



Michigan
Technological
University



CS5740/4740 Spring 2023: Special Topic on Data Security (2)

Enabling Secure Data Recovery

Bo Chen

Department of Computer Science
Michigan Technological University

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

<https://snp.cs.mtu.edu>

bchen@mtu.edu

Data Security

- Data security: protecting digital data, such as those in a database, from destructive forces and from the unwanted actions of unauthorized users, such as a cyberattack or a data breach (wikipedia)



- Laws and regulations for data security
 - Family Educational Rights and Privacy Act (FERPA) (US)
 - Health Insurance Portability and Accountability Act (HIPAA) (US)
 - General Data Protection Regulation (GDPR)(EU): organizations may face significant penalties of up to €20 million or 4% of their annual revenue if they do not comply with the regulation
 - Data Protection Act (DPA) (UK)
 - Personal Information Protection and Electronic Documents Act (PIPEDA) (CA)

Data Security (cont.)

- Confidentiality/Privacy – has been discussed in our special topic on data security (1)
- Integrity checking and recoverability – **will be discussed in this talk**
- Secure deletion - will be discussed in the next talk
- Access control/flow control/inference control

Denning, Dorothy E., and Peter J. Denning. "Data security." ACM Computing Surveys (CSUR) 11, no. 3 (1979): 227-249.

Outline

- Data Integrity Checking and Recovery in the Public Clouds (**infrastructures**)
- Data Recovery from Malware Attacks in Personal Computers/Mobile Devices (**terminal devices**)

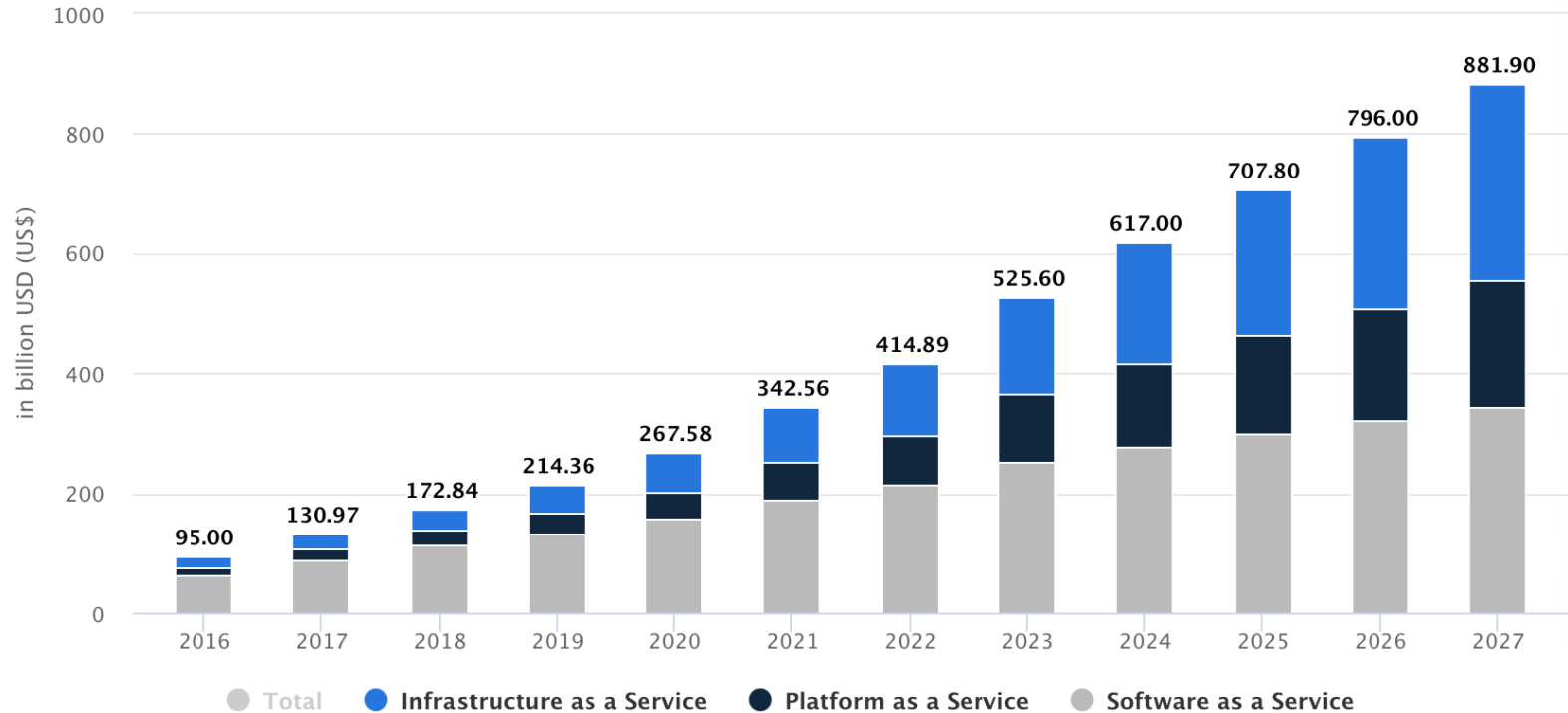
Outline

- Data Integrity Checking and Recovery in the Public Clouds
- Data Recovery from Malware Attacks in Personal Computers/Mobile Devices

Clouds are Everywhere Today

Source: Statista

REVENUE BY SEGMENT



Major cloud service providers



Traditional Cloud Storage Is Fully Centralized

- The cloud storage provider (CSP) creates, manages, and maintains dedicated IT infrastructures/data centers
 - Users outsource their data to the CSPs' data centers



<https://aws.amazon.com/about-aws/global-infrastructure/>



Traditional Cloud Storage Is Fully Centralized (cont.)

- Pros and cons:
 - Pros:
 - easy deployment, easy management
 - Cons:
 - dedicating computing infrastructure, leading to high cost of creating the cloud and hence high price of cloud usage
 - vulnerable to unexpected instances like power outage, flooding
 - do not scale well for the large number of IoT devices

Transitioning Centralized Cloud Storage to Decentralized Cloud Storage

- **Decentralized** cloud storage: connect users who need file storage with hosts worldwide offering **underutilized** hard drive capacity
 - The idea is similar to the **sharing economies** like Airbnb
 - Users from the network form **virtual data centers**



The Cloud Is Untrusted

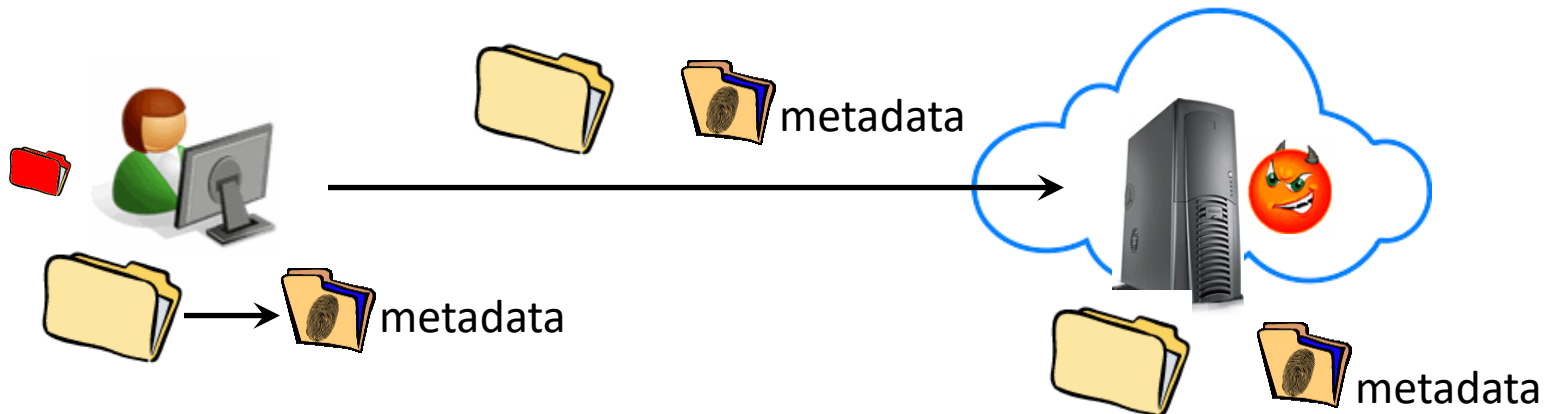
- Data owners may be **reluctant** to outsource their data to the cloud
 - Amazon S3, Microsoft Azure storage, Dropbox, Google Drive, Microsoft OneDrive, etc (centralized cloud providers)
 - FileCoin, Storj, Sia, etc (decentralized cloud providers)
- The cloud server is **not necessarily trusted**, and 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)

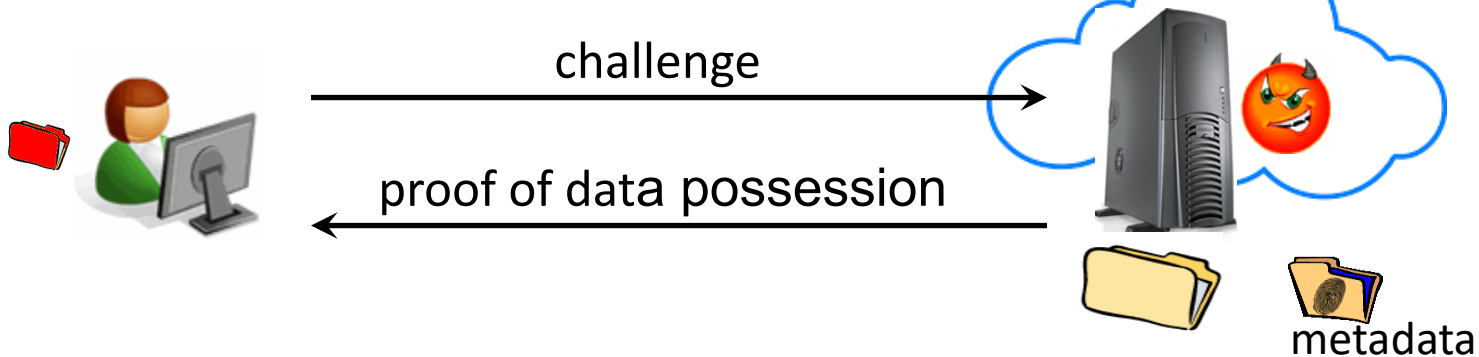
- Remote Data integrity Checking (RDC) allows the data owner to check the **integrity** of data stored at an **untrusted cloud provider**
 - RDC [Ateniese et al., CCS '07; Juels et al., CCS '07; Shacham et al., ASIACRYPT '08]

Setup

Client may now delete the file

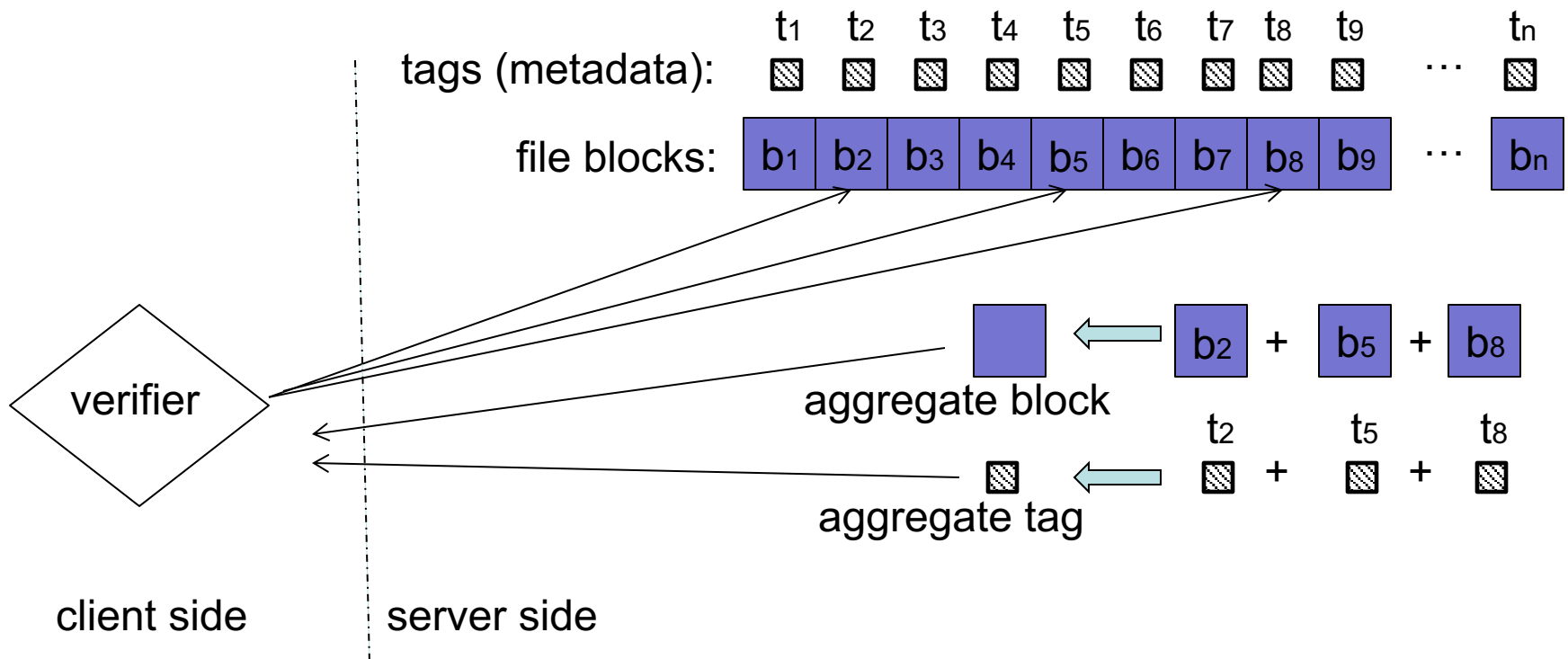


Challenge (periodically)



How Can RDC Efficiently Check Integrity of Big Data?

- 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]

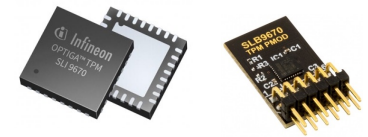


Who Can be The Trusted Verifier?

- It is difficult to find a trusted verifier, as any computer is vulnerable and may be compromised
 - This is more significant for a decentralized cloud storage
- How can we ensure the trustiness of the verifier even if its OS is compromised?
 - Trusted hardware

Trusted Hardware

- Trusted platform module (TPM): also known as ISO/IEC 11889, is an international standard for a **secure cryptoprocessor**, a dedicated microcontroller designed to secure hardware through integrated cryptographic keys



- Need an independent co-processor
- Disadvantages: need to purchase additional hardware, a lot of energy consumption
- Trusted execution environment (TEE): **a secure area of a main processor**
 - A TEE as an isolated execution environment provides security features such as isolated execution, integrity of applications executing with the TEE, along with confidentiality of their assets
 - It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity
 - Advantage: **TEE is a part of the existing processor, and no need to purchase additional hardware**



Trusted Execution Environment (TEE)

- Intel
 - Trusted Execution Technology
 - Software Guard Extensions (SGX)
 - "Silent Lake" (available on Atom processors)
- AMD
 - Secure Encrypted Virtualization (SEV)
 - Secure Memory Encryption (SME)
 - Transparent SME (TSME)
- ARM (mostly for embedded systems and mobile devices)
 - TrustZone
- IBM
 - IBM Secure Service Container
 - IBM Secure Execution
- Etc.

Intel SGX

- **Intel** Software Guard Extensions (SGX): integrated into Intel processors 7th generation (or later)
 - Personal computers/Servers
 - A set of security-related instruction codes that are built into some modern Intel CPUs
 - A pivot by Intel in 2021 resulted in the deprecation of SGX from the 11th and 12th generation Intel Core Processors, but development continues on Intel Xeon for cloud and enterprise use.
- Allow user-level as well as operating system code to define private regions of memory (enclaves)
 - The enclave is encrypted/ decrypted using keys only accessible to the processor (**the keys are not able to be extracted by OS/ software**)
 - Both the confidentiality and integrity of the contents in the enclave are protected and unable to be either read or saved by any process outside the enclave itself
 - **The assumption is that Intel is trusted**



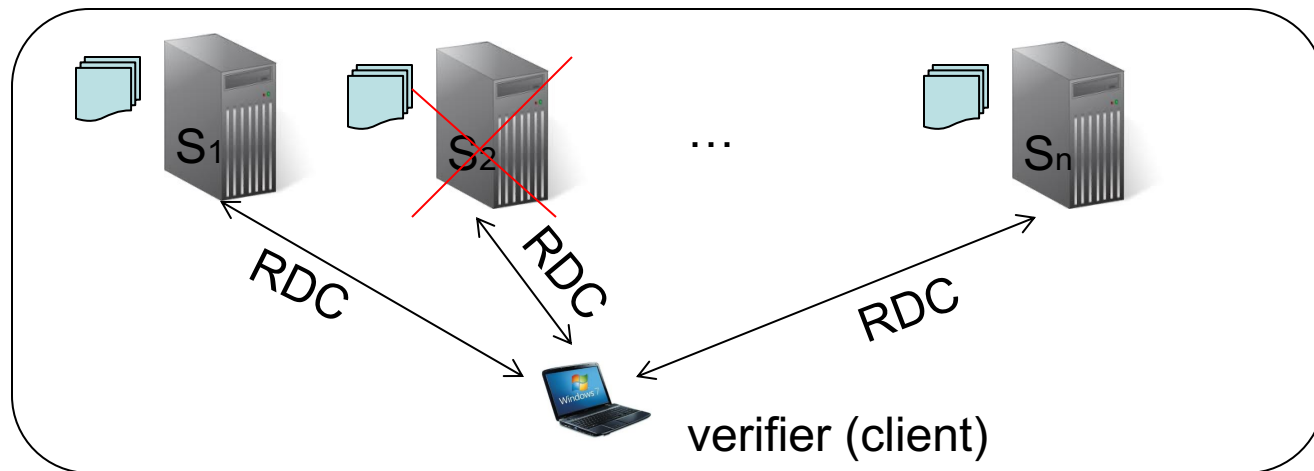
Small Corruption

- 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

How to Enable Data Recovery Once Corruption Is Detected?

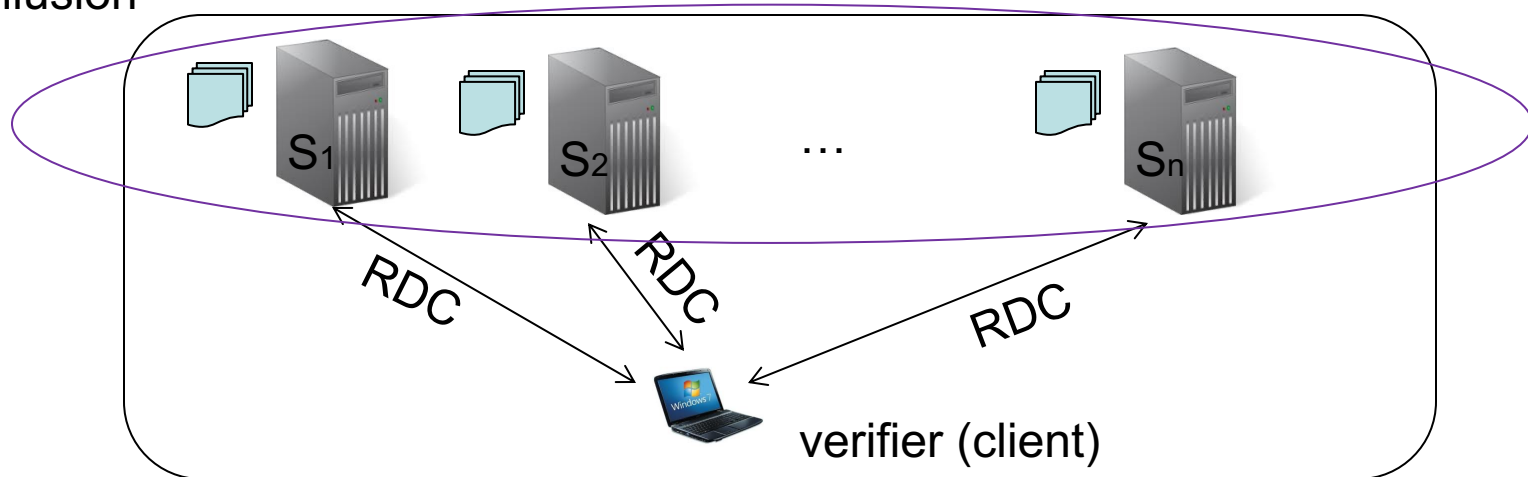
- Ensure long-term data reliability
- Data should be stored redundantly at multiple servers/data centers
 - Replications
 - Erasure coding
 - Network coding



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
 - Differentiated the replicas

collusion



Curtmola, Reza, Osama Khan, Randal Burns, and Giuseppe Ateniese. "MR-PDP: Multiple-replica provable data possession." In 2008 the 28th international conference on distributed computing systems, pp. 411-420. IEEE, 2008.

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 Issues

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

Bo Chen and Reza Curtmola. Remote Data Integrity Checking with Server-Side Repair. *Journal of Computer Security*, vol. 25, no. 6, pp. 537-584, 2017.

Bo Chen, Anil Kumar Ammala, 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

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

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

Outline

- Data Integrity Checking and Recovery in The Public Clouds
- Data Recovery from Malware Attacks in Personal Computers/Mobile Devices

Ransomware

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

You should be familiar with it if you are from CS5472 class

Growth in Ransomware Variants Since December 2015



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

Main Types of Ransomware

- Locker ransomware
- Crypto-ransomware



How to Combat Locker Ransomware?

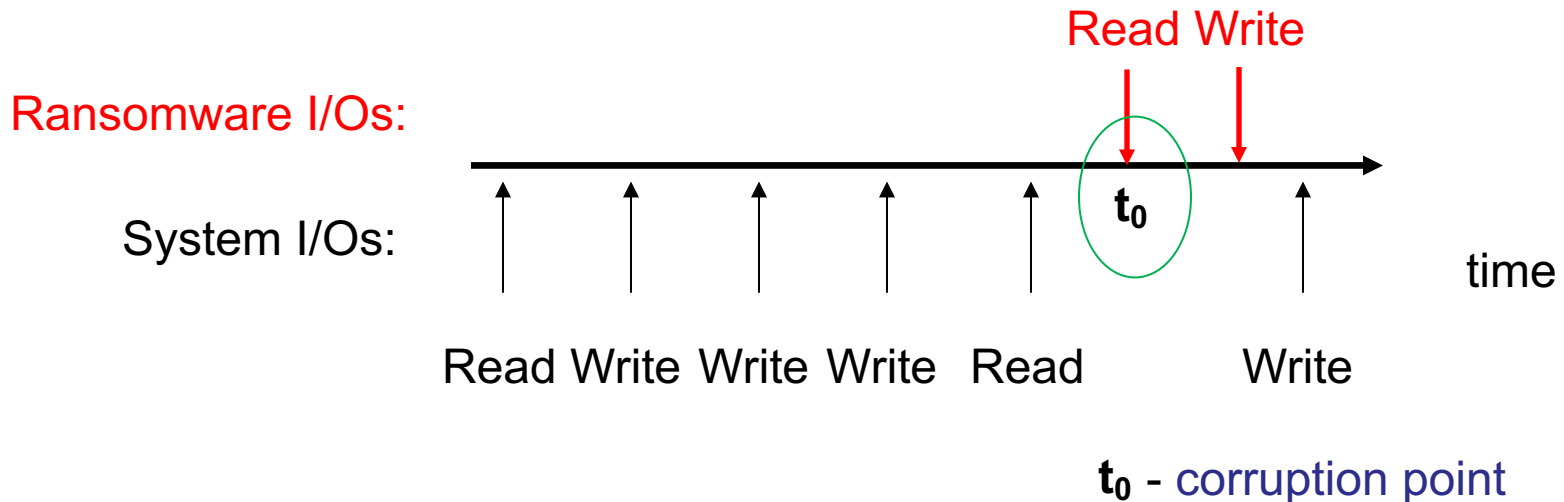
- Observation: only the system is locked by the ransomware, but the data are stored intact
- Unplug the storage medium (e.g., SSD drives, microSD cards), plug the storage medium to a new computing device, and copy out the data
- Plug the storage device back to the device which has been locked, and re-install/initialize 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

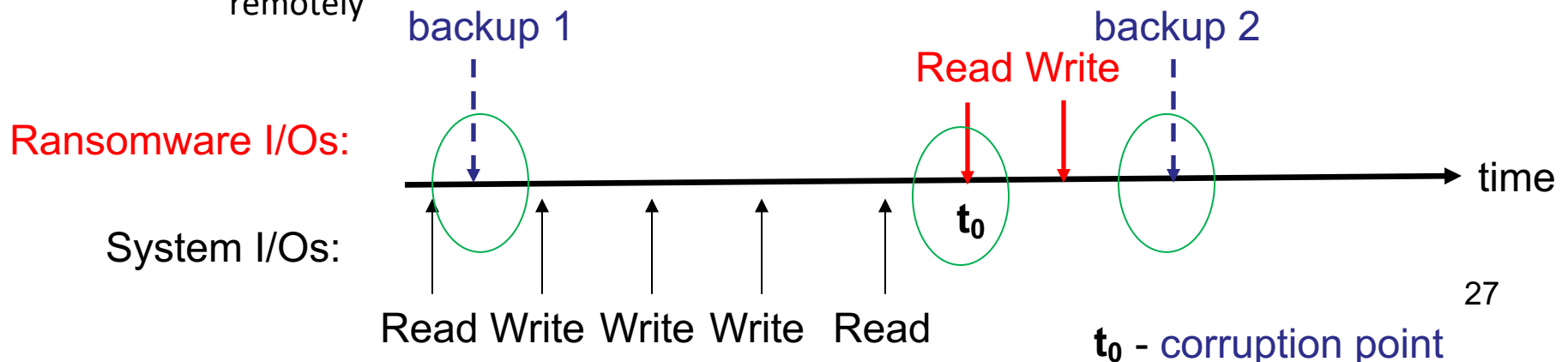
- After a computing device is hacked by ransomware, the victim data will be recovered by backups
- A challenging issue is how to *ensure data stored in the victim device is recoverable to the exact point right before the corruption* (i.e., corruption point)?



Wen Xie, Niusen Chen, and **Bo Chen**. Enabling Accurate Data Recovery for Mobile Devices against Malware Attacks. 18th EAI International Conference on Security and Privacy in Communication Networks ([SecureComm '22](#)), Kansas City, Missouri, October 2022.

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?

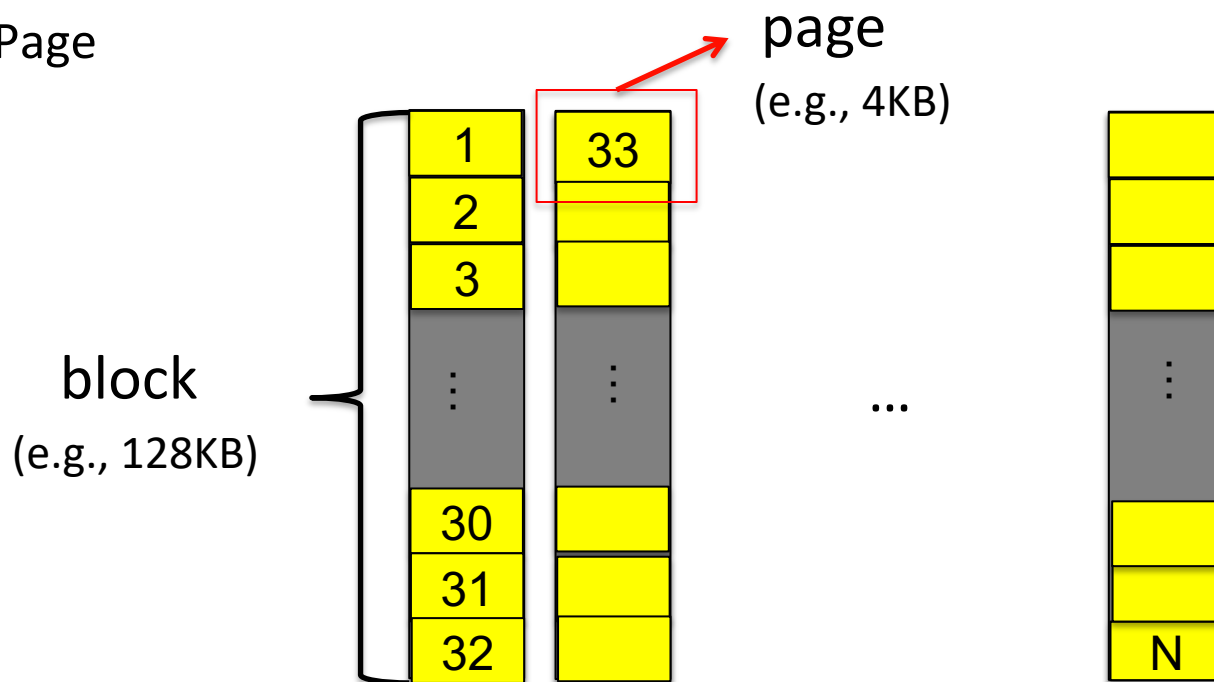
- Data can be backed up locally after each single write
- This could be problematic:
 - Creating backups after each single write incurs **a large overhead**
 - The ransomware may **compromise the entire OS and all the local backups** created at the upper layers may be corrupted and cannot used for data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

Background on Flash Memory

NAND Flash Memory



- NAND flash memory as mass storage
 - SSD (used widely in personal computers and servers)
 - Flash memory cards like SD cards, MMC card, UFS cards (used broadly in mobile devices/IoT devices)
- NAND flash organization
 - Block
 - Page



How to Program Flash Memory?

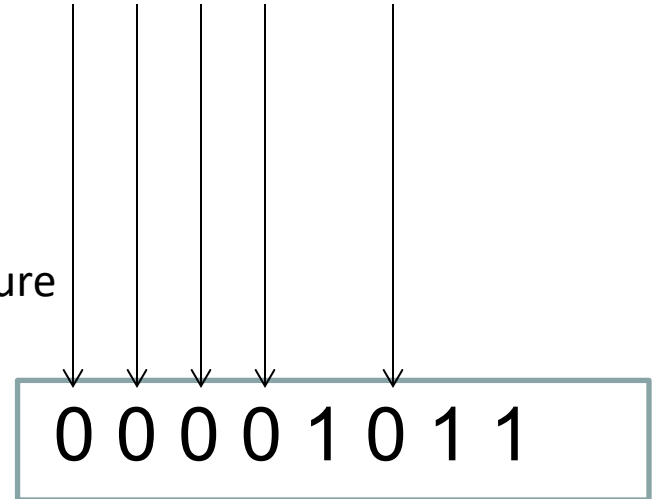
All '1' initially:



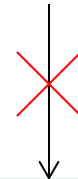
Write **0x0b**:

Rule:

- 1) 1 can be programmed to 0
- 2) 0 cannot be programmed to 1 except performing an erasure



Modify 0x0b to **0x0f**?



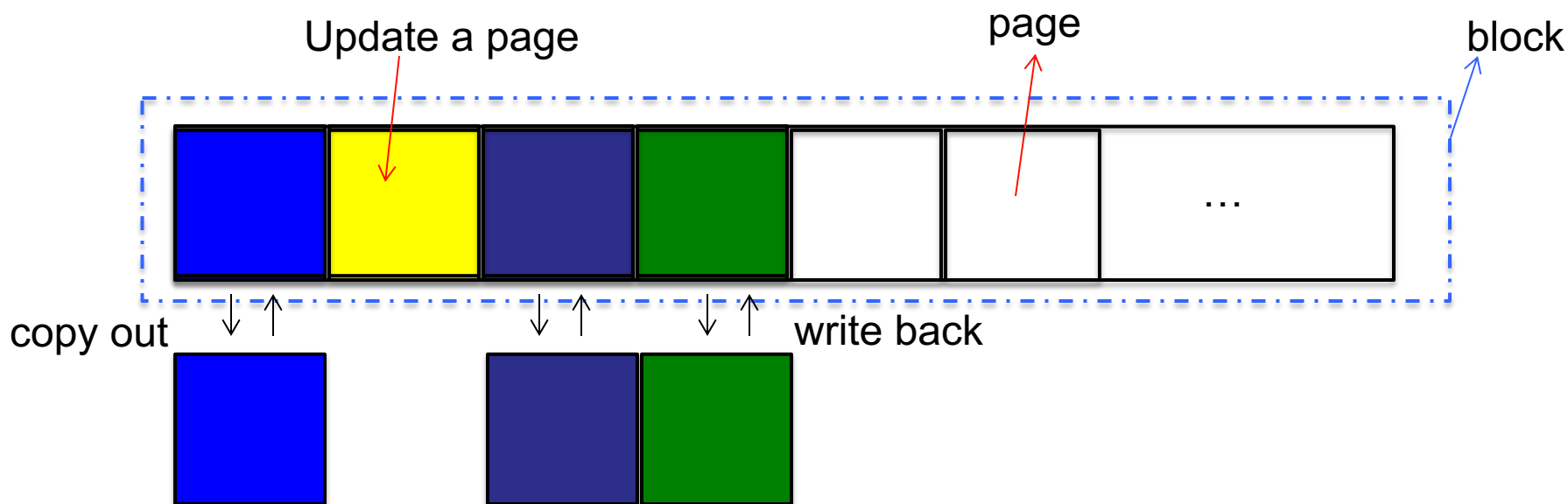
Need to erase to all '1' first



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)



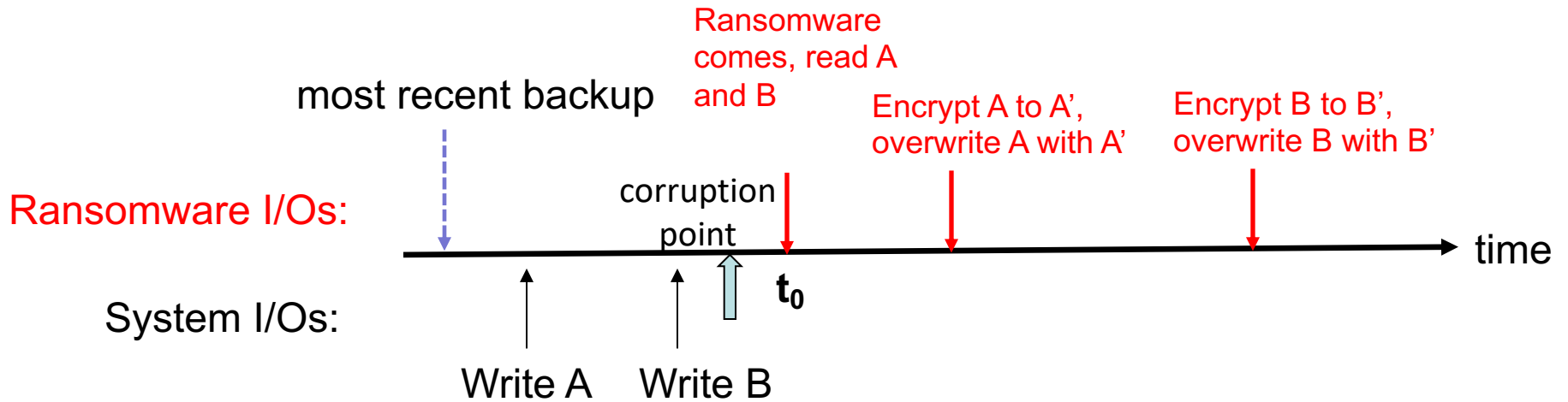
- Over-write may cause significant **write amplification**
- Usually prefer **out-of-place update** instead of in-place update

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**)

Solution on Restoring Data to The Corruption Point after Ransomware Attacks

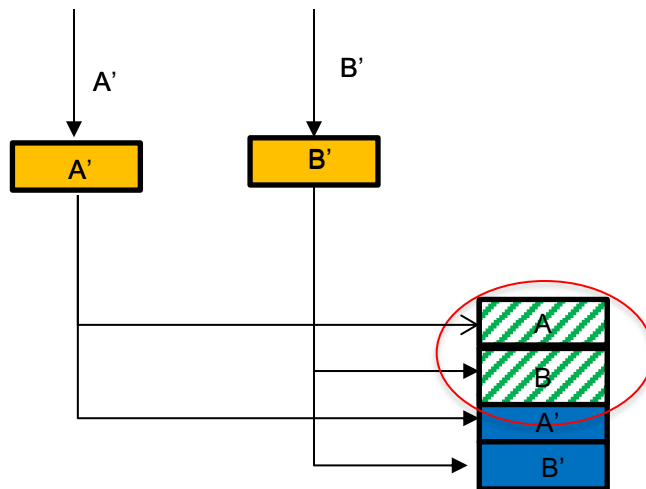
Towards Restoring The Corruption Point



Data at the corruption point = most recent backup + (A + B)

-  Logic Page (OS view)
-  Physical Page
-  Invalid Physical Page

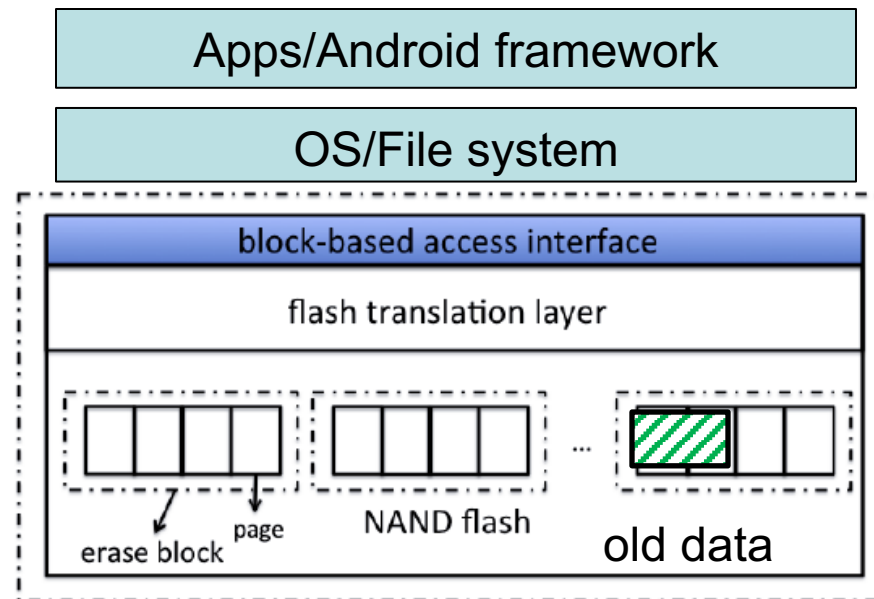
Application layer
File system layer
Flash memory layer



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

- 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: temporarily freeze the garbage collection on those flash blocks which hold the data not yet been backed up remotely.

A Few Additional Questions (cont.)

- Can I recover the data in a fine-grained manner?
 - For example, the victim user can choose which files to be restored rather than restoring the entire storage (slow, or may not be necessary)
- Integrate both file system forensics and the flash memory data extraction

Niusen Chen, Josh Dafoe, and **Bo Chen**. Poster: Data Recovery from Ransomware Attacks via File System Forensics and Flash Translation Layer Data Extraction. 2022 ACM Conference on Computer and Communications Security (**CCS '22**) Posters, Los Angeles, CA, November 2022.

Acknowledgments

- Our secure data recovery project is currently supported by US National Science Foundation under grant number 2225424-CNS

<https://snp.cs.mtu.edu/research/cloudsec.html>