Michigan Technological University

SnP Lab

CS5740 Spring 2020: Special Topic on Data Security (1)

Enabling Data Recovery from Malicious Attacks

Bo Chen

Department of Computer Science

Michigan Technological University

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

https://snp.cs.mtu.edu

bchen@mtu.edu

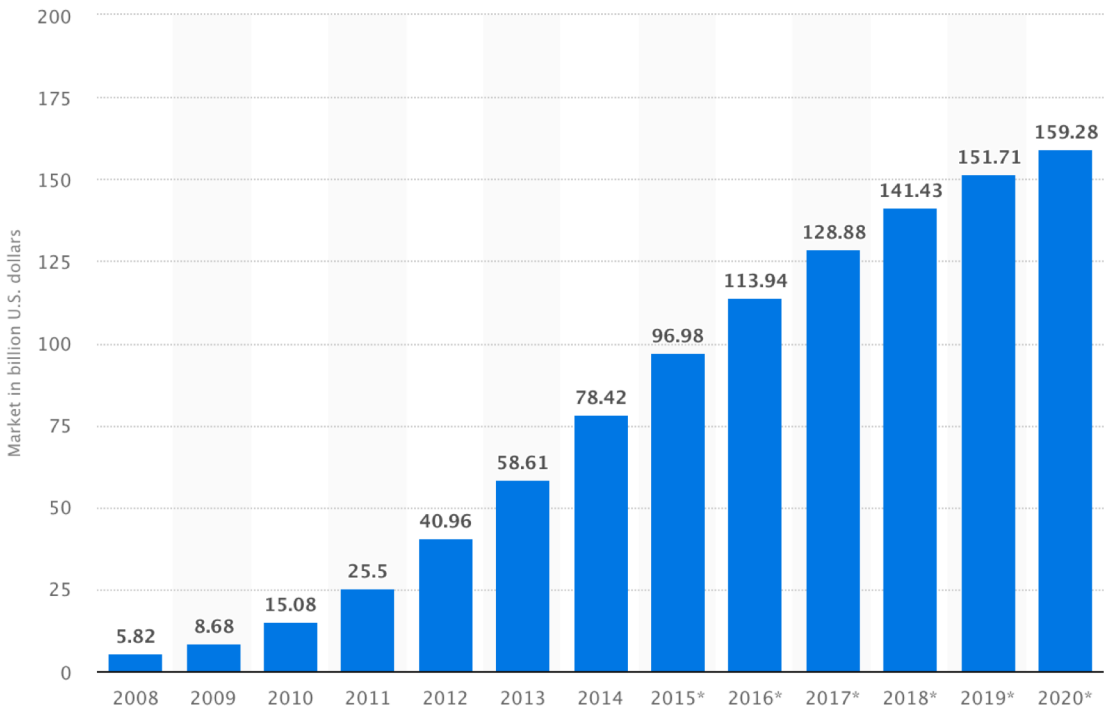
Outline

- Data Recovery in Public Clouds
- Data recovery in Mobile Devices

Outline

- Data Recovery in Public Clouds
- Data recovery in Mobile Devices

Clouds are Everywhere Today



Public cloud computing market worldwide 2008-2020

AWS

Microsoft Azure

Google Cloud

rackspace

KAMATERA

Major cloud service providers

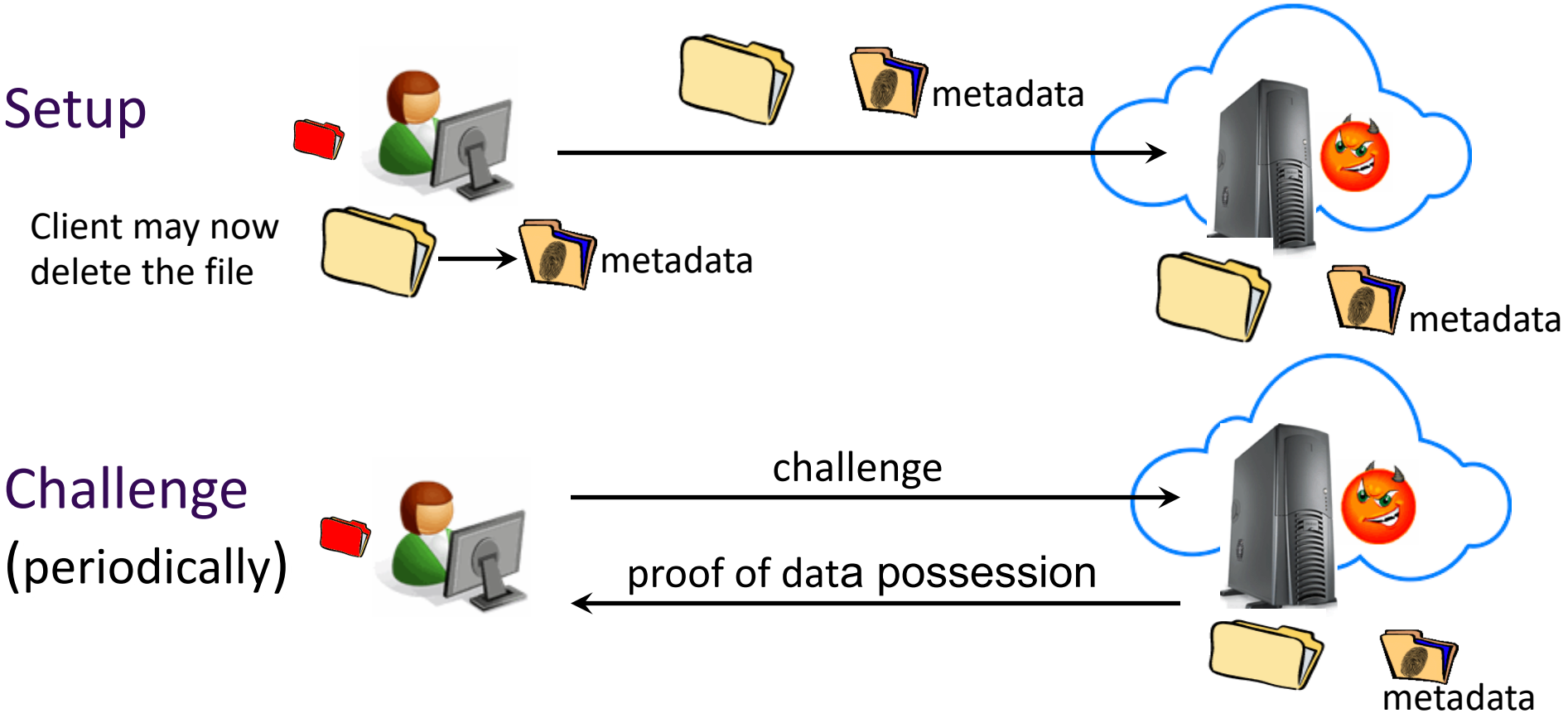
Cloud Providers Are Not Trusted

- Data owners may be reluctant to outsource their data to public Cloud Storage Providers (CSP)
  - Amazon S3, Microsoft Azure storage

- The CSP is not necessarily trusted, and may try to hide data loss incidents caused by:
  - Insider or outsider attacks
  - Hardware failures, management errors
  - Unexpected accidental events

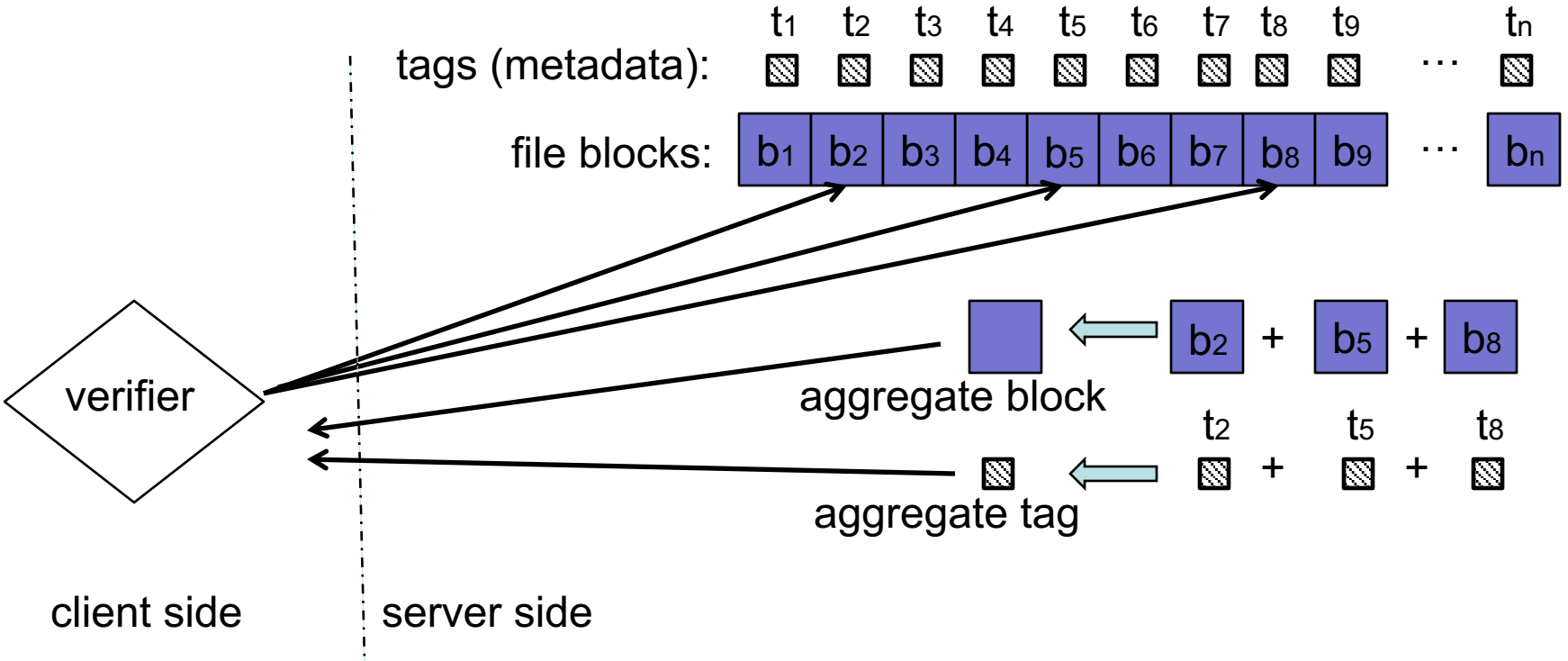
Remote Data integrity Checking (RDC)

- Remote Data integrity Checking (RDC) allows the data owner to check the integrity of data stored at an untrusted cloud provider
  - RDC [Ateniese et al., CCS '07; Juels et al., CCS '07; Shacham et al., ASIACRYPT '08]



How to Efficiently Verify Data Integrity

- Adopt spot checking technique for efficiency: the verifier (client) *randomly* samples a certain number of blocks for checking (*rather than check the whole outsourced data*)
  - It shows that if the adversary corrupts 1% of the data, by randomly sampling 460 blocks, the verifier can detect the corruption with 99% probability [AB+07, AB+11]



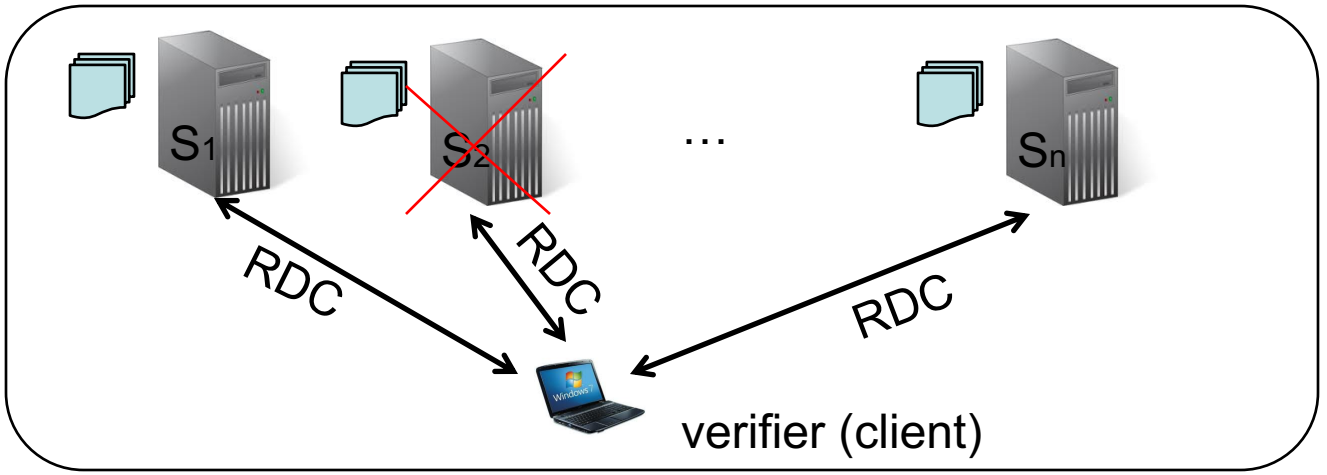
Small Corruption

- What if the adversary only corrupts a small portion of the outsourced data?
- Incorporate error correcting code (e.g., erasure coding) to restore small corruption
- Data outsourced to the cloud may be updated, but error correcting code is usually update unfriendly
- Accommodate both data updates and error correcting code in a secure manner [SPCC ’12]

Bo Chen and Reza Curtmola. Robust Dynamic Provable Data Possession. The Third International Workshop on Security and Privacy in Cloud Computing (SPCC '12), Macau, China, June 2012

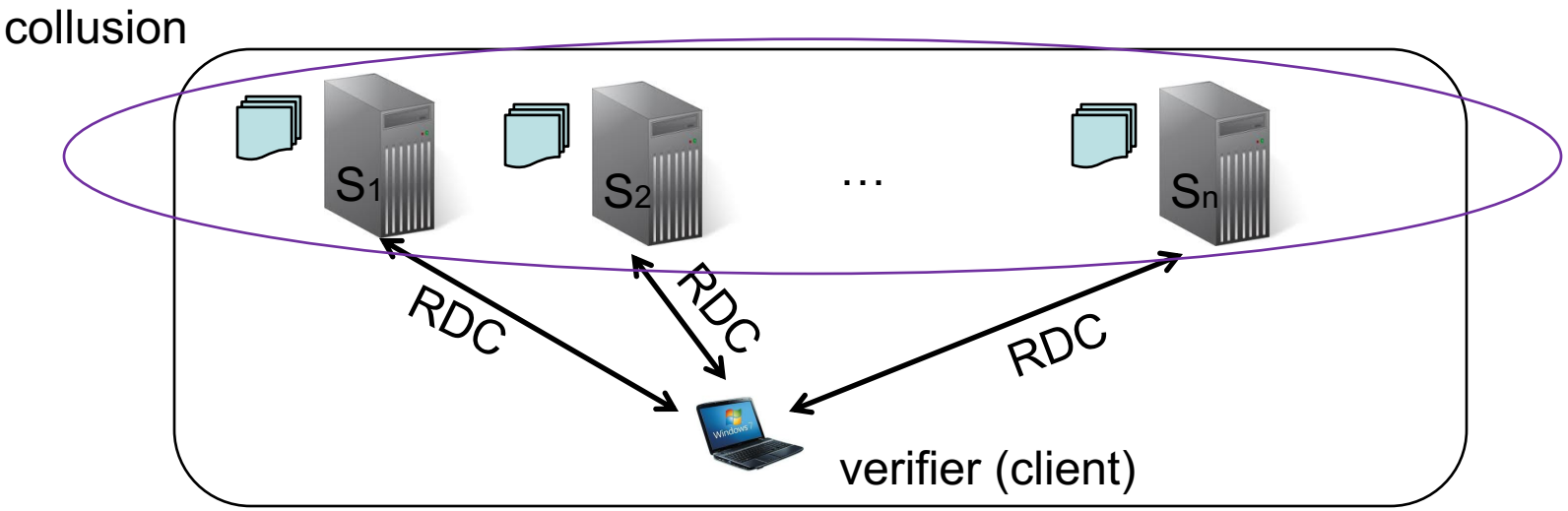
What If A Large Data Corruption Are Found?

- Ensure long-term data reliability
- Data should be stored redundantly at multiple servers/data centers
  - Replications
  - Erasure coding
  - Network coding



What If Cloud Servers Collude?

- The untrusted cloud servers may collude and only store one copy
- Ensure each untrusted server will honestly store the data [CCSW '10]



Bo Chen, Reza Curtmola, Giuseppe Ateniese, and Randal Burns. Remote Data Checking for Network Coding-based Distributed Storage Systems. The Second ACM Cloud Computing Security Workshop (CCSW '10),Chicago, IL, USA, October 2010

Some Other Interesting Problems

- Enforcing self-repairing
- Proofs of multiple locations
- Proofs of multiple drives
- Proofs of version control

Bo Chen, Anil Kumar Ammula, and Reza Curtmola. Towards Server-side Repair for Erasure Coding-based Distributed Storage Systems. The Fifth ACM Conference on Data and Application Security and Privacy (CODASPY '15), San Antonio, TX, USA, March 2015 (Acceptance rate: 29.7%)

Bo Chen and Reza Curtmola. Towards Self-Repairing Replication-Based Storage Systems Using Untrusted Clouds. The Third ACM Conference on Data and Application Security and Privacy (CODASPY '13), San Antonio, TX, USA, Feb. 2013 (Acceptance rate: 22.4%)

Bo Chen and Reza Curtmola. Auditable Version Control Systems. The 21th Annual Network and Distributed System Security Symposium (NDSS '14), San Diego, CA, USA, Feb. 2014 (Acceptance rate: 18.6%)

Outline

- Data Recovery in Public Clouds
- Data recovery in Mobile Devices

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

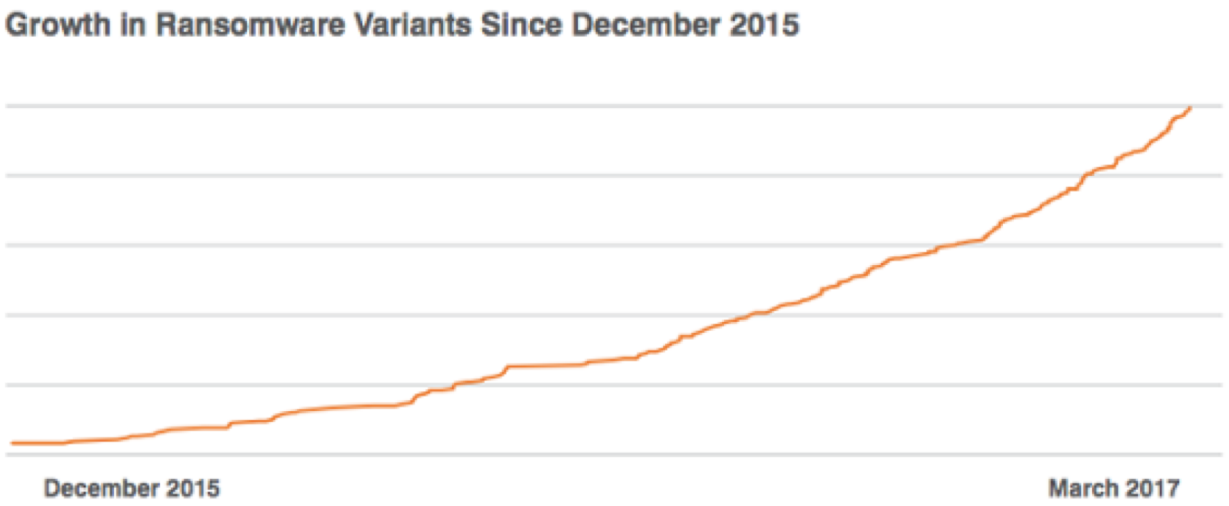


Figure 6: Indexed growth in total number of observed ransomware strains, December 2015-March 2017

Source: Proofpoint Q1 2017 Quarterly Threat Report

Main Types of Ransomware

- Locker ransomware
- Crypto-ransomware

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

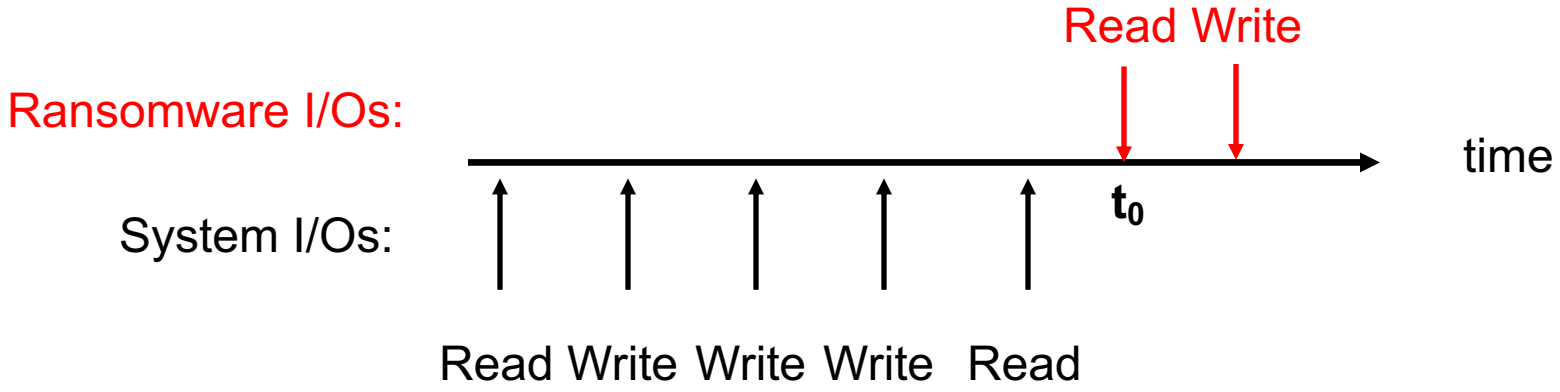
Crypto-ransomware Defense

- 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

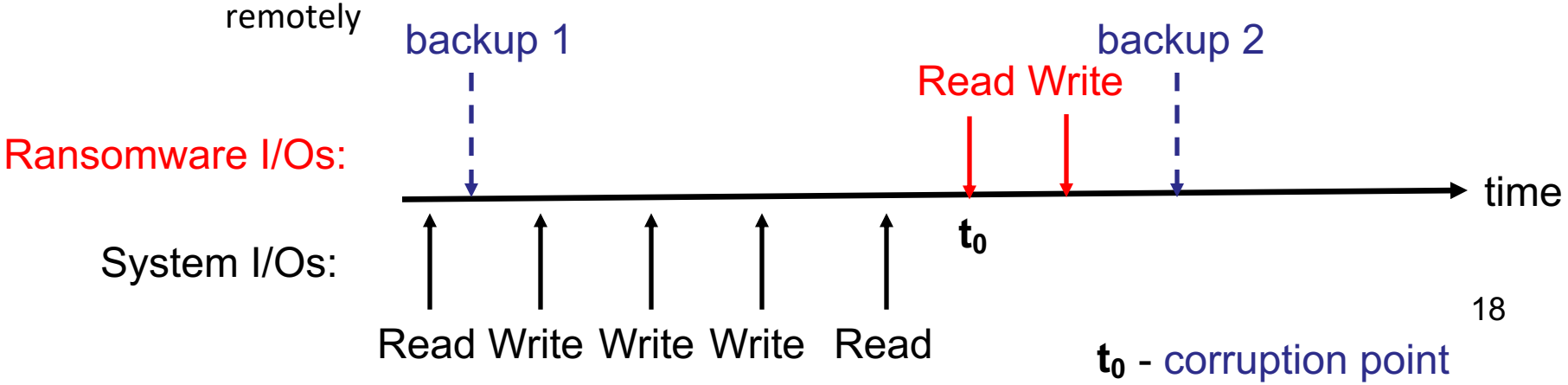
- A challenging issue is how to *ensure data stored in the victim device is recoverable to the exact point right before the corruption* (i.e., corruption point)?



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)

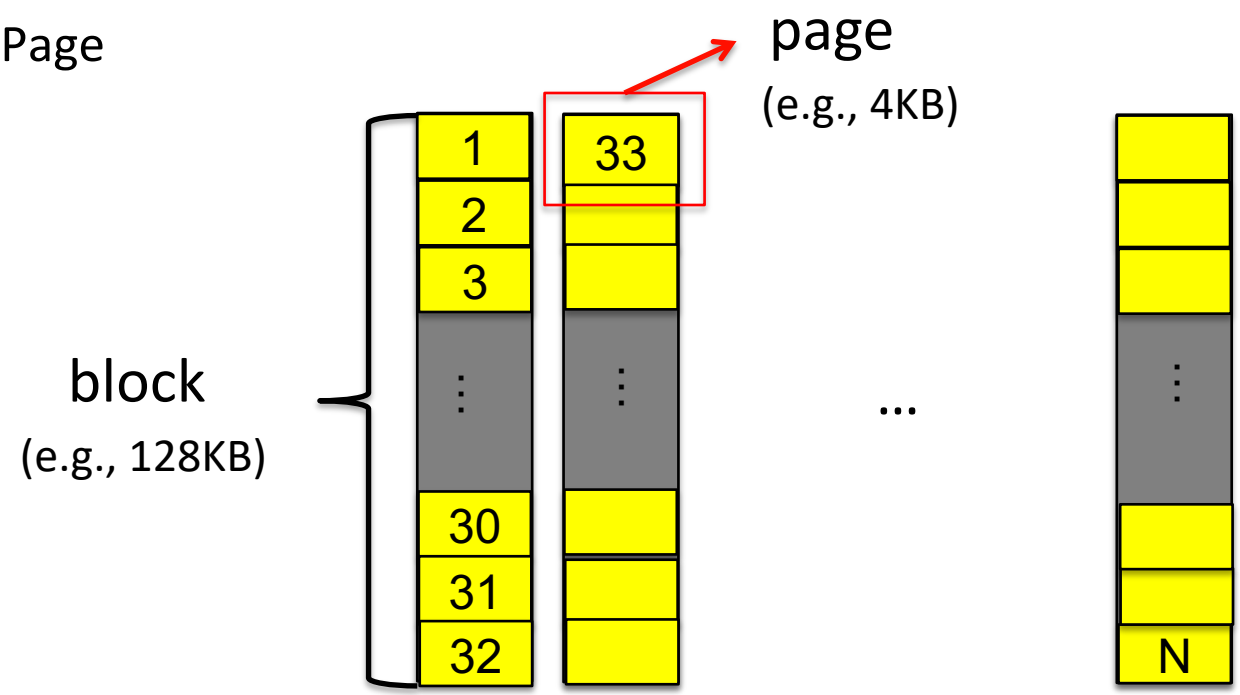
- This could be problematic:
  - Creating backups after each single write incurs a large overhead
  - The ransomware may compromise the entire OS and all the local backups created at the upper layers may be corrupted and cannot used for data recovery

Background on Flash Memory

NAND Flash Memory

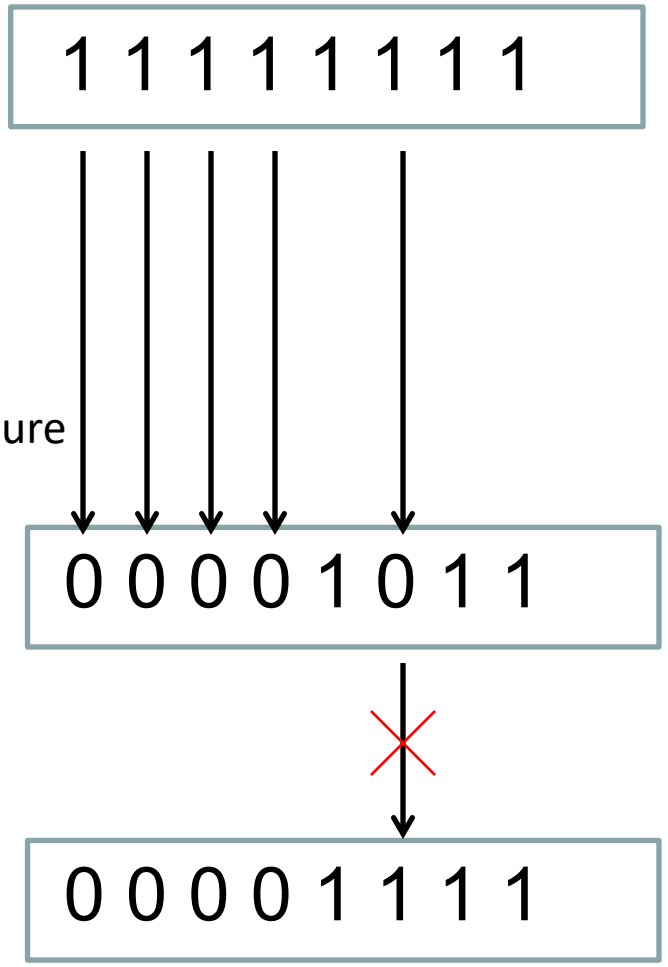
- Flash memory
  - NAND flash (broadly used for mass-storage of mobile devices)
  - NOR flash (used for storing program code that rarely needs to be updated, e.g., a computer's BOIS)

- NAND flash organization
  - Block
  - Page



How to Program Flash Memory?

All ‘1’ initially:



Write **0x0b**:

Rule:

1) 1 can be programed to 0

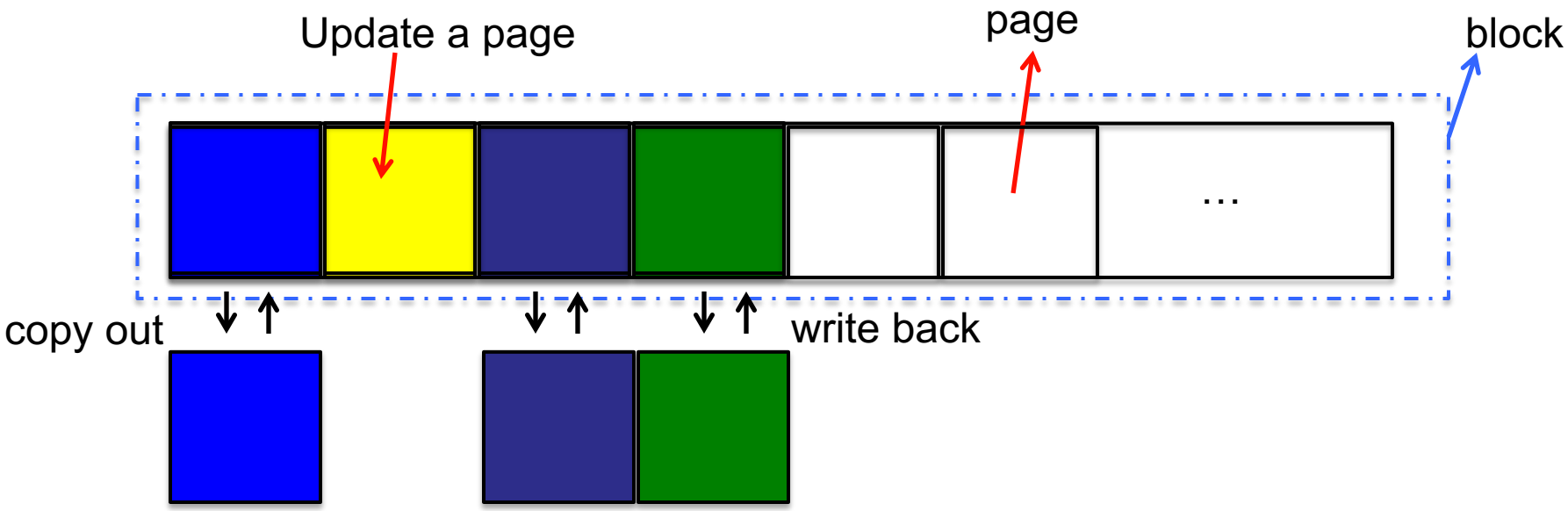
2) 0 cannot be programed to 1 except performing an erasure

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?

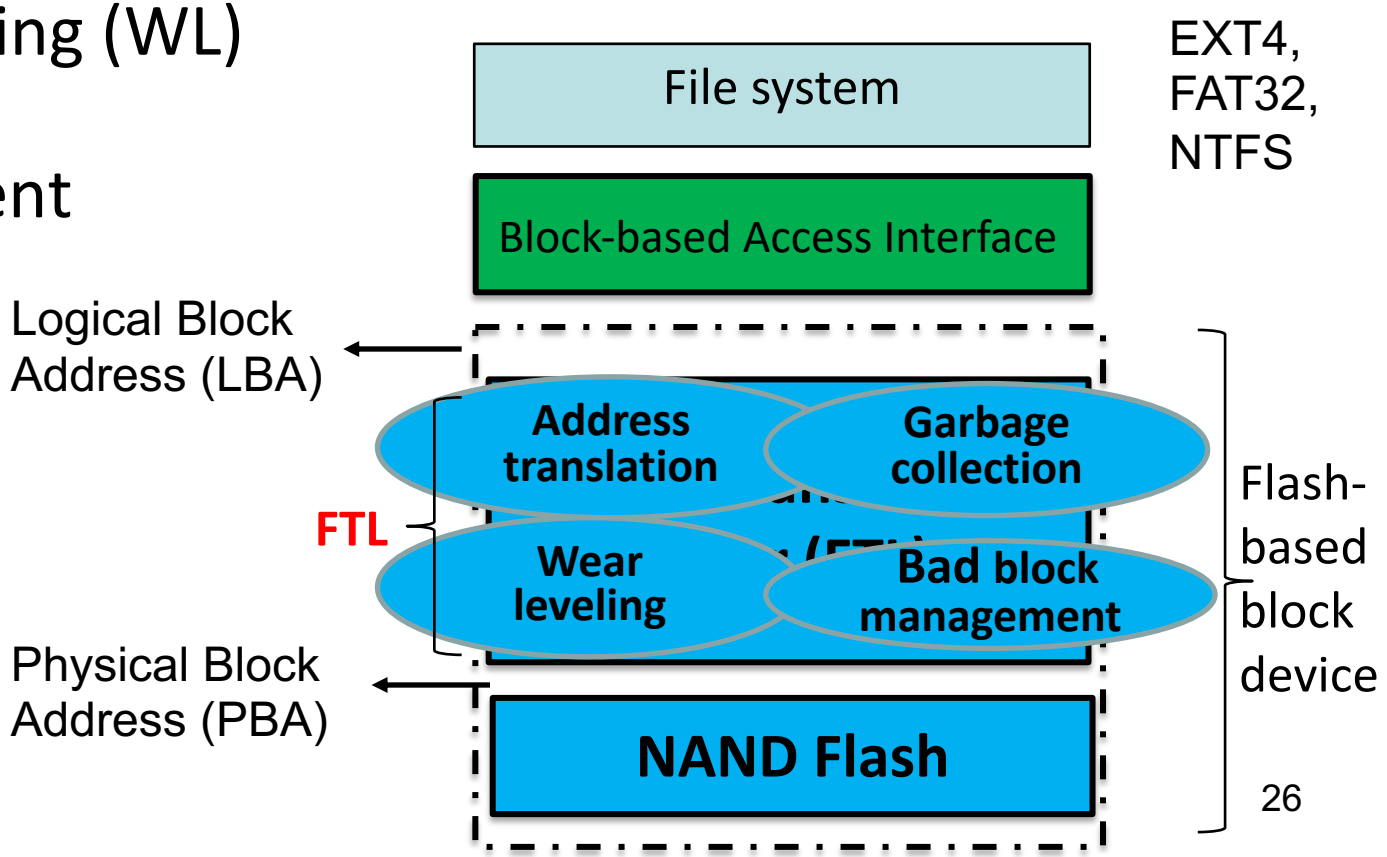
- Flash-specific file systems, which can handle the special characteristics of NAND flash
  - YAFFS/YAFFS2, UBIFS, F2FS, JFFS/JFFS2

- 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)

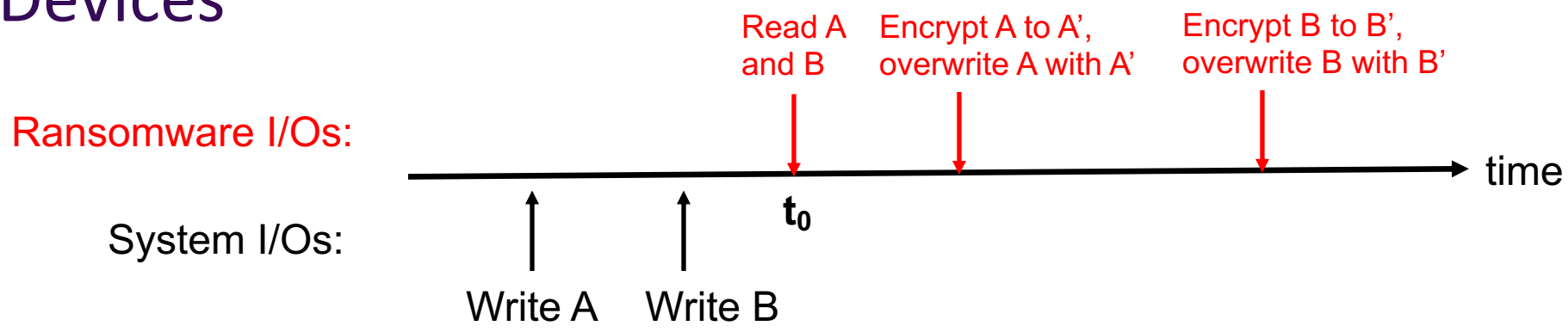
FTL usually provides the following functionality:

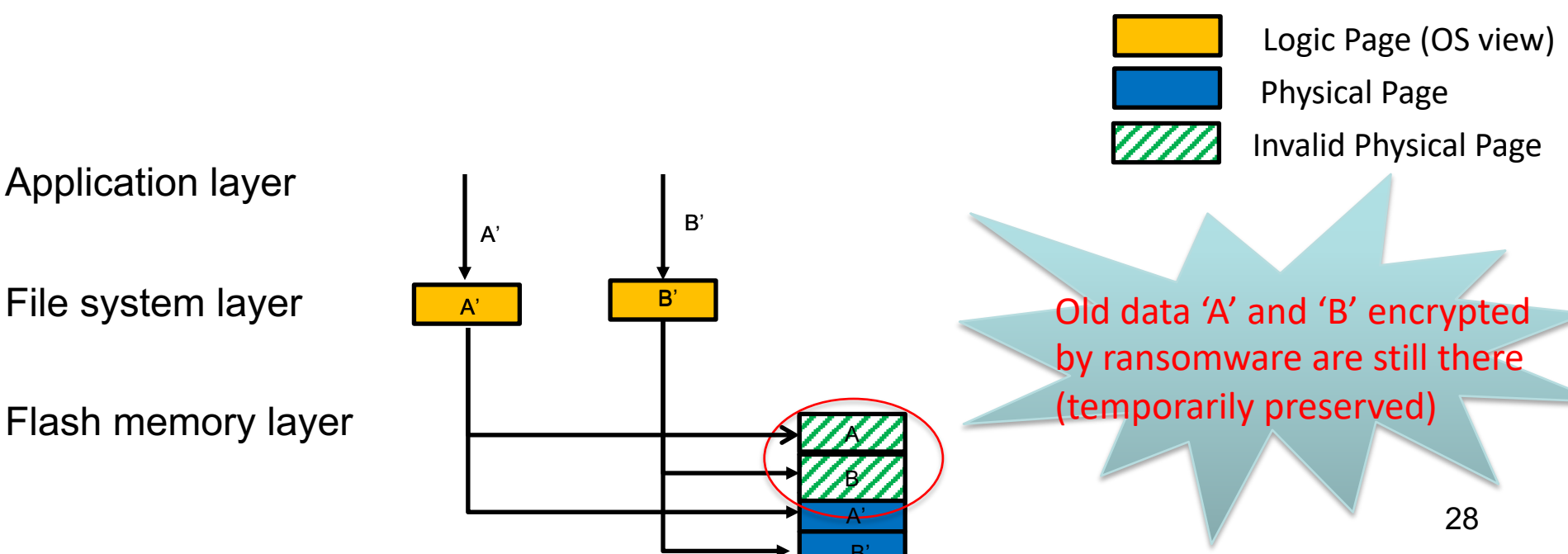
- ✓ Address translation
- ✓ Garbage collection (GC)
- ✓ Wear leveling (WL)
- ✓ Bad block management



Solution on Restoring Data Stored in A Mobile Device to The Corruption Point after Ransomware Attack

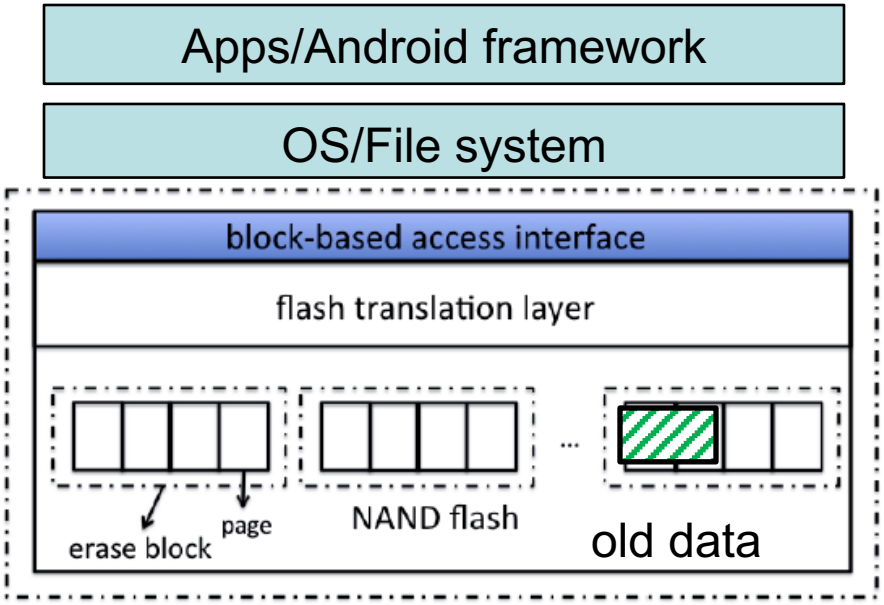
Towards Restoring The Corruption Point in Mobile Devices





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%)

Acknowledgments

- Our data recovery project is currently supported by US National Science Foundation under grant number 1938130-CNS