CS5740 Special topic on data security (2)

Plausibly Deniable Encryption Systems for Mobile Devices

Bo Chen

Department of Computer Science

Michigan Technological University

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

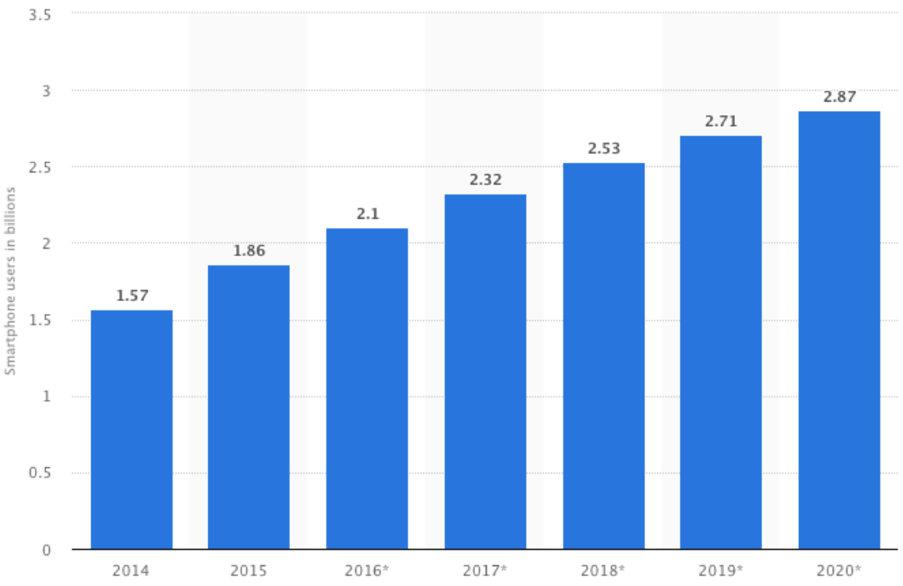
https://snp.cs.mtu.edu

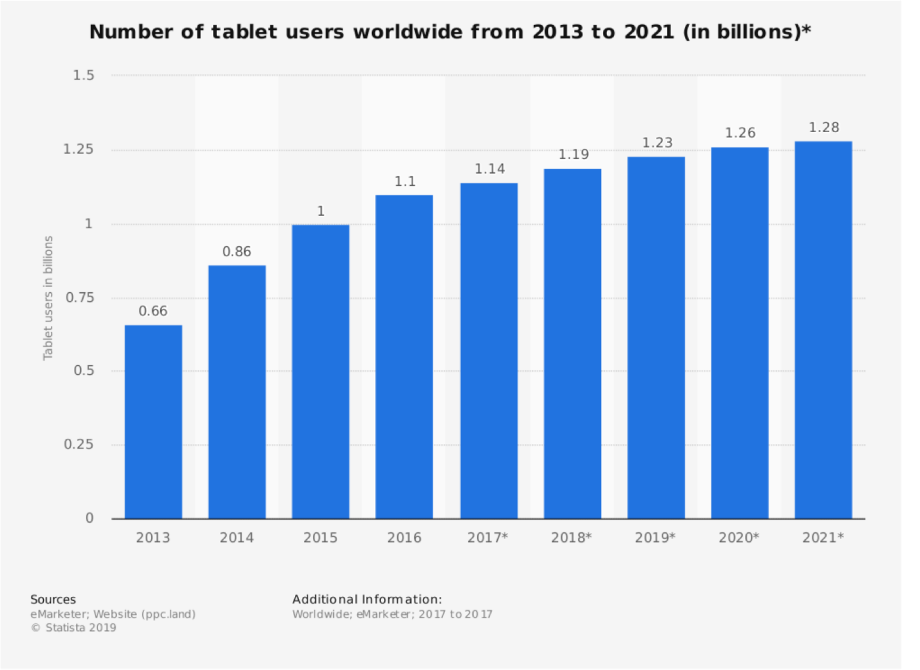
bchen@mtu.edu

Final Exam

- Thursday, April 30 from 3:00pm-5:00pm in Canvas
- Will enable the exam at 2:55pm.

Mobile Devices are Turning to Mainstream Computing Devices




Number of tablet users worldwide from 2013 to 2021 (in billions)*

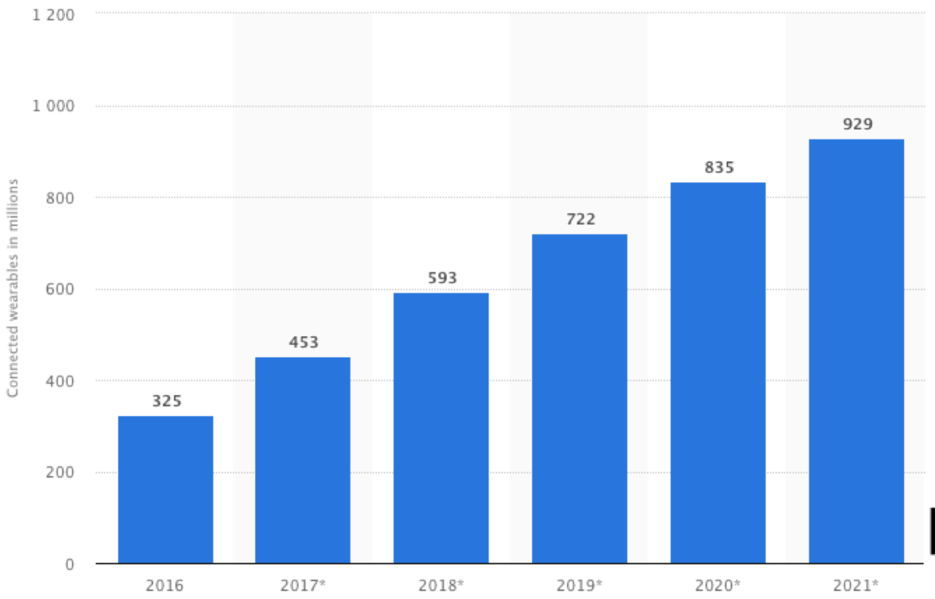
Sources
eMarketer; Website (ppc.land)
© Statista 2019

Additional Information:
Worldwide; eMarketer; 2017 to 2017

Number of smartphone users worldwide from 2014 to 2020 (in billions)

Number of tablet users worldwide from 2013 to 202 (in billions)

Mobile Devices are Turning to Mainstream Computing Devices (cont.)



Number of connected wearable devices worldwide from 2016 to 2021 (in millions)

Mobile Devices Are Increasingly Used for Critical Applications

- Mobile devices are increasingly used to handle sensitive data
  - Online banking
  - Ecommerce
  - Cryptocurrency/stock trading
  - Naked photos
  - A human rights worker collects evidence of atrocities in a region of oppre
  - Etc.

- Security issues in mobile computing devices
  - Confidentiality
  - Integrity
  - Authentication
  - Access control

Coercive Attack against Confidentiality

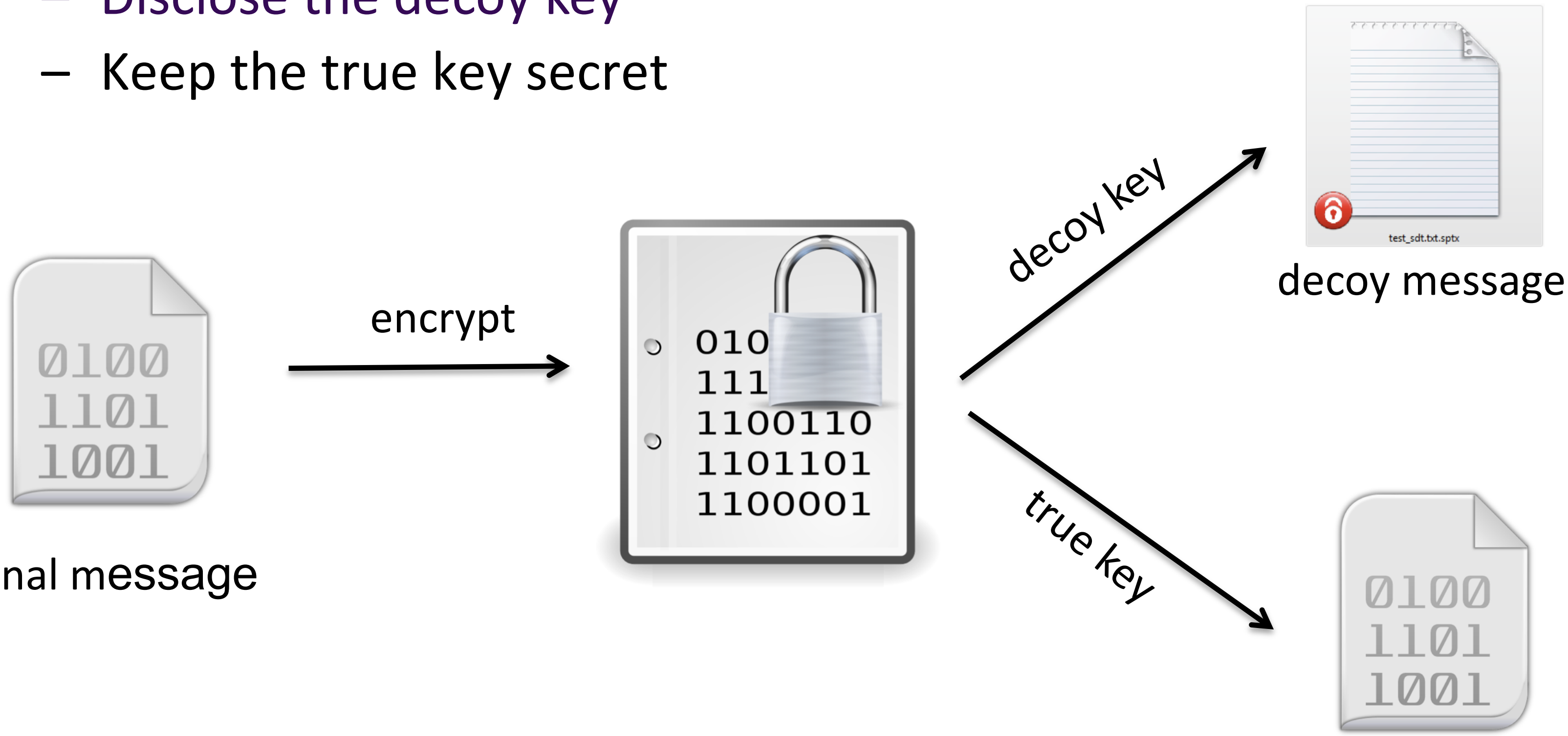
- To protect confidentiality of sensitive data, we can simply encrypt them
  - AES
  - 3DES

- Conventional encryption is vulnerable to a coercive attack

An attacker forces the device’s owner to disclose the decryption key

Plausible Deniable Encryption (PDE)

- Plausible Deniable Encryption (PDE) [Canetti et al., CRYPTO ’97]: a crypto primitive designed for mitigating coercive attacks
  - Disclose the decoy key
  - Keep the true key secret



encrypt

decoy key

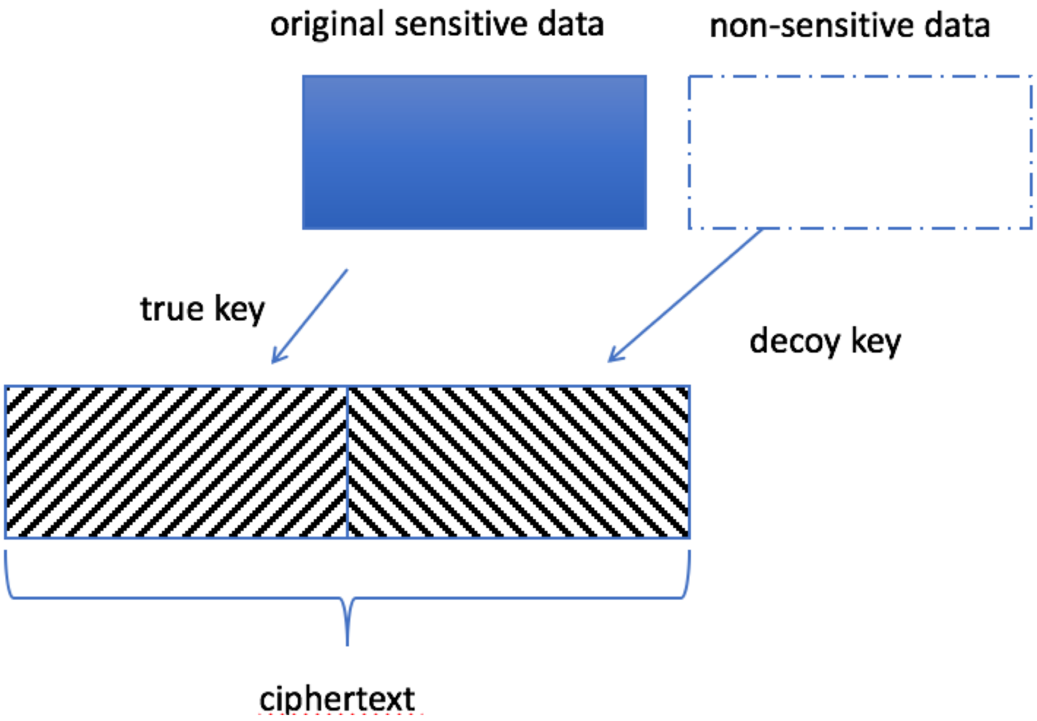
true key

original message

decoy message

original message

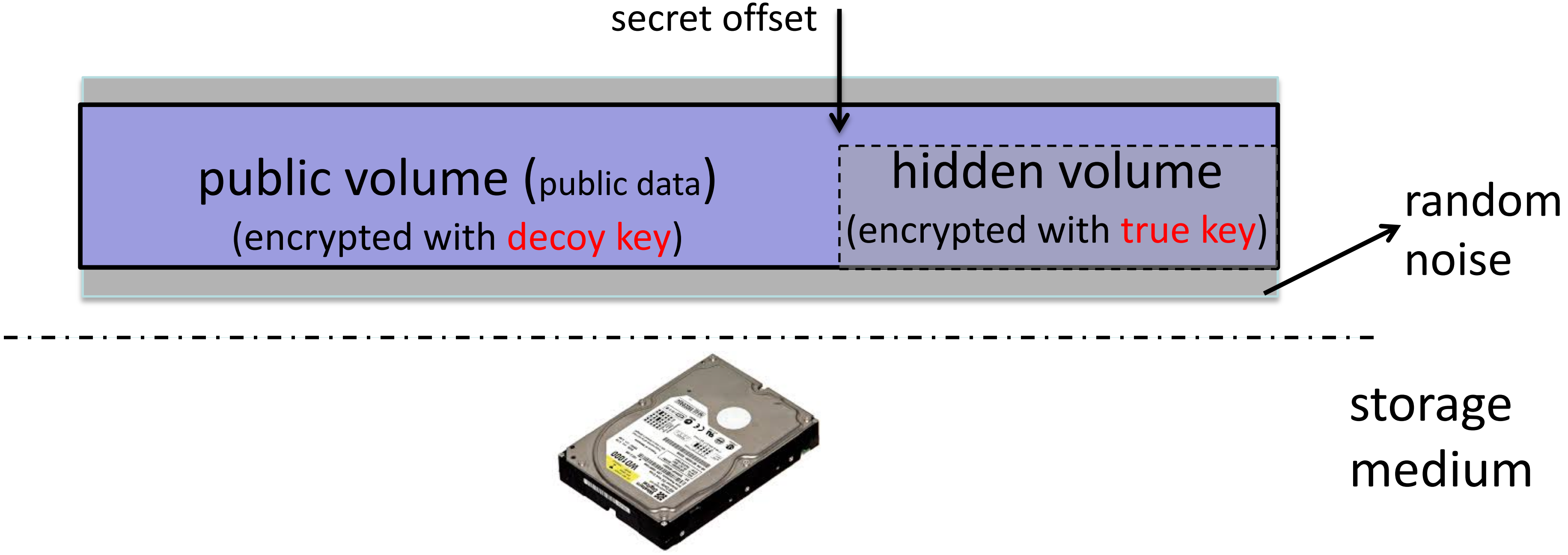
Instantiating PDE (Plausibly Deniable Encryption) in Cryptography



- Issues: the size of ciphertext is increased. Deniability is easily compromised

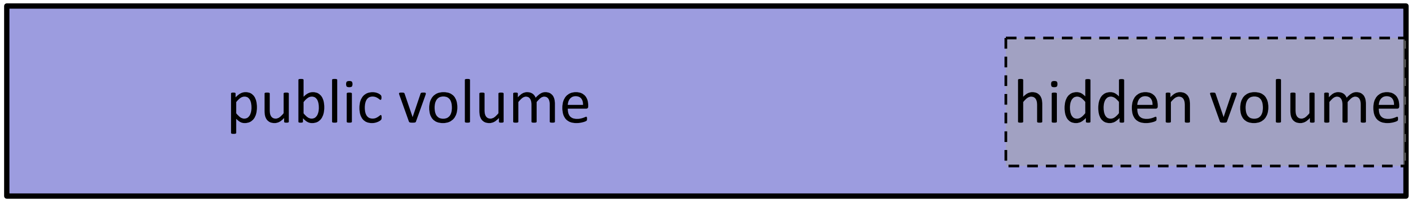
Implementing PDE in Systems - Hidden Volume

- Hidden volume [TRUECRYPT ’04] realizes the concept of PDE in systems
  - Only the decoy key will be disclosed
  - The encrypted hidden volume cannot be differentiated from the random noise


secret offset
public volume (public data)
(encrypted with decoy key)
hidden volume
(encrypted with true key)
random noise
storage medium

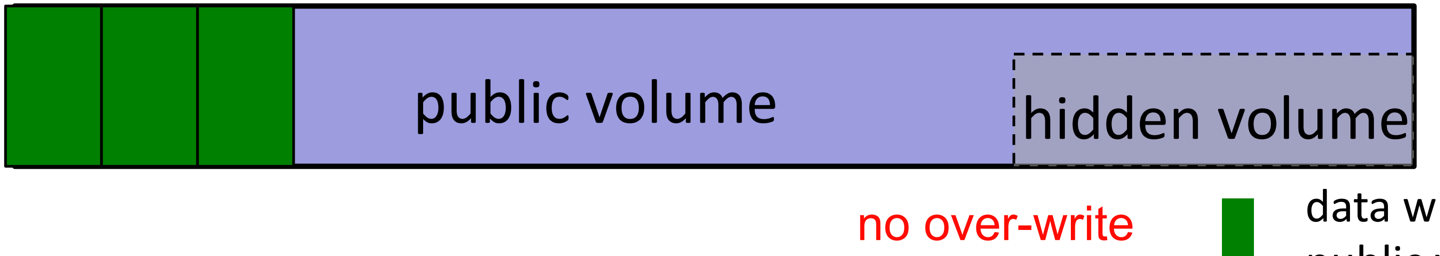
The Challenges: Over-writing Issues

- The data written to the public volume may over-write the data in the hidden volume
  - The hidden volume is part of the public volume



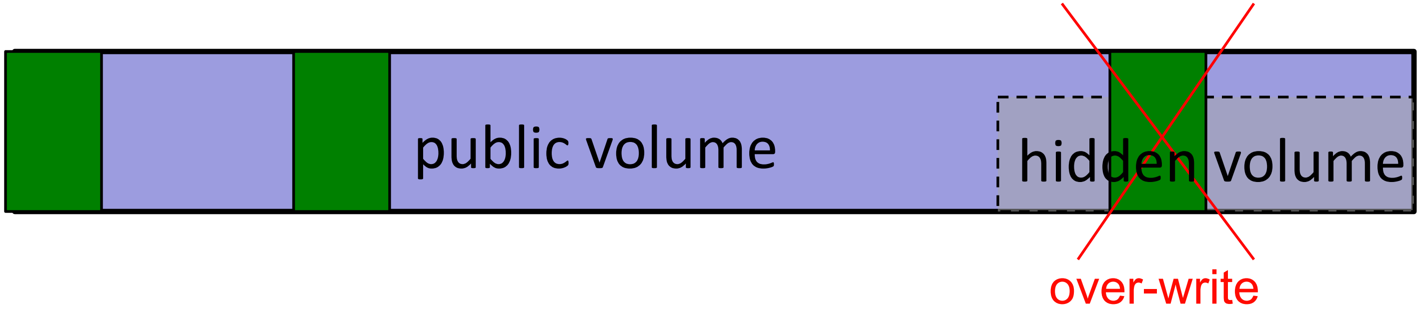
The Challenges: Over-writing Issues (cont.)

- File systems really matter for over-write issues
  - FAT allocates blocks sequentially



🟩 data written to public volume

  - EXT4 does not allocate blocks sequentially

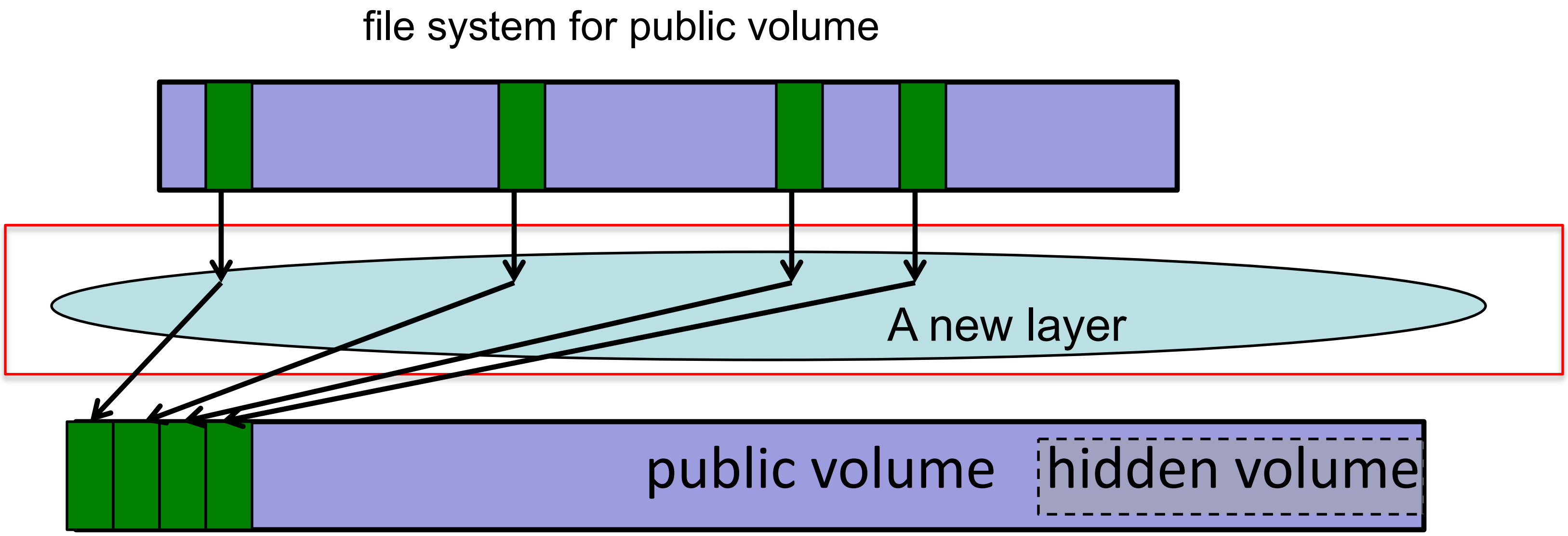


It is challenging to allow any file systems to be deployed while mitigating the over-write issues

MobiPluto – Key Insights [ACSAC '15]

To realize file system friendly design, a new layer is introduced to decouple the file system and the underlying PDE system

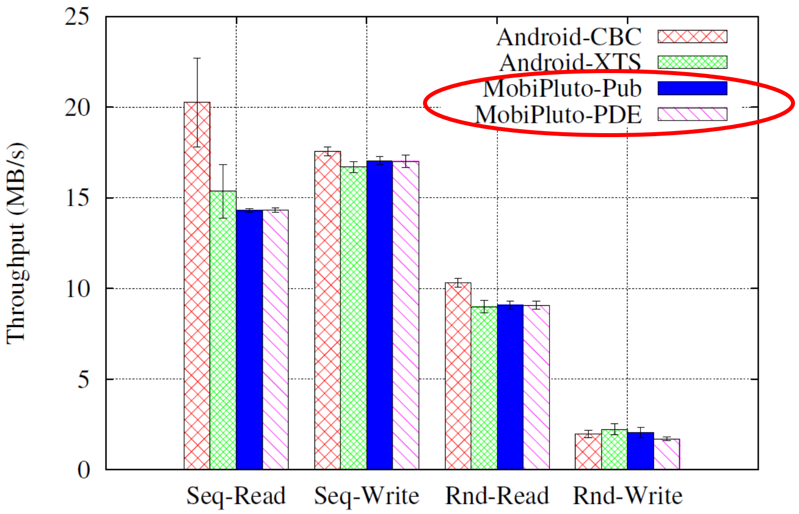
1. Provide virtual volumes to file systems
2. Any block-based file system can be built on a virtual volume
3. Non-sequential allocation on the virtual volume will be converted to sequential allocation on the underlying layer



Bing Chang, Zhan Wang, Bo Chen, and Fengwei Zhang. MobiPluto: File System Friendly Deniable Storage for Mobile Devices. 2015 Annual Computer Security Applications Conference (ACSAC '15), Los Angeles, California, USA, December 2015 (Acceptance rate: 24.4%)

Evaluation Highlights

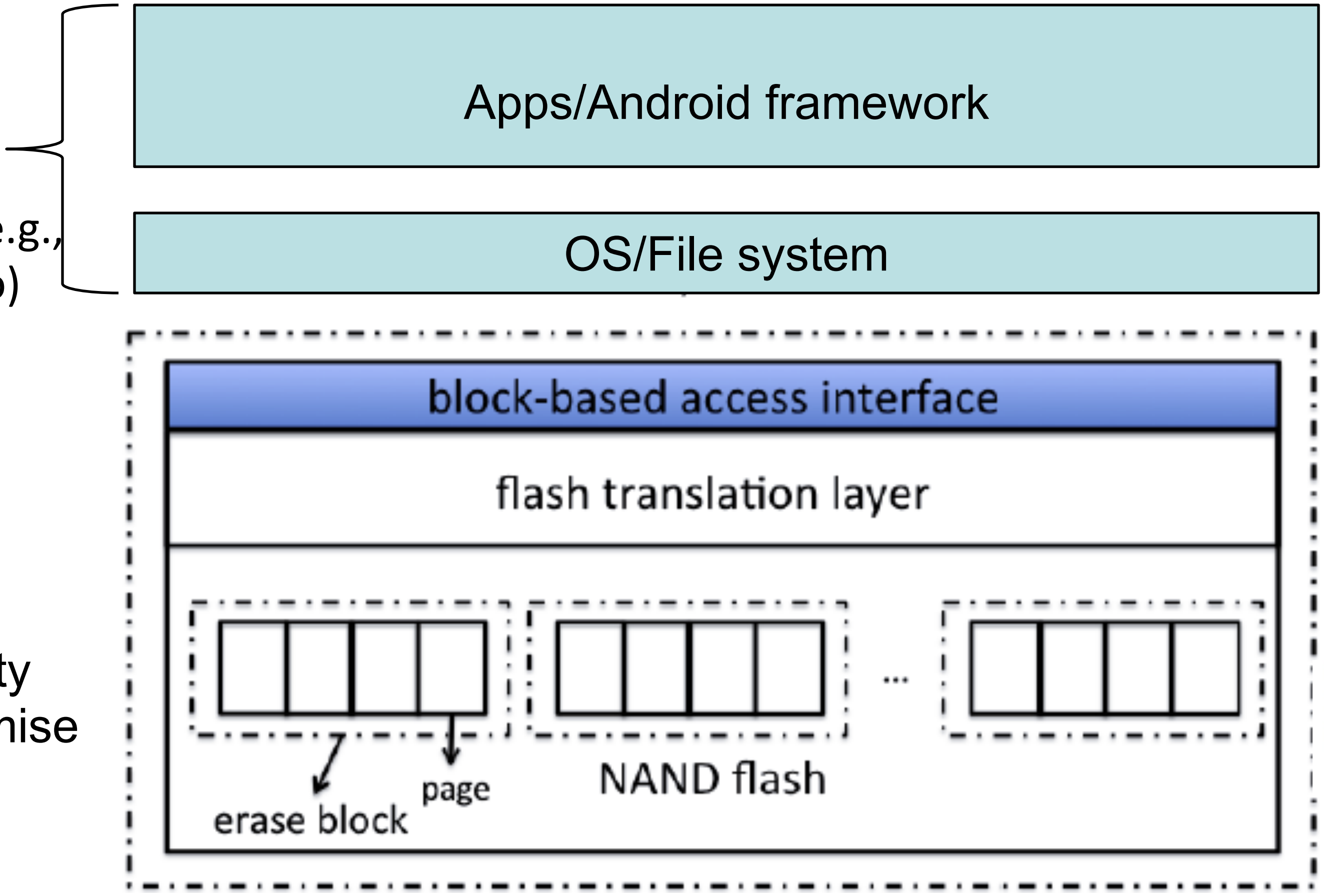
- Implemented a prototype of our solution on LG Nexus 4



Throughput (MB/s) from AndroBench

Deniability Compromise in The Lower Layer?

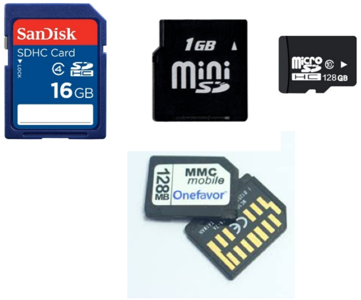
Existing deniable storage systems (e.g., MobiPluto)



✗

deniability compromise

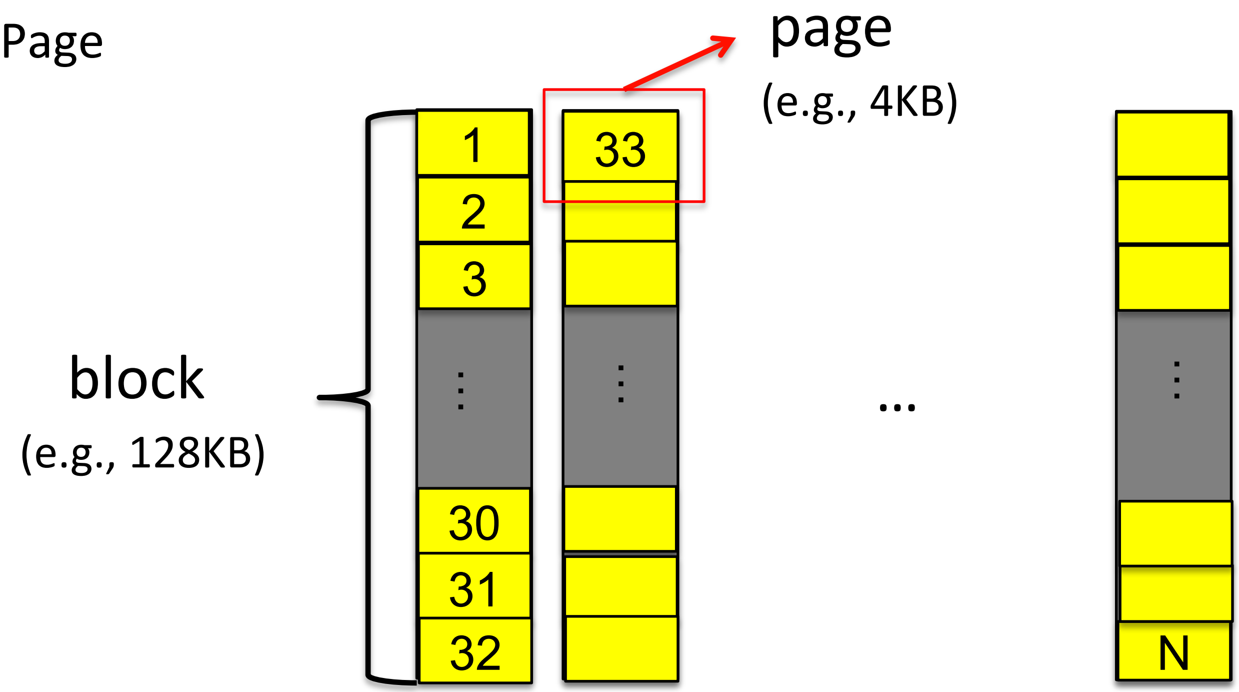
Flash storage (eMMC card/SD card)



NAND Flash Memory

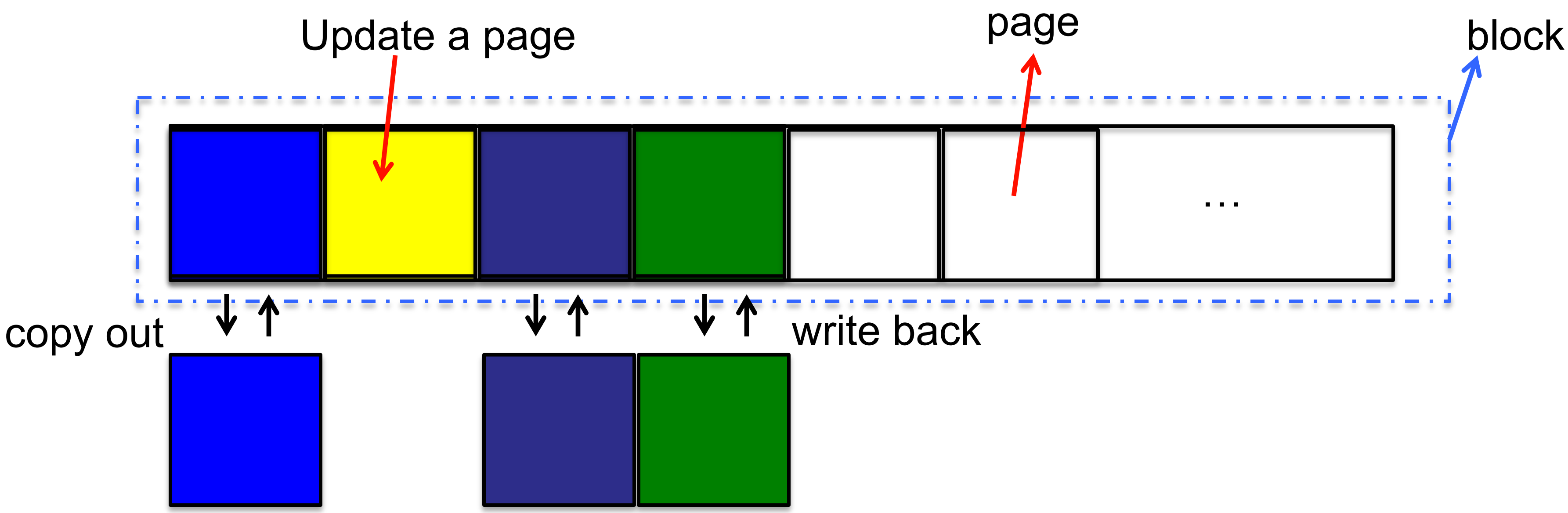
- Flash memory
  - NAND flash (broadly used for mass-storage devices)
  - NOR flash (used for storing program code that rarely needs to be updated, e.g., a computer's BOIS)

- NAND flash organization
  - Block
  - Page



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 NAND Flash

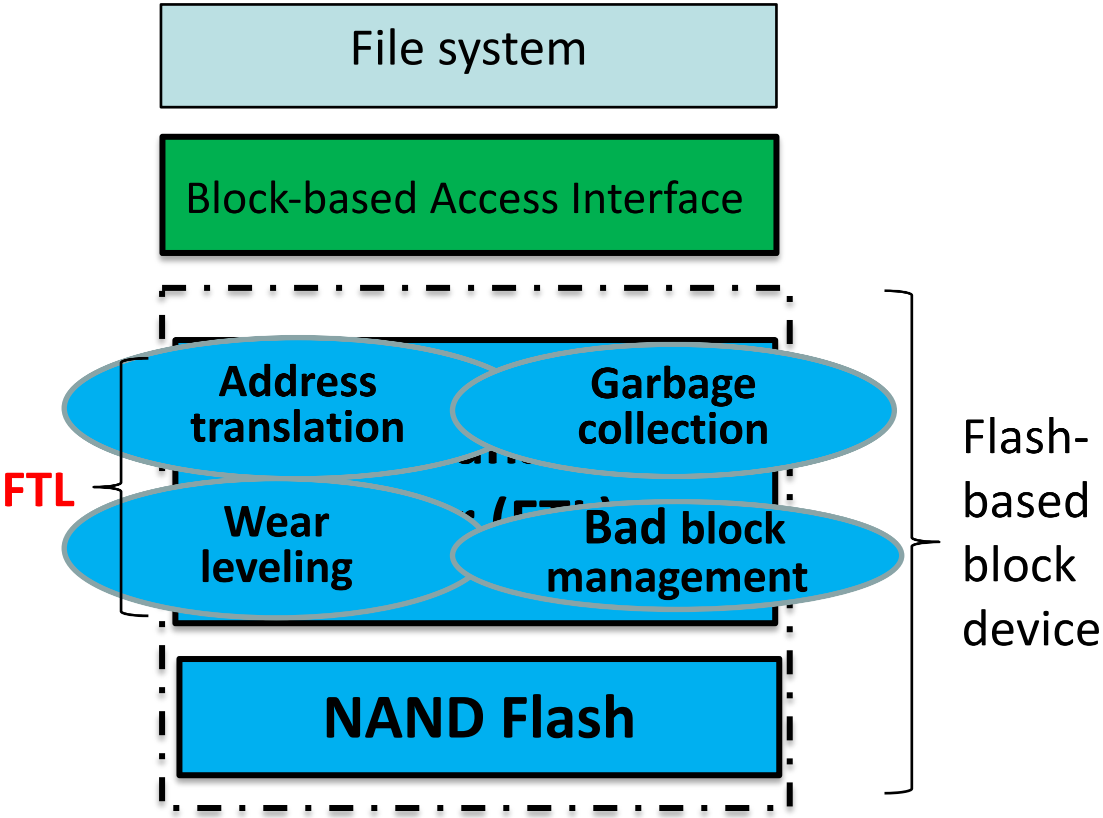
- Flash-specific file systems, which can handle the special characteristics of NDND flash
  - YAFFS/YAFFS2, UBIFS, F2FS, JFFS/JFFS2

- 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



File system

Block-based Access Interface

FTL

Address translation

Garbage collection

Wear leveling

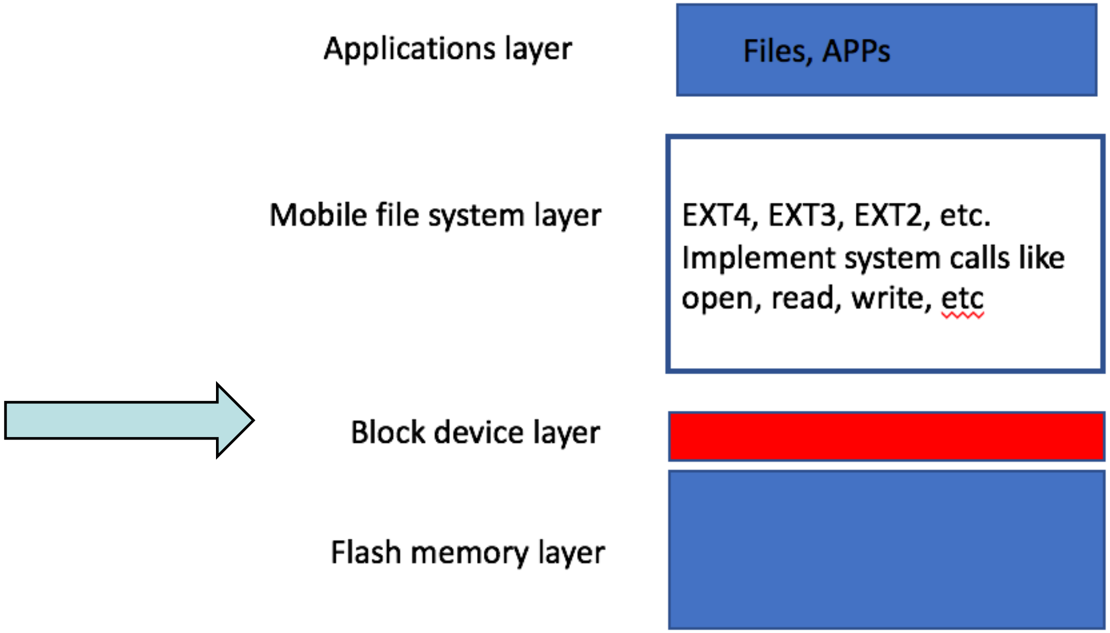
Bad block management

NAND Flash

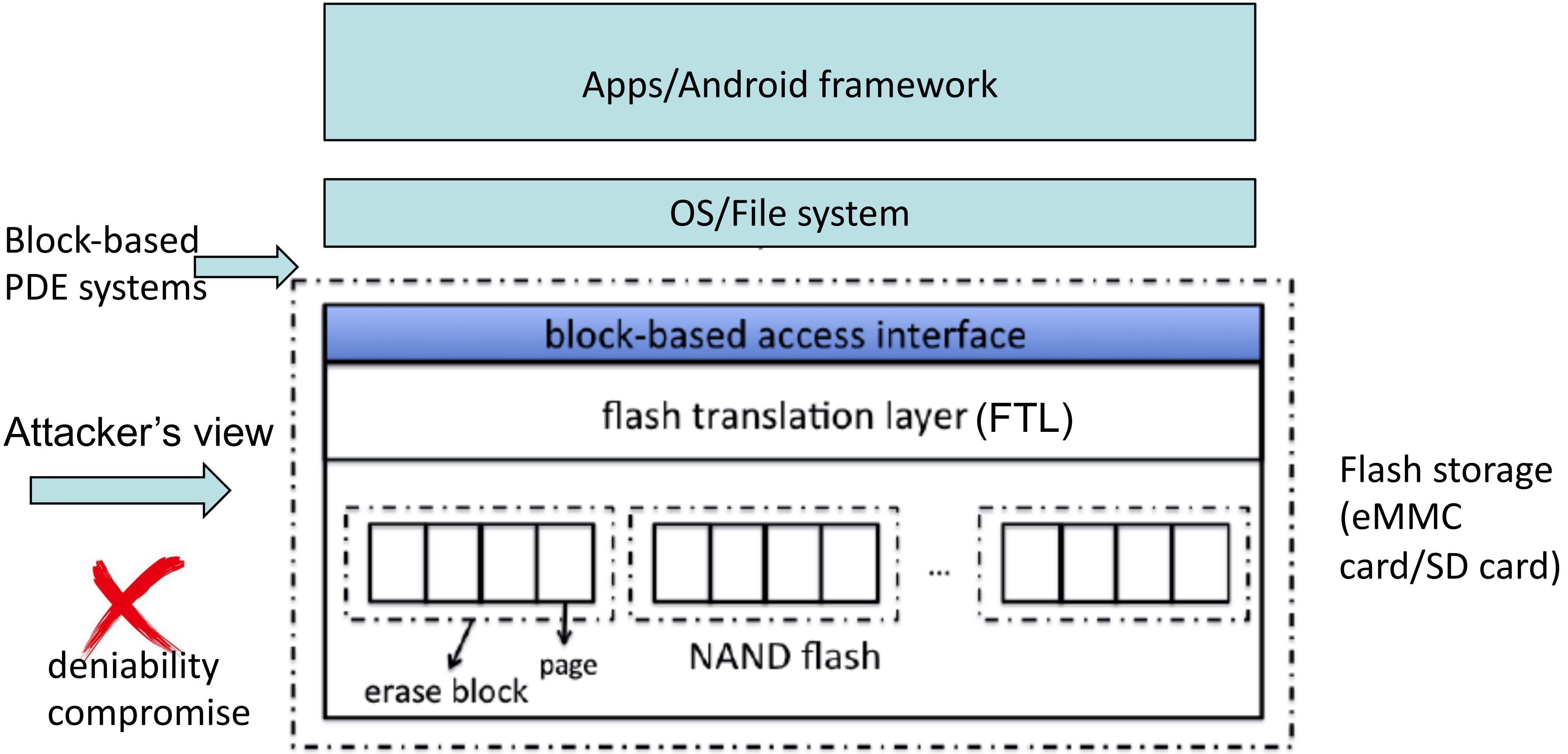
Flash-based block device

Existing PDE Systems for Mobile Devices

- Most of the existing PDE systems deploy hidden volume on top of the block device
  - Mobiflage [Skillen et al., NDSS ’13]
  - MobiHydra [Yu et al., ISC ’14]
  - MobiPluto [Chang et al., ACSAC ’15]
  - MobiCeal [Chang et al., DSN ’18]



Deniability May be Compromised When Deploying Hidden Volume on The Block Layer



Apps/Android framework

OS/File system

Block-based PDE systems

block-based access interface

flash translation layer (FTL)

Attacker’s view

erase block

page

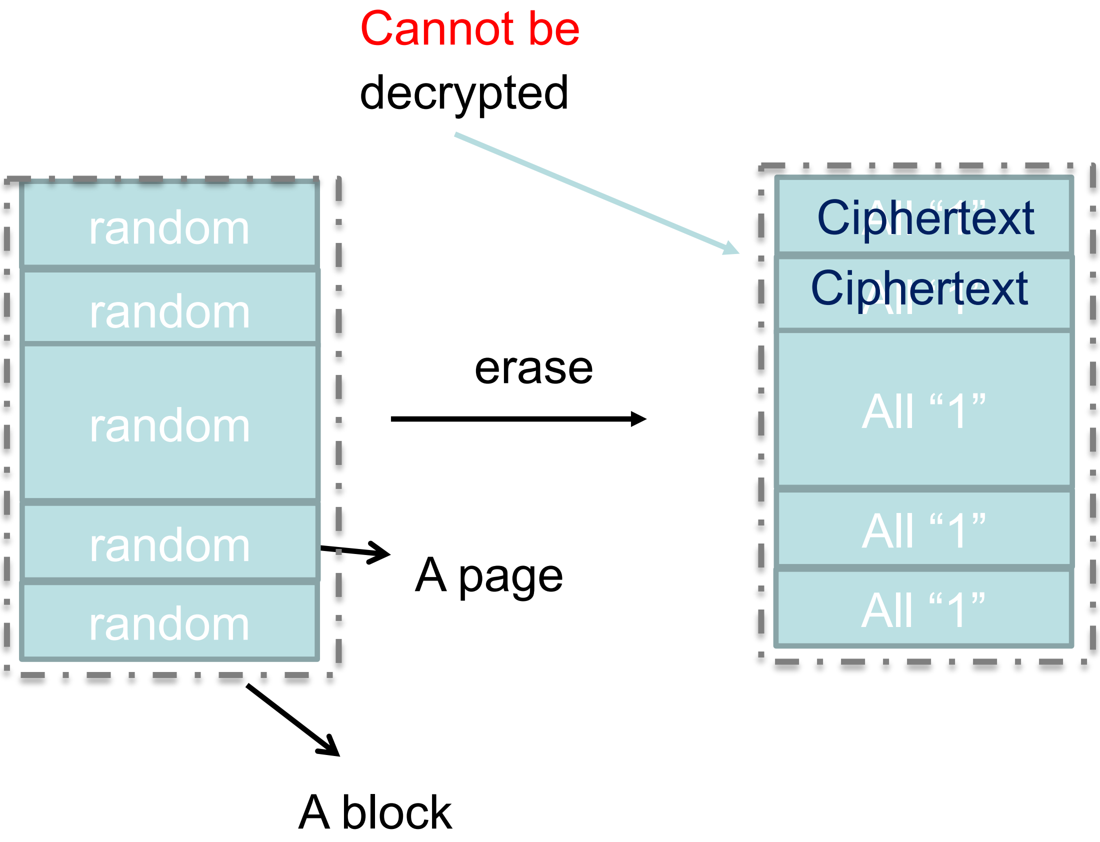
NAND flash

...

deniability compromise

Flash storage (eMMC card/SD card)

Compromise of Existing PDEs Built on top of the block device (1)



Cannot be decrypted

random

random

random

random

random

erase

A page

A block

Ciphertext

Ciphertext

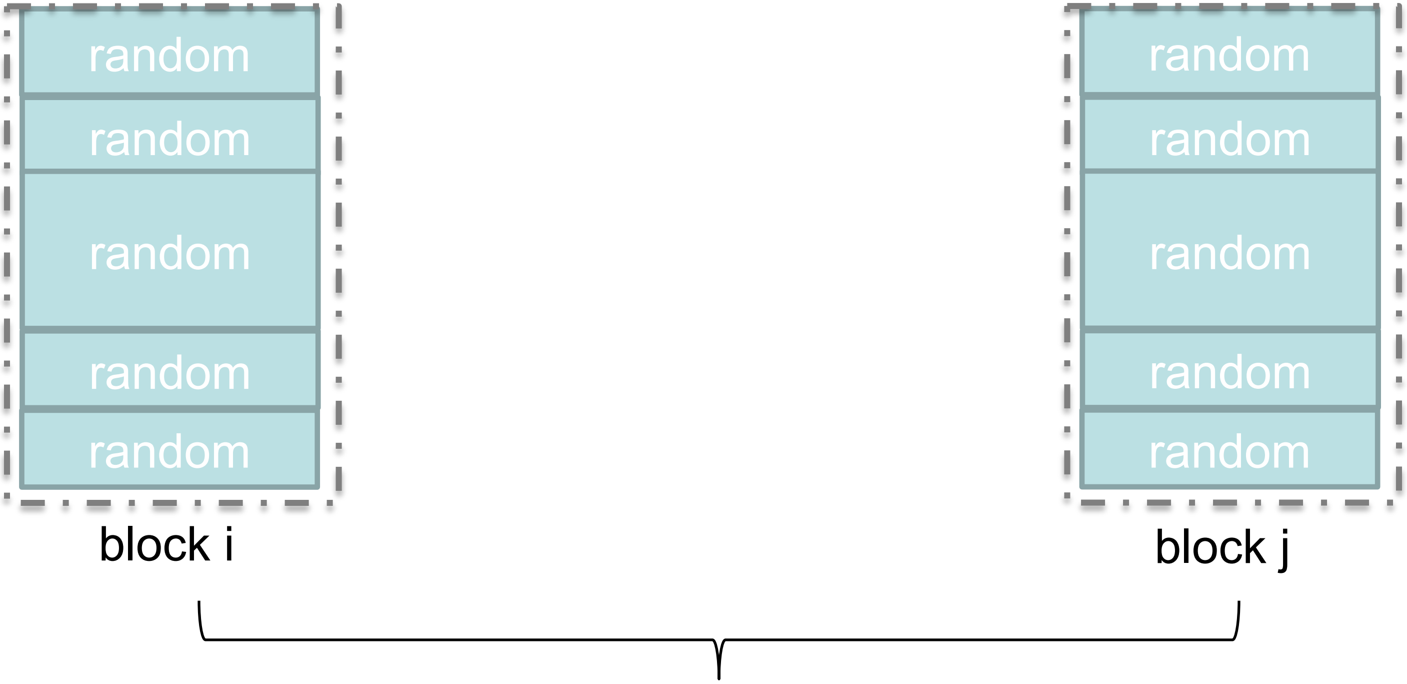
All “1”

All “1”

All “1”

A flash block partially used by the hidden volume

Compromise of Existing PDEs Built on top of the block device (2)



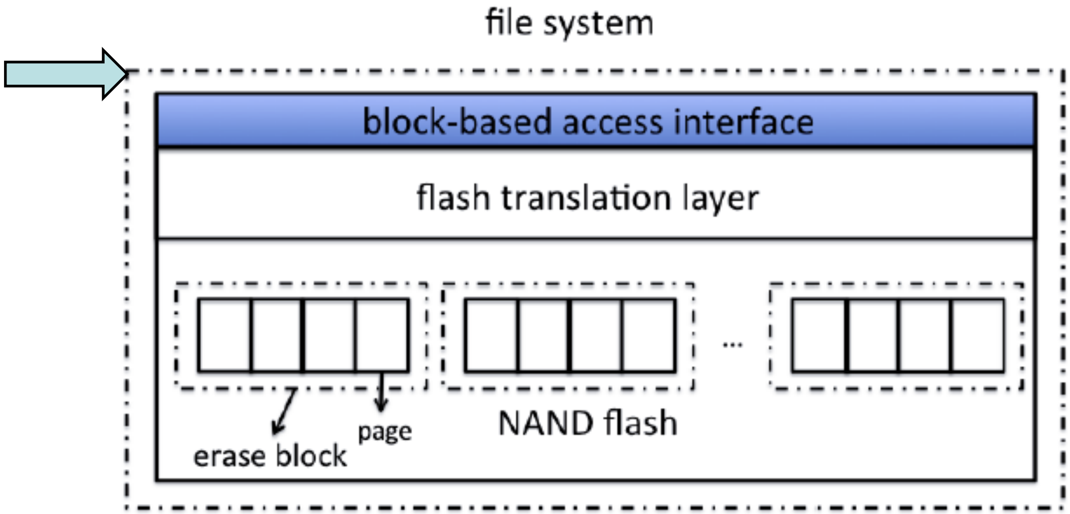
Block i and block j have duplicate randomness

Special flash memory operations like wear leveling and garbage collection on the hidden volume will create duplicate randomness

Refer to our paper published in CCS ’17 for more compromises

Fundamental Reasons for Compromises of The Existing Block-based PDE Systems

- Built on top of block device (outside the black box of flash memory), and cannot manage the internal flash memory
- Unexpected ``traces'' of hidden sensitive data could be created in the flash memory which is out of the control of the block-based PDE



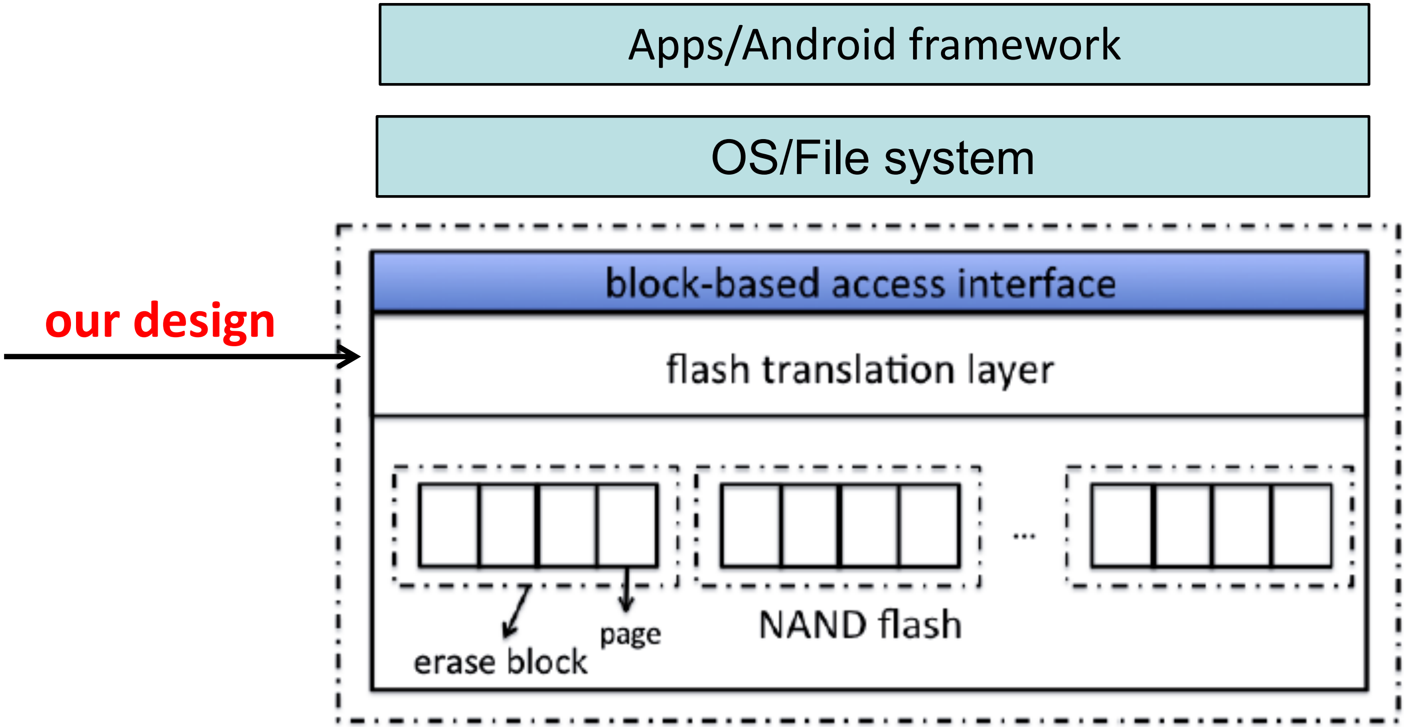
Block-based PDE systems

Existing PDE Systems for Mobile Devices (cont.)

- The sole PDE system built into the flash memory is DEFY [Peters et al., NDSS ’15]
  - Strongly rely on special properties of the flash file system YAFFS (hence not applicable to FTL, which is a dominant flash architecture)
  - Suffering from deniability compromise since it simply disables garbage collection (insecure)

Our FTL-based PDE System [CCS '17]

Key insight 1: move the public/hidden volume design down to the flash translation layer (FTL).

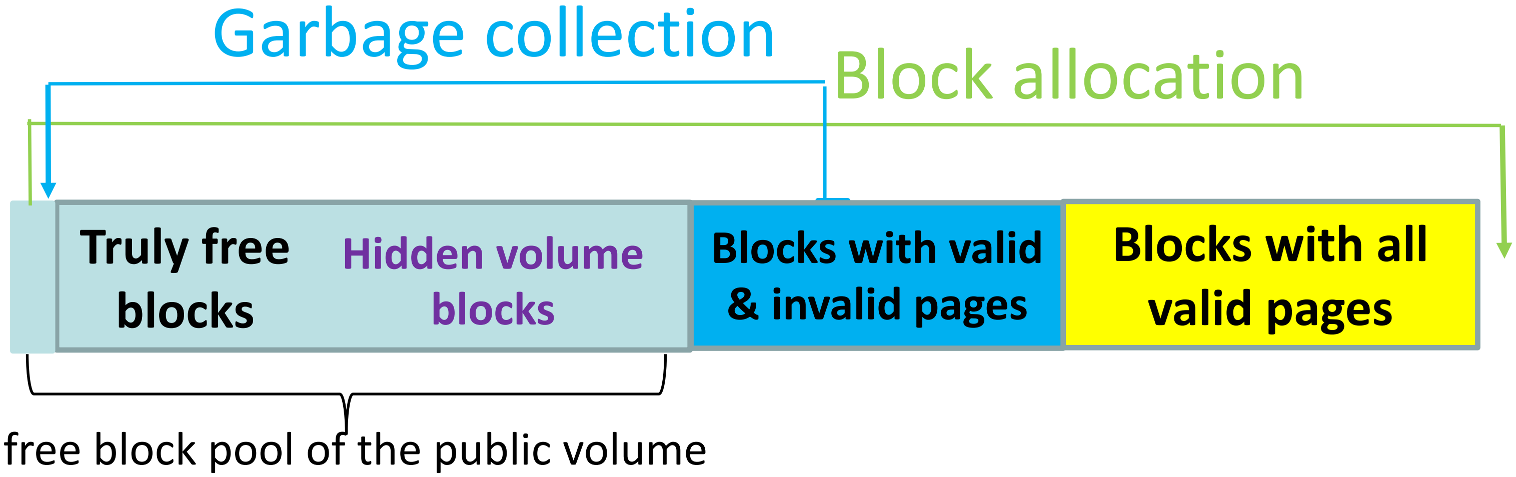


Shijie Jia, Luning Xia**, Bo Chen**, and Peng Liu. DEFTL: Implementing Plausibly Deniable Encryption in Flash Translation Layer. 2017 ACM Conference on Computer and Communications Security (CCS '17), Dallas, Texas, USA, Oct 30 - Nov 3, 2017 (Acceptance rate: 18%)

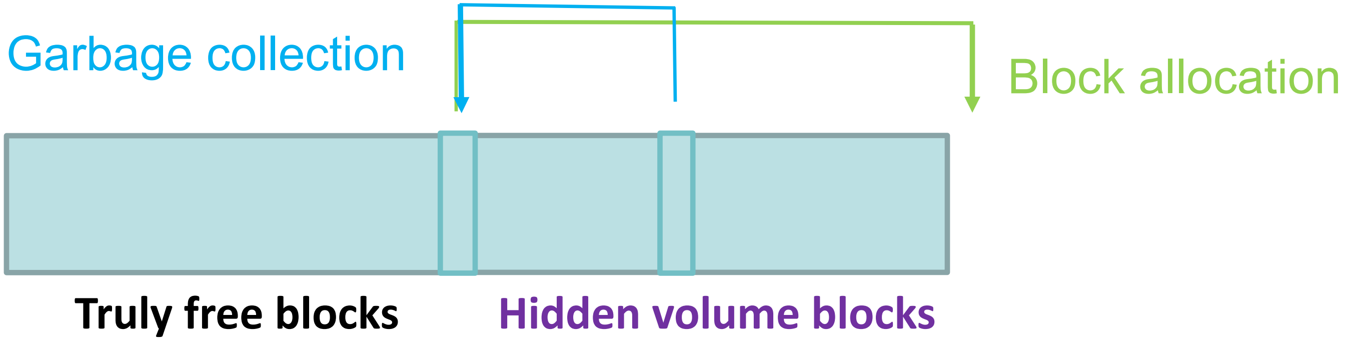
Our FTL-based PDE System (cont.)

Key insight 2: to mitigate the over-write issue:

1) The public volume will allocate blocks from the head of the free block pool; active garbage collection will be performed to fill the head of the free block pool.


Garbage collection
Block allocation
Truly free blocks
Hidden volume blocks
Blocks with valid & invalid pages
Blocks with all valid pages
free block pool of the public volume

2) The hidden volume will allocate blocks from the tail of the truly free blocks; active garbage collection will be performed to fill the tail of the truly free blocks.



Acknowledgments

- The PDE project has been supported by US National Science Foundation under grant number 1928349-CNS

- Don’t forget to send your grade for today’s presenters