

PV204 Security technologies



File and disk encryption

Milan Brož xbroz@fi.muni.cz
Petr Švenda svenda@fi.muni.cz
Faculty of Informatics, Masaryk University



Data storage encryption

- Lecture
 - File and disk encryption
 - Distributed storage encryption
 - Abstraction layers, hardware acceleration
 - Cryptography basic principles
 - Confidentiality and integrity protection
 - Encryption modes
 - Key management
 - Tool examples (Windows, Linux, Android)
 - Attacks and common issues
- Lab – disk encryption attack examples

File and disk encryption

MOTIVATION & STORAGE LAYERS OVERVIEW

Motivation

Offline, "Data at Rest" protection

notebook, external drives, data in cloud, backups

Key removal = easy data disposal

Confidentiality protection

company policy to encrypt all mobile devices
prevents data leaks (stolen device)

Integrity protection (not often yet)

Overview

(Distributed) Storage Stack

layers accessing storage through blocks (sectors)
near future: non-volatile byte-addressable memory
distributed => adding network layer

Full Disk Encryption (FDE)

self-encrypted drives
(software) sector-level encryption

Filesystem-level encryption

general-purpose filesystem with encryption
cryptographic file systems

Storage stack & encryption layers

Userspace	Application	(Application specific)
OS kernel	Virtual file-system (directories, files, ...)	File-system encryption
	Specific file-system (NTFS, ext4, XFS, ...)	
	Volume Management (partitions, on-demand allocation, snapshots, deduplication, ...)	Disk encryption
	Block layer (sectors I/O)	
	Storage transport (USB, SCSI, SAS, SATA, FC, ...)	HW-based disk encryption (self-encrypted drives, chipset-based encryption)
	Device drivers	
“Hardware”	Hardware (I/O controllers, disks, ...)	

Clustered and distributed storage

Clustered => cooperating nodes

Distributed => storage + network

Software Defined Storage/Network (SDS, SDN)

- commodity hardware with abstracted storage/network logic
- encryption is “just” one logic function
- usually combination with classic storage (and encryption)

Distributed storage & encryption

Shared volumes (redundancy)

=> disk encryption

Clustered file-system

=> file-system encryption (in theory)

Distributed Object Store

- Direct object encryption (in theory)
- Underlying storage encryption (FDE)

Cloud storage & encryption

Many users with shared storage backend
Compression & Deduplication & Snapshots ...

Encryption on client side (end-to-end)

efficiency for deduplication/compression lost
~ homomorphic encryption?

Encryption on server side

confidentiality for clients partially lost
(server has access to plaintext)

Full Disk Encryption (FDE)

Block device – transparent disk sector level

- Disk, partition, VM disk image
- Ciphertext device / virtual plaintext device
- Atomic unit is sector (512 bytes, 4k, 64k)
- Consecutive sector number
- Sectors encrypted independently

One key decrypts the whole device

- Media (volume) key – one per device
- Unlocking passphrases/keys
- Usually no integrity support (only confidentiality)

Filesystem-level Encryption

File/Directory

- Atomic unit is filesystem block
- Blocks encrypted independently
- **Generic filesystems with encryption**
 - Some metadata can be kept in plaintext (name, size, ...)
- **Cryptographic filesystems**
 - Metadata encrypted
 - ~ stacked layer over generic filesystem

Multiple keys / multiple users

File vs. disk encryption

Full disk encryption

- + for notebook, external drives (offline protection)
- + transparent for filesystem
- + no user decision later what to encrypt
- + hibernation partition and swap encryption
- more users – whole disk accessible
- key disclosure – complete data leak
- usually no integrity protection

File vs. disk encryption

Filesystem based encryption

- + multiple users
- +/- user can decide what to encrypt
- + copied files keeps encryption in-place
- + more effective (only really used blocks)
- + should provide integrity protection (not always!)
- more complicated sw, usually more bugs
- unusable for swap partitions

File vs. disk encryption

Combination of disk & file encryption

Distributed storage

- **Must** use also network layer encryption
- Difference in network and storage encryption (reply attack resistance, integrity protection, ...)

File and disk encryption

CRYPTOGRAPHY

Cryptography algorithms primitives

Symmetric encryption

- block ciphers

- cipher block mode

- hash algorithms

Key management

- Random Number Generators (RNG)

- Key Derivation Functions (KDF)

- Asymmetric cryptography

Deniable encryption / Steganography

Data confidentiality & integrity

Confidentiality

Data are available only to authorized users.

Integrity

Data are consistent and has not been modified by unauthorized user.

(And all modifications must be detected.)

Note: reply attack (revert to old snapshot) detection cannot be provided without separate trusted store.

Data integrity / authenticated encryption

Poor man's authentication (= no authentication)

- User is able to detect unexpected change
- Very limited, cannot prevent old content replacement

Integrity – additional overhead

- Where to store integrity data?
- Encryption + separate integrity data
- Authenticated modes (combines both)

File and disk encryption

DATA ENCRYPTION, ENCRYPTION MODES

Symmetric encryption (examples)

AES, Serpent, Twofish, ...

Encryption-only modes

- Storage encryption mostly CBC, XTS
- Length-preserving encryption, block tweak

Authenticated modes (encryption + integrity)

- Integrity protection often on higher layer.

Storage standards IEEE 1619 or FIPS/NIST

Propagation of plaintext changes

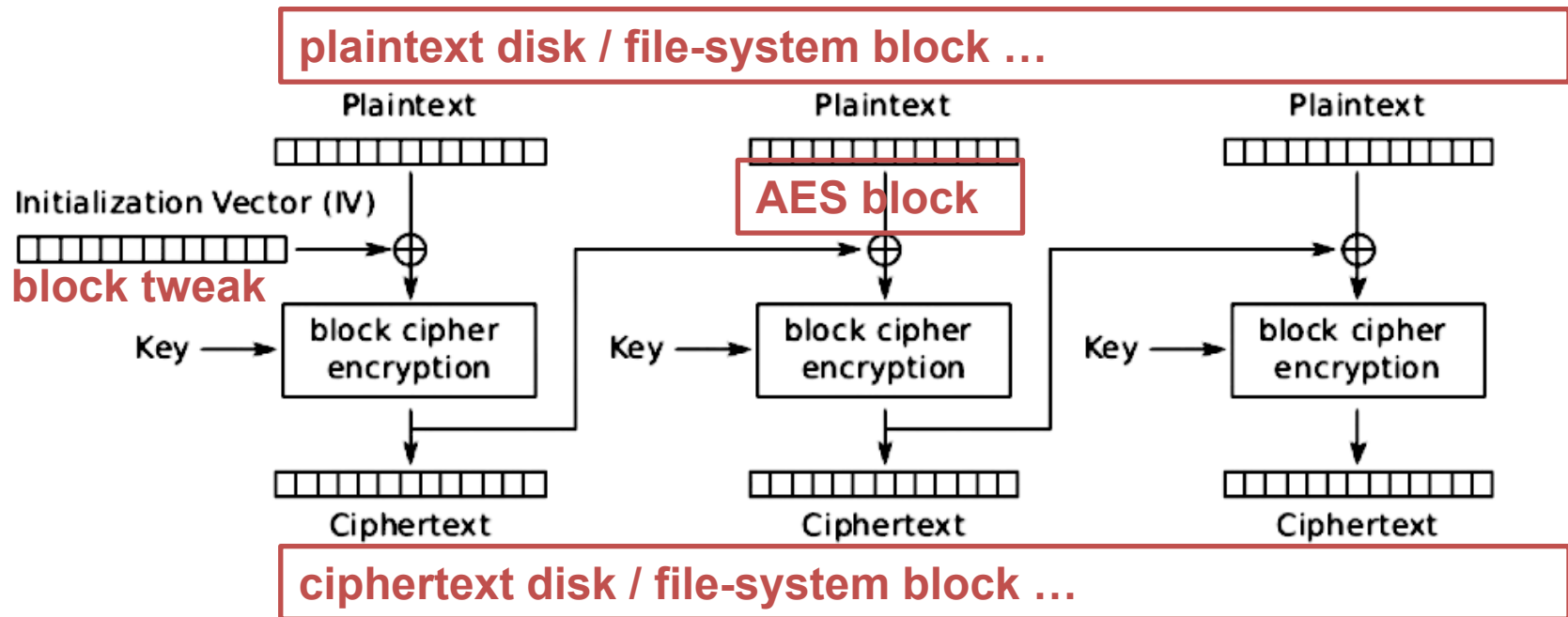
Arbitrary change in plaintext sector should transform to randomly-looking change in whole ciphertext sector.

Solutions:

- Ignore it 😊 and decrease granularity of change
=> change location inside ciphertext sector
- Use wide mode (encryption block size = sector size)
 - requires at least 2x encryption loop
 - modes are patent encumbered
- Additional operations
 - Elephant diffuser in Windows Bitlocker

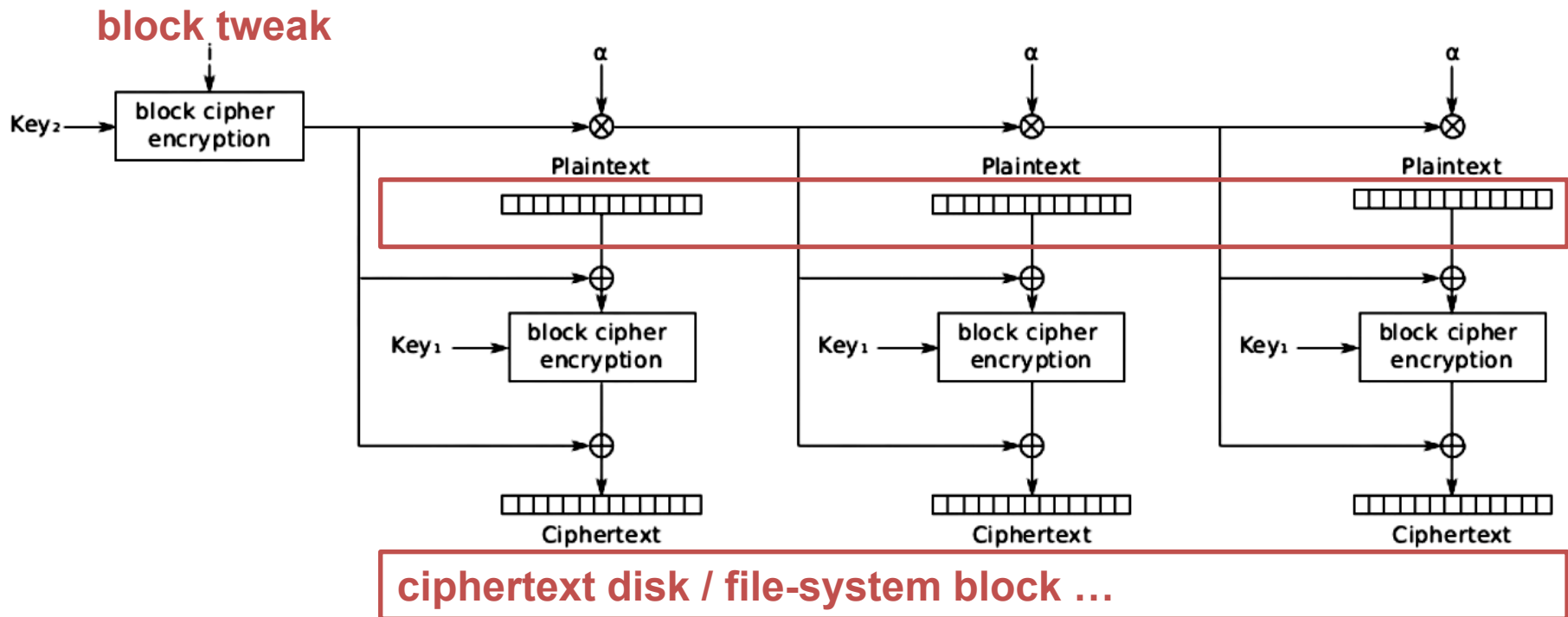
Cipher-Block-Chaining (CBC) mode

- Blocks cannot be encrypted in parallel
- Blocks can be decrypted in parallel
- Tweak must be non-predictable (watermarking!)



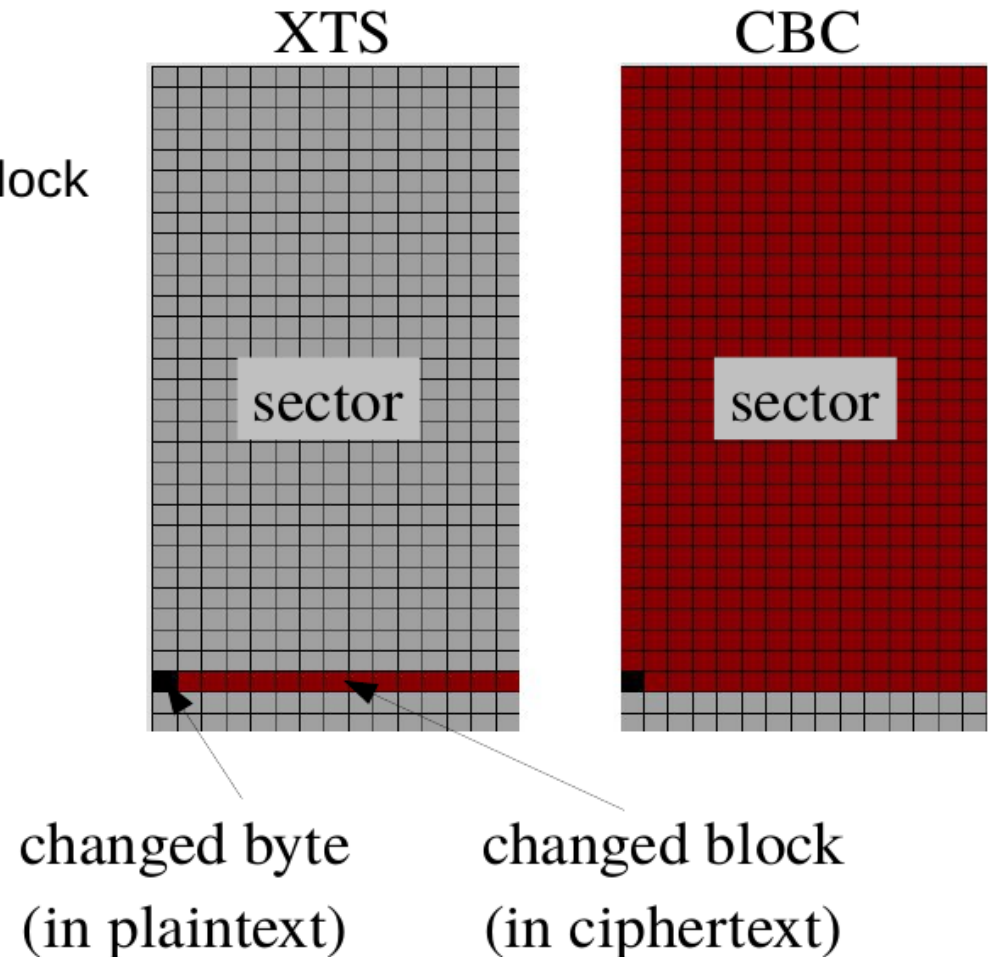
XOR-Encrypt-XOR (XEX/XTS) mode

- Encryption/decryption can be run in parallel
- Tweak can be predictable nonce (sector offset)



CBC and XTS change propagation

- **CBC** – cipher block chaining
 - ciphertext XOR with next block
- **XTS / XEX** (XOR encrypt XOR)
 - internally 2 keys
 - key for tweak
 - encryption key



Steganography / deniable encryption

Plausible deniability:

existence of encrypted file/disk is deniable
if adversary cannot prove that it exists

Steganography

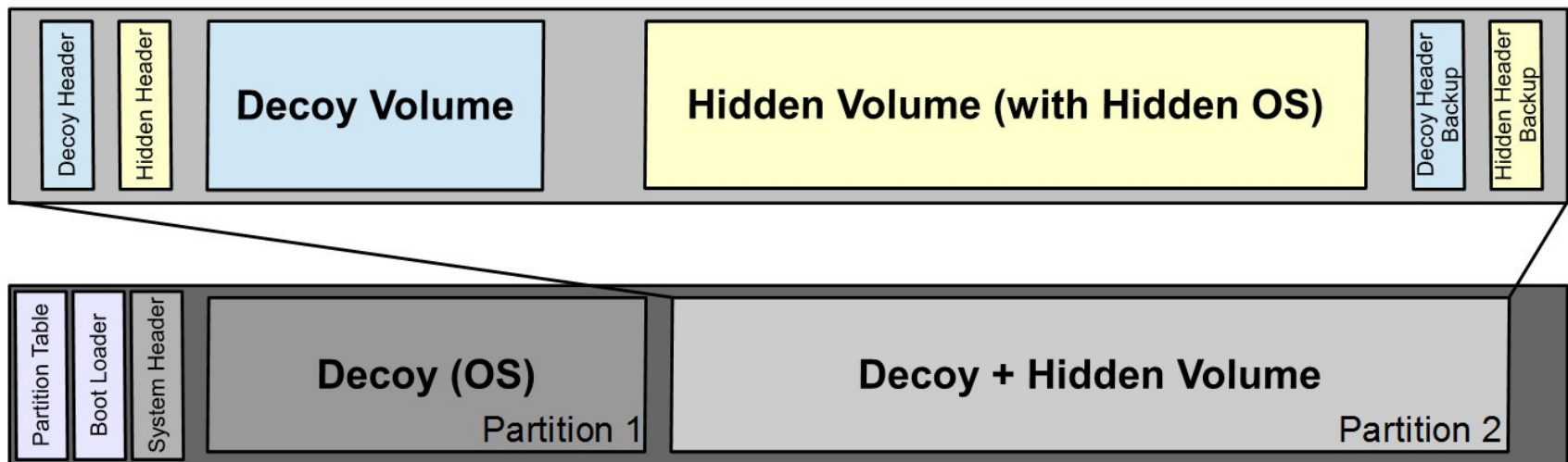
hiding data in another data object

Steganographic file-systems

Deniable disk encryption

Trivial example: TrueCrypt hidden disk

- FAT linear allocation
- Hide another disk in unallocated space



Deniable encryption problems

Side-channels

tracking activity that cannot be explained for decoy system

- Software: link to recently open documents, ...
Suspicious parameters (FAT), disabled TRIM, ...
- Hardware: internal SSD block allocations
(access to “unused” areas)

Social engineering / “rubber hose” analysis

Incompatibility with new drives (TRIM)

File and disk encryption

KEY MANAGEMENT

Key generation

Encryption key (~ Media Encryption Key – MEK)

- Used to encrypt device
 - change means complete reencryption
- Usually generated by secure RNG

Unlocking key (~ Key Encryption Key – KEK)

- Used to unlock key store, token, ...
- Independent key change (MEK remains the same)
- Usually derived from passphrase
 - PBKDF2 (Password Based Key Derivation)

Key storage

Outside of encrypted device / filesystem

- Another device, file, token, SmartCard, TPM
- On a key server (network)
- Protected by another key (KEK).

On the same disk (with encrypted data)

- metadata (header)
- brute force and dictionary attack resistance

Integration with key management tools

- LDAP, Active Directory, ...

Key removal and recovery

Key removal (wipe of key) = data disposal

- intended (secure disk disposal)
- unintended (error) => complete lost of data

Key recovery

- Trade-off between security and user-friendly approach
- Metadata backups
- Multiple metadata copies
- Key Escrow (key backup to different system)
- Recovery key to regenerate encryption key

File and disk encryption

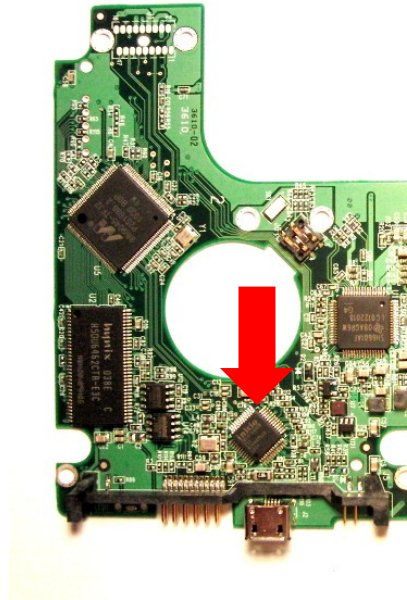
COMMON TOOLS

Examples of HW-based encryption

- **Self-encrypting drives (SED)**
 - Encryption on the same chip providing media access
- **Chipset-based encryption**
 - Encryption on controller chip (e.g. USB bridge)
- **Hardware acceleration**
 - AES-NI, accelerators, ASICs, GPUs, ...
- **Secure hardware / tokens**
 - HSM, TPM, SmartCards, ...

Examples of HW-based encryption

SATA disk
Encryption on USB-bridge



Examples of tools – filesystem encryption

Windows EFS

Linux eCryptfs

POSIX-compliant stacked encrypted file-system

ZFS (Solaris and ports)

supports GCM/CCM authenticated modes

Examples of tools – full disk encryption

Windows Bitlocker

Optionally eDrive – self-encrypted drives
Combination with secure boot

TrueCrypt / VeraCrypt / CipherShed

Linux LUKS / dm-crypt

Linux dm-crypt / dm-verity

used for Android encryption / verified boot

MacOS FileVault

File and disk encryption

ATTACKS EXAMPLES

Attacks always get better, they never get worse.

- **Against algorithm design**
 - Wrongly used encryption mode
 - Insufficient initialization vector
- **To implementation**
 - Insufficient entropy (broken RNG)
 - Weak derivation from weak passwords
 - Side channels
- **Obtaining key or passphrase in open form**
 - Cold Boot
 - “Black bag analysis” - Malware, key-logger
 - Social engineering
 - “Rubber-hose cryptoanalysis”

Integrity attacks

No integrity protection

- Inserted random block
=> undected data corruption
- Inserted block from other part of disk
- Random error (RAM bit flip)
=> “silent data corruption”

Weak integrity protection

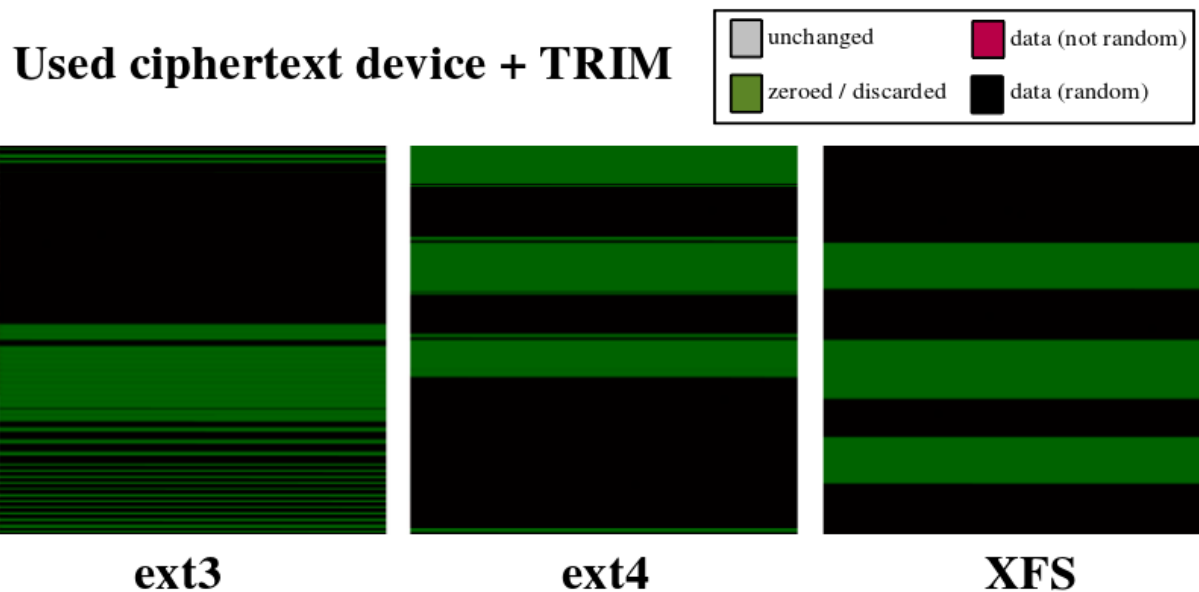
- Inserted previous content of (ciphertext) block
=> replay attack

Example: Sony PlayStation attack

- No need to know the exact key value
- Device works as decryption service
- Make disk image
- Write your own file
- Make second image and make diff (location file)
- Insert data you want into file's place
- Start PS and ask for your file
 - Decrypted new data
 - Key is same for all blocks

TRIM / discard and encryption

- TRIM informs SSD drive about unused space
- Unused space is detectable
- Pattern recognition example
- Incompatible with deniable encryption



LAB



Laboratory – FDE attack examples

Basic understanding of some tools

VeraCrypt, LUKS

Scanning memory image for encryption key

ColdBoot attack principle

Advanced: flawed algorithm and watermarking

Revealing TrueCrypt hidden disk existence (CBC)

□

HW key-logger attack

Homework assignment (bonus)

- **Analyse pv204_assignment.tc**
 - TrueCrypt volume compatible with TrueCrypt 7.1a and VeraCrypt 1.17
 - The volume is protected by a 9-character long password, which begins with "pv204_XXX" where X means digit [0-9].
- Find the password and unlock the volume.
- Investigate encryption keys and header salt.
- Describe found problems (max one A4 doc)
- **Please read notes in assignment archive!**
- Submit before: 20.5. 6am (full number of points)
 - Every additional started day (24h) means 1.5 points penalization