked, and re-install/initialize the system, then copy the data back

Crypto-ransomware Defense

- Crypto-ransomware behaviors:
  - Encrypt the victim data, and delete the original data
    - In systems, the delete operation is implemented by overwriting the data with garbage data
  - Or encrypt the victim data, and use the ciphertext to overwrite the original data
- Data recovery from crypto-ransomware attacks
  - Option 1: obtain the decryption key
    - Pay the ransom: money loss; cannot guarantee the key can work after paying the ransom
    - Extract the key locally: may work if the ransomware uses symmetric encryption, but no guarantee the key can be extracted
  - Option 2: data recovery from backups
    - More reliable

A Challenging Issue When Restoring Victim Data from Backups

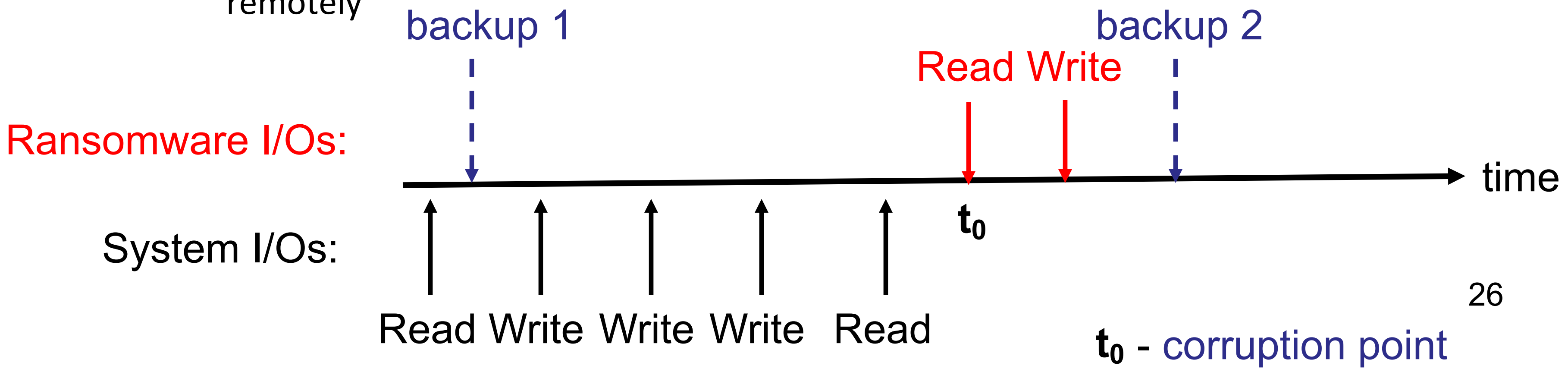
- After a computing device is hacked by ransomware, the victim data will be recovered by backups
- A challenging issue is how to *ensure data stored in the victim device is recoverable to the exact point right before the corruption* (i.e., corruption point)?



t_0 - corruption point

Remote Backups Cannot Ensure Recoverability of Data at The Corruption Point

- Data stored in a computing device may be periodically backed up to a remote server (e.g., a cloud server)
  - E.g., iCloud periodically backs up an iPhone
- The remote backups cannot ensure recoverability of data at the corruption point
  - Each backup operation usually happens periodically (e.g., daily, hourly) rather than continuously
    - No enough battery
    - Internet is not necessarily available any time
    - There is no guarantee that the data at the corruption point have been backed up remotely

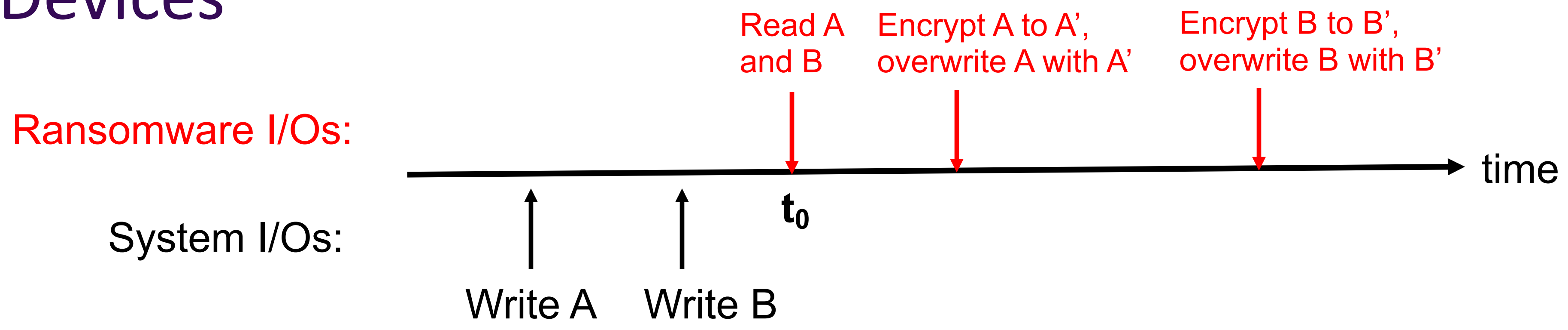


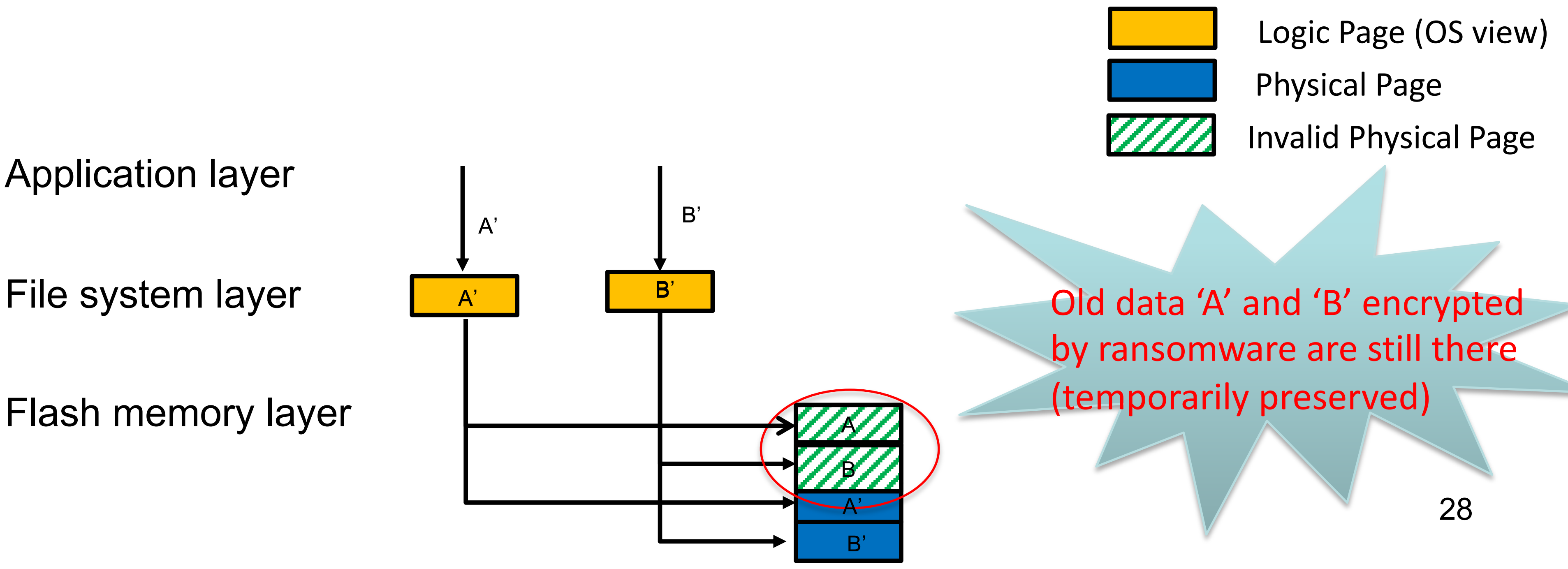
What about Doing Backups Locally at The Upper Layers?

- Data can be backed up locally after each single write
  - User-level backups: the user duplicates a file
  - System-level backups: the OS backs up the entire external storage (e.g., Apple time machine, copy-on-write)

- This could be problematic:
  - Creating backups after each single write incurs a large overhead
  - The ransomware may compromise the entire OS and all the local backups created at the upper layers may be corrupted and cannot used for data recovery

Towards Restoring The Corruption Point in Mobile Devices

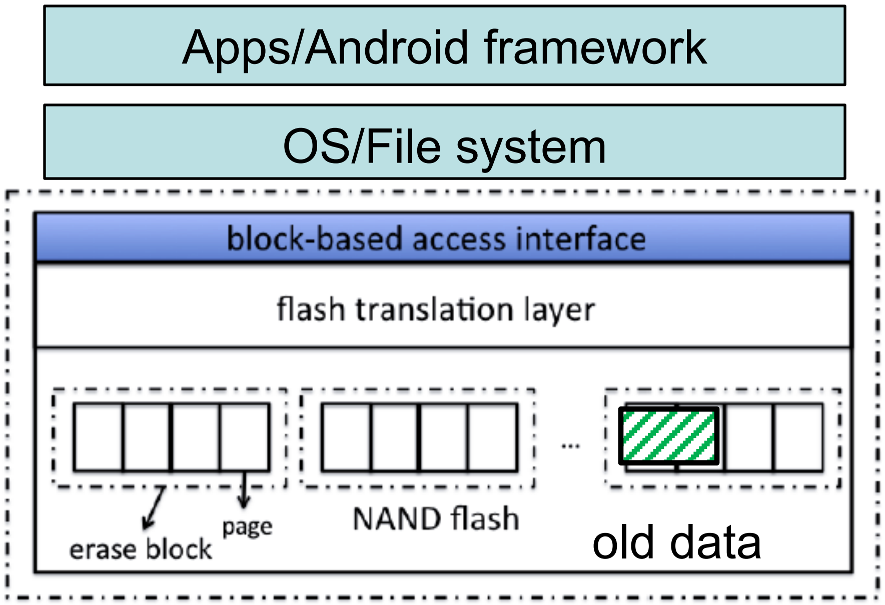

Read A and B
Encrypt A to A', overwrite A with A'
Encrypt B to B', overwrite B with B'
Ransomware I/Os:
time
t_0
System I/Os:
Write A
Write B



Old data 'A' and 'B' encrypted by ransomware are still there (temporarily preserved)

Taking Advantage of The Temporarily Preserved Old Data

- The temporarily preserved old data are the exact data encrypted by ransomware at the corruption point
  - They have been invalidated by the FTL and hence are invisible to the OS and apps from upper layers, and will not be "touched" by ransomware which can compromise the entire OS
  - They can be extracted to restore the data at the corruption point



A Few Additional Questions [CODASPY '19]

- How can we ensure that the temporarily preserved old data will not be reclaimed by garbage collection?
  - Garbage collection in the flash memory storage may reclaim space occupied by invalid data, leading to deletion of the temporarily preserved old data
  - Our solution: a phase garbage collection strategy
    - The garbage collection runs regularly if no ransomware is detected; the garbage collection will be temporarily disable if any ransomware is detected

Peiying Wang, Shijie Jia, **Bo Chen**, Luning Xia and Peng Liu. MimosaFTL: Adding Secure and Practical Ransomware Defense Strategy to Flash Translation Layer. The Ninth ACM Conference on Data and Application Security and Privacy (CODASPY '19), Dallas, TX, USA, March 2019 (Acceptance rate: 23.5%)

A Few Additional Questions (cont.)

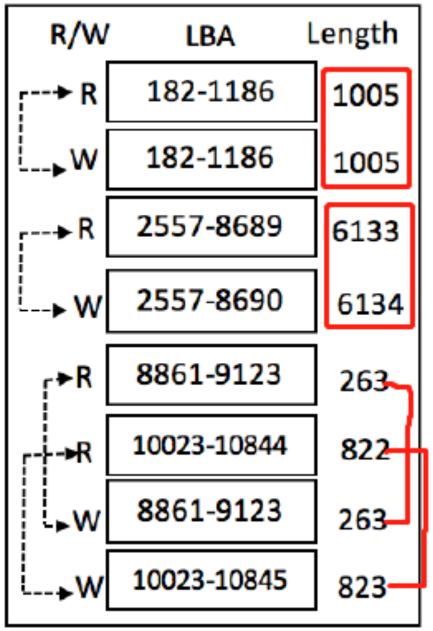
- We need a ransomware detector, but a ransomware detector running in the user/system level may not be able to function correctly
  - The user/system level ransomware detector may be disturbed by ransomware which can compromise the OS
- Our solution: incorporates the malware detector into the FTL layer
  - It remains transparent to the OS, and hence cannot be "touched" by the OS-level ransomware
- But is it possible to detect ransomware in the FTL layer?

A Ransomware Detector in FTL

- Studying access activities on flash memory causing by real-world ransomware
  - 518 ransomware samples (11 different ransomware families) collected from VirusTotal and Github
- Training
  - Use half of samples from each family for training purpose
  - For each sample running in the upper layers, collect 150 successive access on the flash memory (LBAs)
    - Each access contains access type, the starting destination address, and the size of I/O request

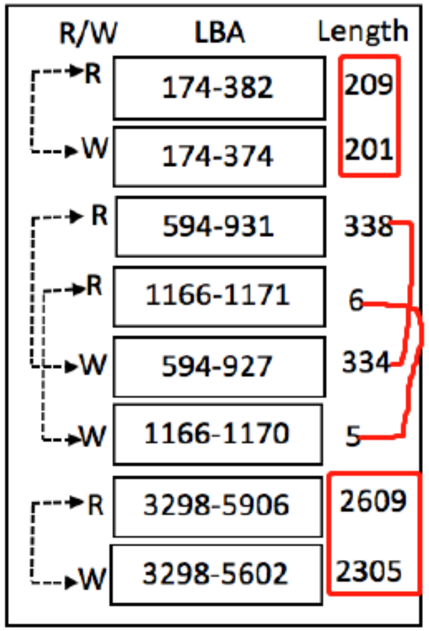
A Ransomware Detector in FTL (cont.)

- We observed 4 types of access patterns on the flash memory causing by real-world ransomware



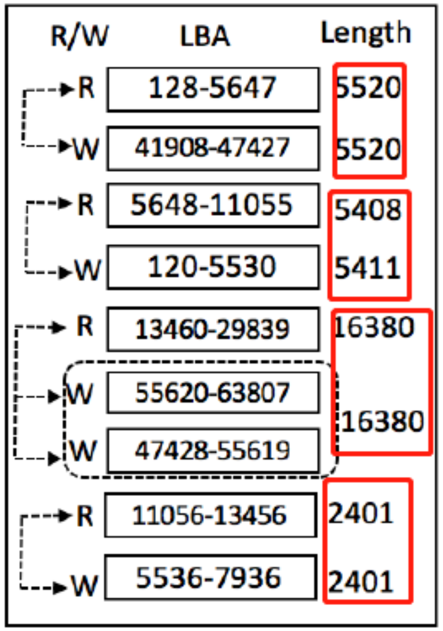
R/W	LBA	Length
R	182-1186	1005
W	182-1186	1005
R	2557-8689	6133
W	2557-8690	6134
R	8861-9123	263
R	10023-10844	822
W	8861-9123	263
W	10023-10845	823

(1) Type A



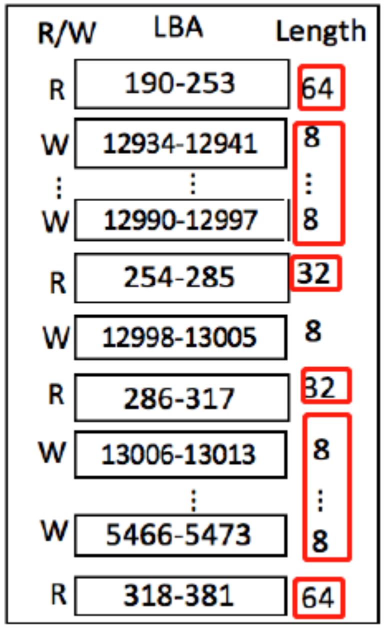
R/W	LBA	Length
R	174-382	209
W	174-374	201
R	594-931	338
R	1166-1171	6
W	594-927	334
W	1166-1170	5
R	3298-5906	2609
W	3298-5602	2305

(2) Type B



R/W	LBA	Length
R	128-5647	5520
W	41908-47427	5520
R	5648-11055	5408
W	120-5530	5411
R	13460-29839	16380
W	55620-63807	
W	47428-55619	16380
R	11056-13456	2401
W	5536-7936	2401

(3) Type C



R/W	LBA	Length
R	190-253	64
W	12934-12941	8
⋮	⋮	⋮
W	12990-12997	8
R	254-285	32
W	12998-13005	8
R	286-317	32
W	13006-13013	8
	⋮	⋮
W	5466-5473	8
R	318-381	64

(4) Type D

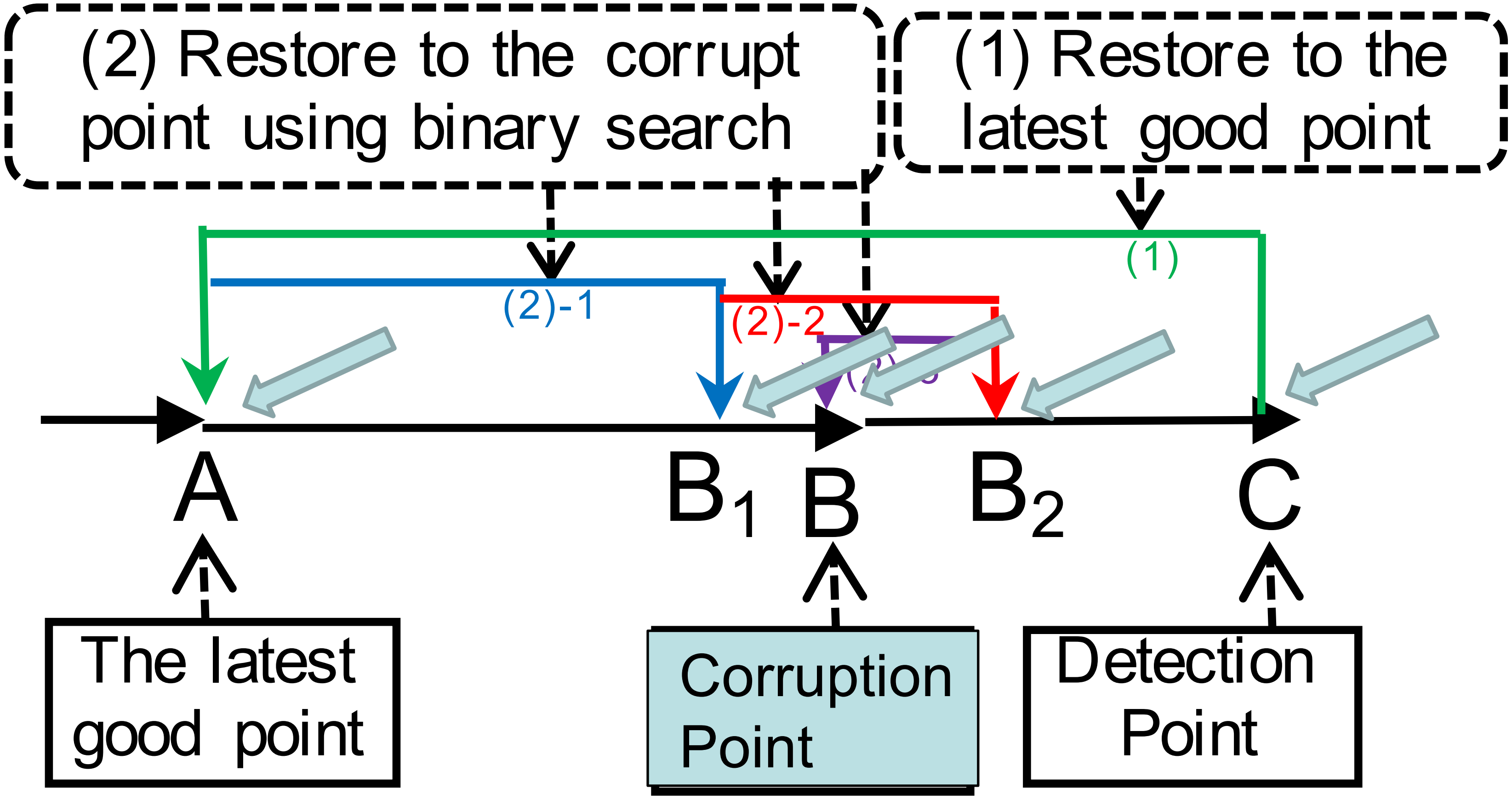
LBA – Logical Block Address

- Relying on the aforementioned patterns, we are able to detect ransomware when it is running

A Few Questions Needs to Be Addressed (cont.)

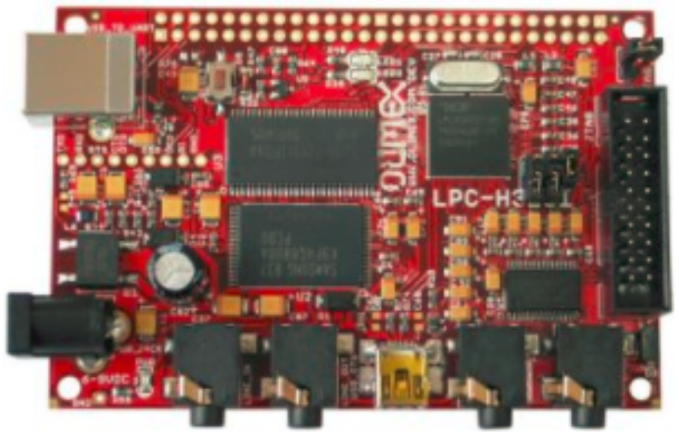
- How can we allow the user to have access to the preserved data, and how to efficiently locate the checkpoint before the malware starts to corrupt data?
  - There is a mapping between the block address and the flash memory address. If the mapping can be restored, the user can have access to the preserved old data again
  - Keep the mappings for each checkpoint over time
  - Locate the checkpoint using a binary search with O(logn) user involvement

How to Efficiently Locate The Corruption Point?

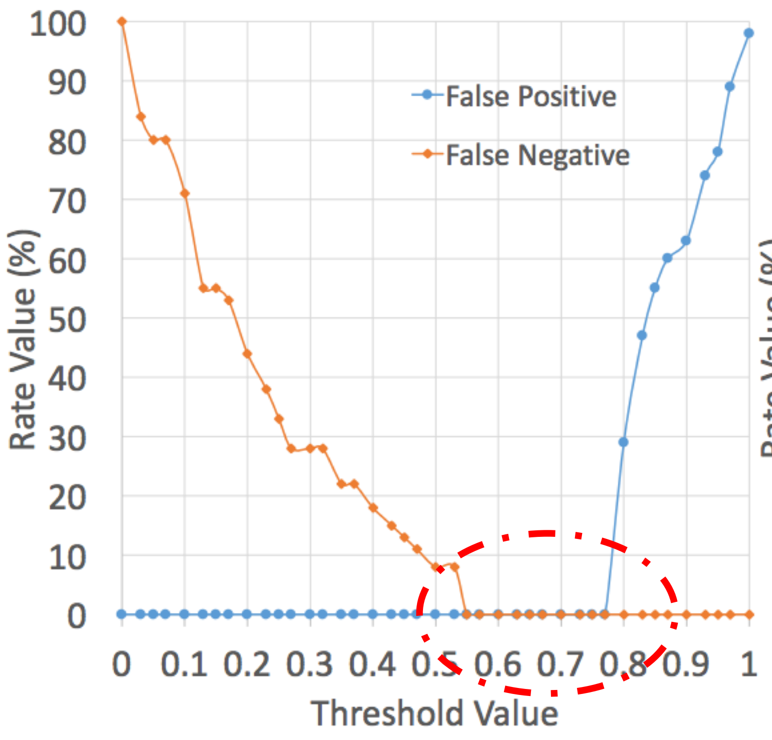

(2) Restore to the corrupt point using binary search
(1) Restore to the latest good point
(1)
(2)-1
(2)-2
(2)-3
A
B_1
B
B_2
C
The latest good point
Corruption Point
Detection Point

Implementation

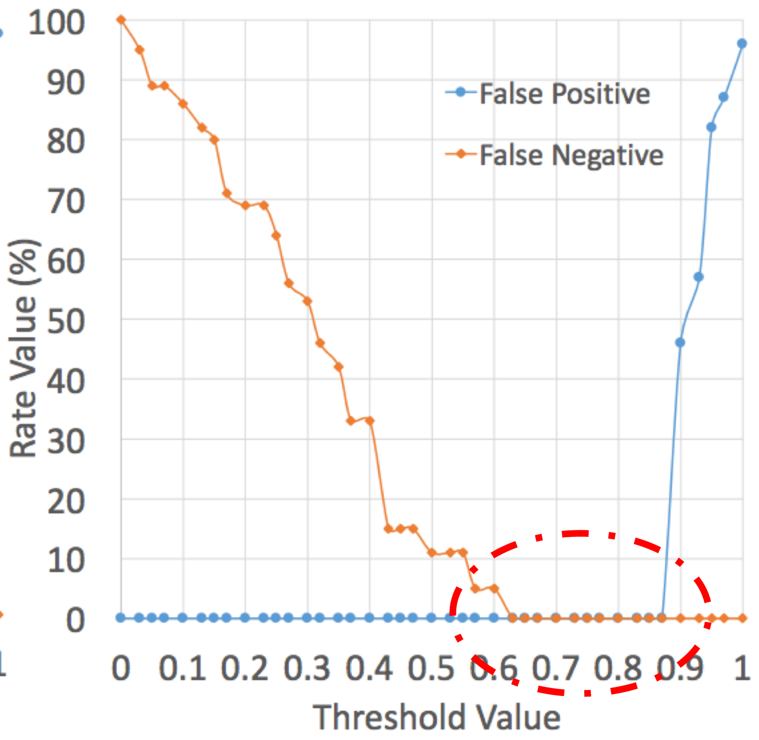
- Implemented our design MimosaFTL using an open-source NAND flash controller framework OpenNFM
- Ported the prototype to a development board LPC-H3131 (180MHz ARM, 512MB NAND flash, USB 2.0)



Evaluation – Detection Accuracy

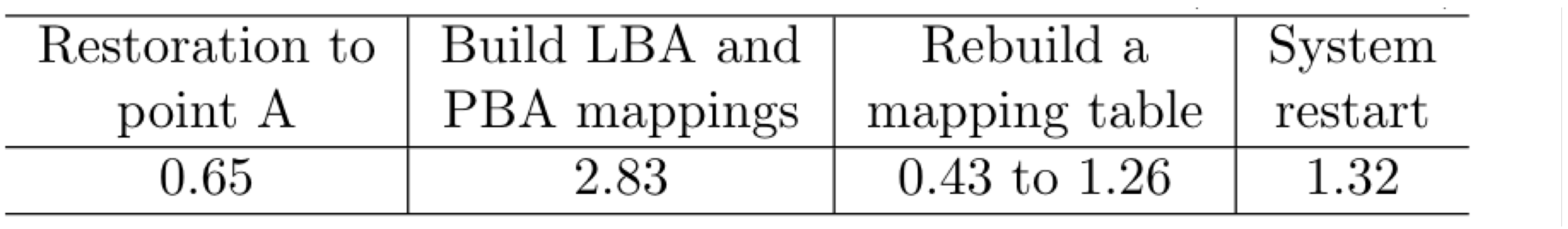


(a) Threshold_1 for type A/B/C ransomware



(b) Threshold_2 for type D ransomware

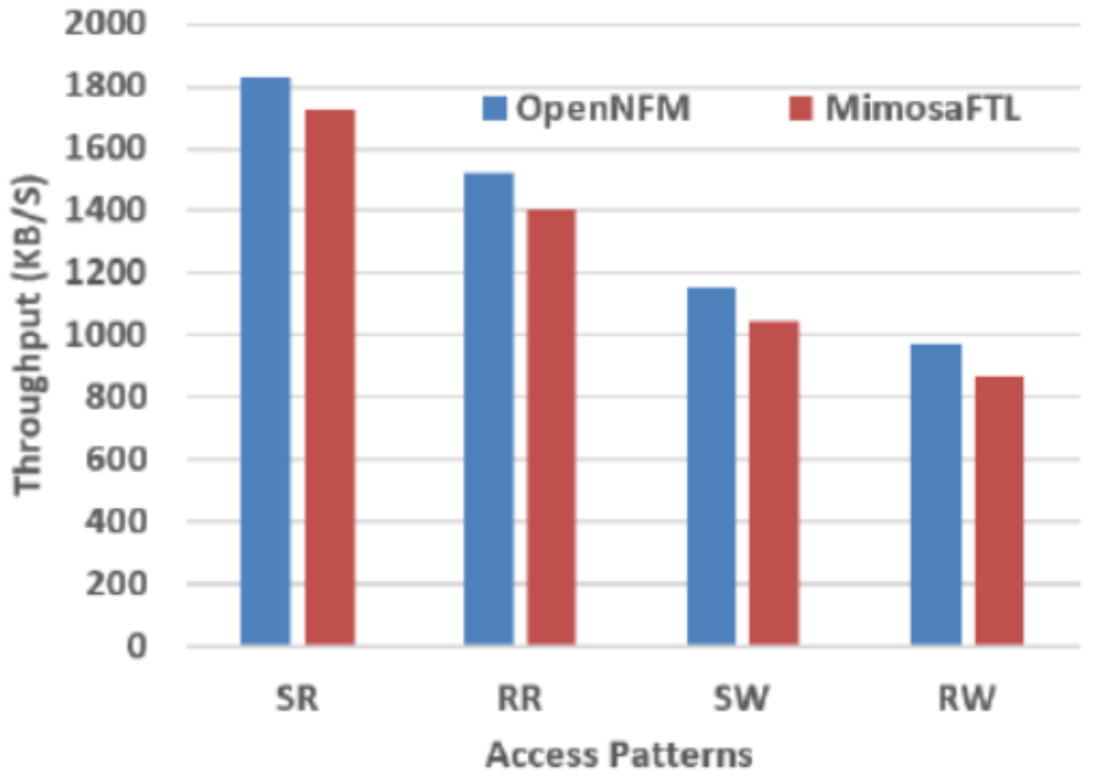
Evaluation – Recovery Efficiency



Restoration to point A	Build LBA and PBA mappings	Rebuild a mapping table	System restart
0.65	2.83	0.43 to 1.26	1.32

Each separate time (in seconds) spent on recovering all the 512MB data in LPC-H3131

Evaluation - Impact of Our Design (MimosaFTL) on Flash-based Block Devices



SR - sequential read,
RR - random read,
SW - sequential write,
RW - random write.

Outline

- Fast system recovery after bare-mental malware analysis
- Data recovery from ransomware without paying ransom
- Other research

Other Research on Protecting Data Stored in Flash Storage Media

- How to ensure data deleted by a mobile device owner are really removed?
  - Secure deletion: [ASIACCS ’19], [Cybersecurity ’18], [ACSAC ’16], [ASIACCS ’16]
  - Ensure confidentiality of data having been removed from flash storage

- How to ensure confidentiality of sensitive data even when the mobile device owner is captured and coerced to disclose the key?
  - Plausibly deniable storage systems: [Computers&Security ’18], [DSN ’18], [CCS ’17], [ACSAC ’15], [ISC ’14]
  - Ensure confidentiality of data still present in flash storage

Acknowledgments

- My research is currently supported by multiple grants from US National Science Foundation (NSF) and National Security Agency (NSA)
  - [1938130-CNS] **Bo Chen**, PI - NSF, *$199,975*, “EAGER: Enabling Secure Data Recovery for Mobile Devices against Malicious Attacks”, October 2019 - September 2021
  - [1928349-CNS] **Bo Chen**, PI - NSF, *$249,286*, “SaTC: CORE: Small: Collaborative: Hardware-Assisted Plausibly Deniable System for Mobile Devices”, October 2019 - September 2022
  - Yu Cai, PI; Tim Wagner, co-PI; Guy Hembroff, co-PI; **Bo Chen**, co-PI - NSA&NSF, *$77,895*, ``Innovative GenCyber Learning Experience for K-12 Teachers Through Storytelling + Teaching + Gaming + Doing'', May 2019 - May 2020
  - Etc

Thank You!

bchen@mtu.edu
https://cs.mtu.edu/~bchen
https://snp.cs.mtu.edu