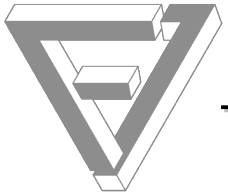
Centre for Research on
Cryptography and Security

Disk encryption ...



Milan Brož
xbroz@fi.muni.cz

ESET

October 30, 2014

FDE - Full Disk Encryption

- Transparent encryption on disk level (sectors)

  FDE (Full Disk Encryption)

  FVE (Full Volume Encryption)

  ...

+ offline protection for notebook, external drives

+ transparent for filesystem

+ no user decision later what to encrypt

+ encryption of hibernation and swap file

+ key removal = easy data disposal

- more users, disk accessible to all of them

- key disclosure, complete data leak

- sometimes performance problems

FDE – layer selection

SW

- **userspace**
  (not directly in OS kernel)

  (In Linux usually FUSE handled)

- **sw driver**
  - low-level storage/SCSI
  - virtual device

  dm-crypt, TrueCrypt, DiskCryptor, Bitlocker, ...
  AES-NI, special chips (mobile), TPM (Trusted Platform Module)

HW

- **driver + hw**
  (hw acceleration)

- **disk controller**

  Chipset FDE (... external enclosure)
  Many external USB drives with "full hw encryption"

- **on-disk**

  HDD FDE
  special drives

  Hitachi: BDE (Bulk Data Encryption)
  Seagate: FDE (Full Disk Encryption)
  Toshiba: SED (Self-Encrypting Drives)
  ...

Chipset or on-disk FDE

- Encryption algorithm is usually AES-128/256
  ... encryption mode is often not specified clearly

- Where is the key stored?
  - proprietary formats and protocols
  - randomly generated?

- External enclosure
  - Encryption often present (even if password is empty)
    => cannot access data without enclosure HW board
  - Recovery: you need the same board / firmware
  - Weak hw part: connectors

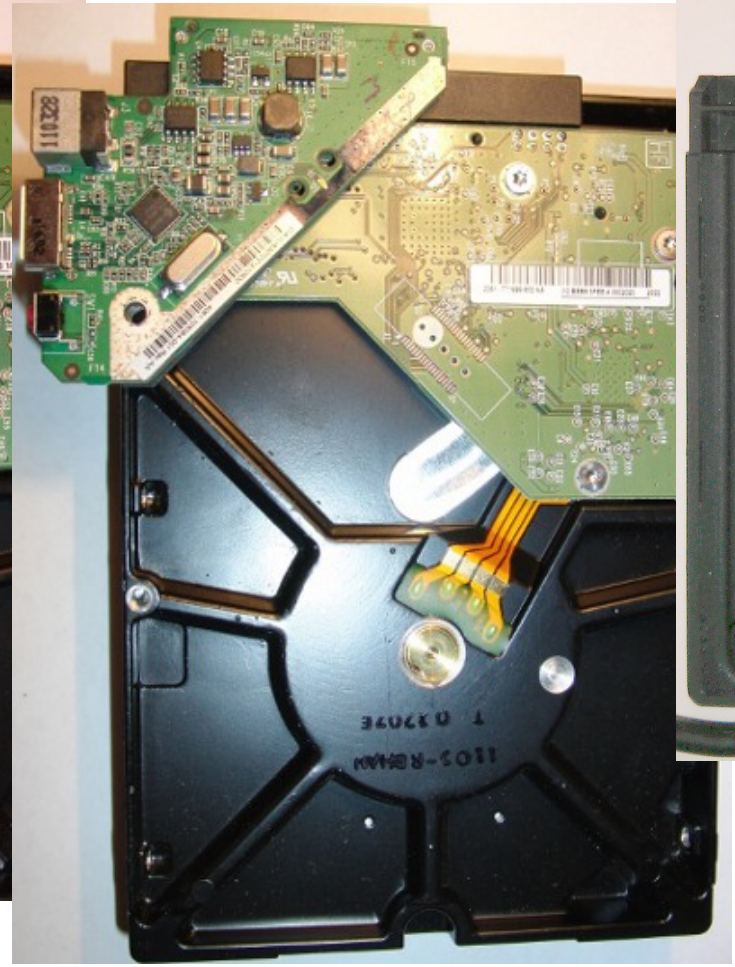
- Full system configuration
  - laptop with FDE support in BIOS / trusted boot

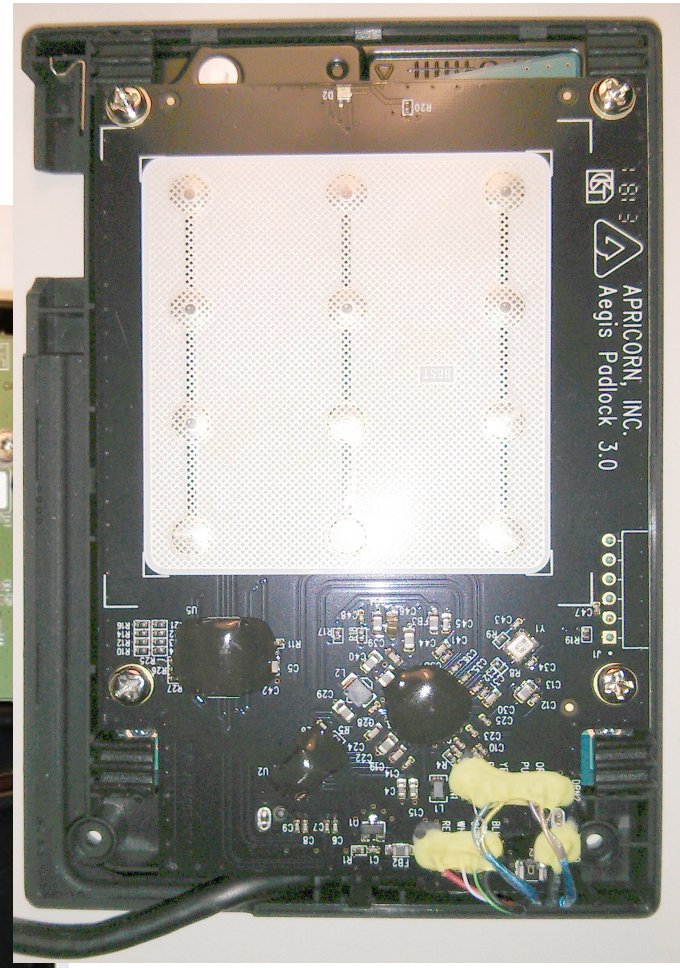
- Vendor lock-in

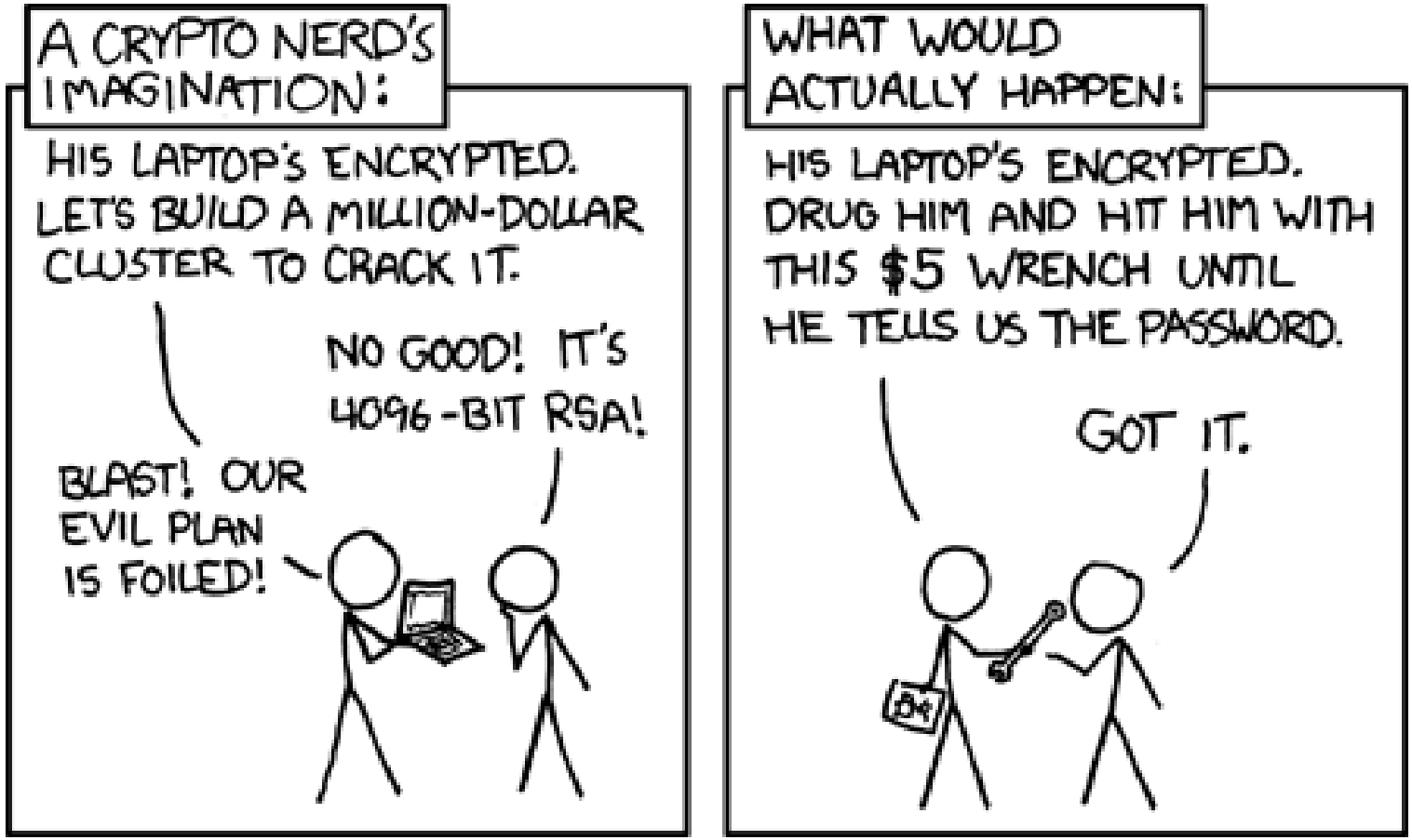
- FIPS 140-2 (mandatory US standard, not complete evaluation of security!)

Chipset FDE examples

- **Encryption on external disk controller board + standard SATA drive**
  - USB3 external disk enclosure
  - AES-256 encryption





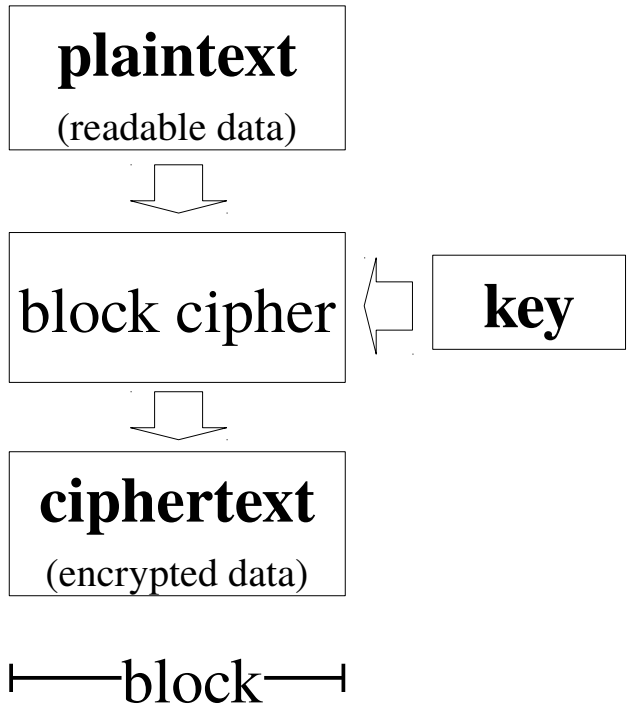


http://imgs.xkcd.com/comics/security.png

Disk encryption ... definitions & theory ___

Plaintext & ciphertext

- **Plaintext – (open) data**
- **Ciphertext – encrypted data**
- **Symmetric algorithms**
  - secret key
  - AES, Twofish, Serpent, ...



Block device, sector

- **Block device**
  disk, partition, virtual block device

- **Sector – block device atomic unit**
  512 bytes, 4096 bytes or 64 kilobytes

- To avoid confusion, let's use
  sector = disk unit (... typically 512 bytes)
  block = encryption block (... typically 16 bytes)

- **Disk encryption:**
  ciphertext block device -> plaintext (open) block device
    - of the same size
    - no space for integrity or sector authentication data

  - without key, ciphertext (device) looks like random data

Block mode vs sector

Goal:
Arbitrary change (plaintext) – **change of the whole sector** (ciphertext)

- **Wide block mode** (block size = sector size)
  - requires at least 2x encryption loop
  - patent encumbered (~ free EME-2)
  - almost not used in reality

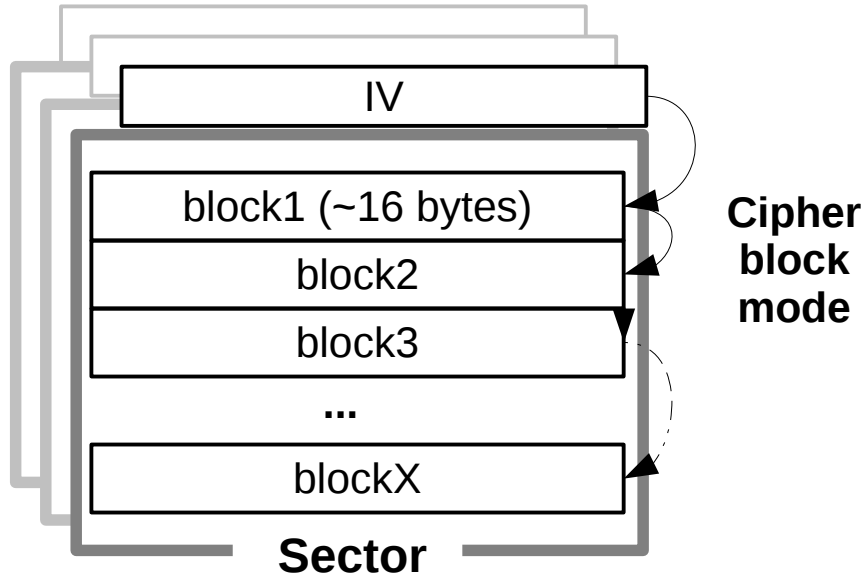
- **Narrow block mode**
  - trade-off, not the whole sector change
  - CBC: Cipher Block Chaining
  - XTS: XEX-based tweaked mode (with ciphertext stealing)

- **Additional operations**
  Elephant diffuser in Bitlocker
    - additional mixing / rotation operation (keyed, not cryptographic)
    - tweak key (independent of encryption key)

Cipher block mode, Initialization vector

- **BLOCK** (of cipher) **< SECTOR** (of disk)

  sector contains several blocks
  chained or parallel processing



- **Narrow Block Mode**

- **Initialization vector IV** (tweak)
  different for every sector
  => same data in different sectors – different ciphertext

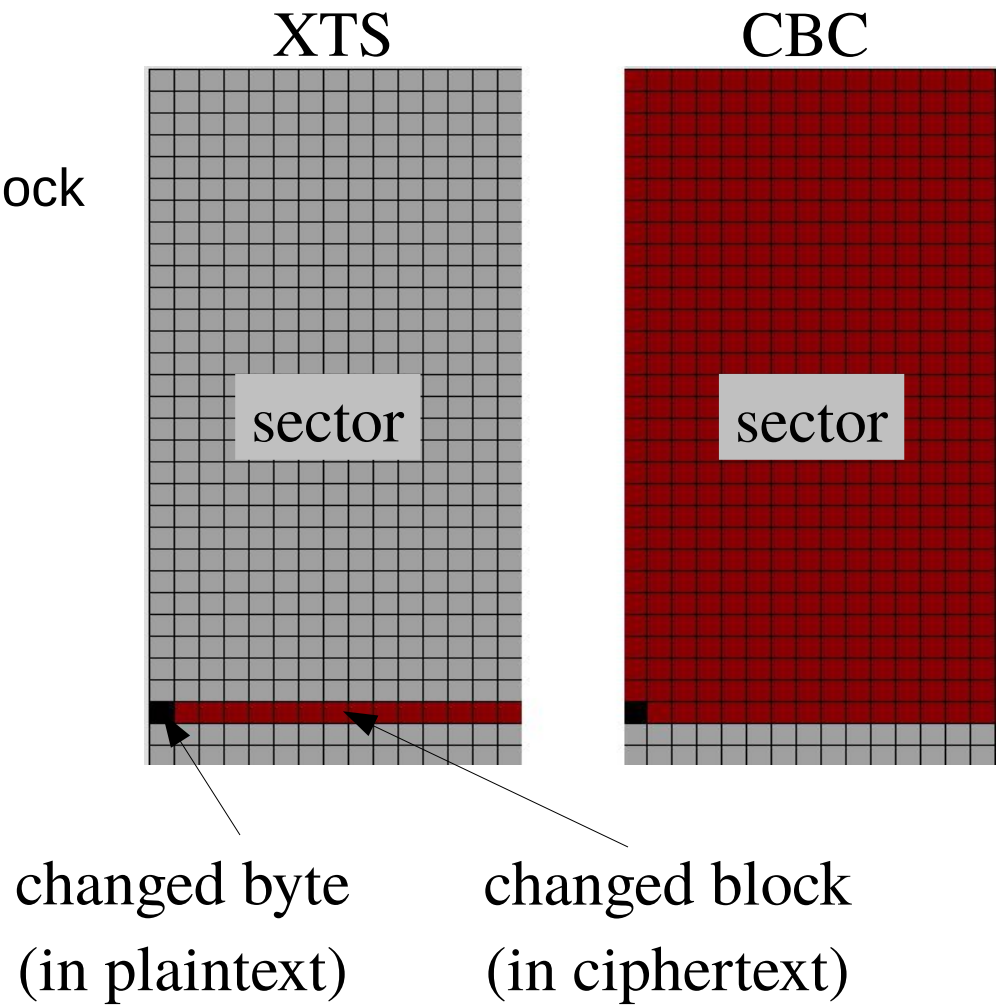
  - usually derived from sector number (and key, if needed)
    e.g. ESSIV – Encrypted Salt-Sector IV

  - XTS second encryption key -> IV is directly sector number
    (known to attacker)

(Narrow) cipher block mode – examples

How a change in plaintext changes ciphertext?

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



Authenticated encryption

- Will detect ciphertext sector change

- Requires additional space
  - for integrity data
  - for authenticated tag

- Authenticated modes (like GCM)

- Integrity usually on higher level (filesystem)
  - used e.g. in ZFS storage system

- Should detect "silent data corruption" early
  - error propagation
  - impact of single bit error with FDE is worse

GELI authenticated mode

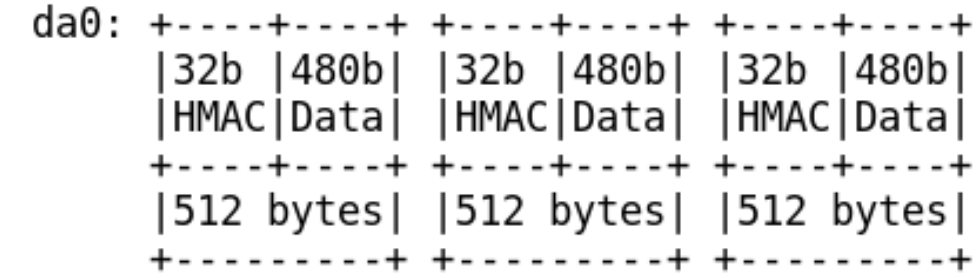
FreeBSD disk encryption system (by P.J.Dawidek)

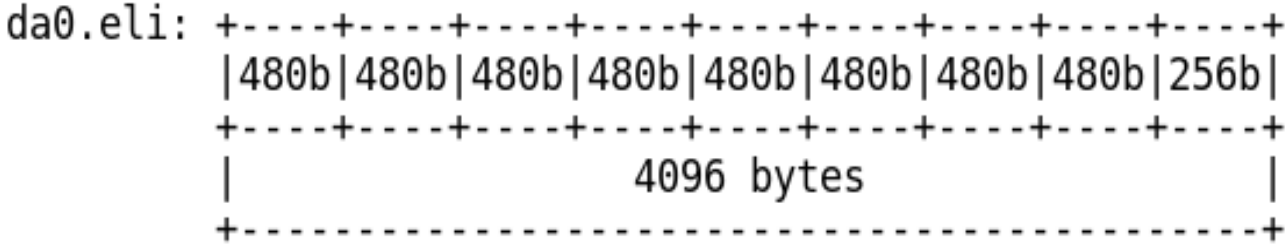
- Authenticated encryption mode

  http://svnweb.freebsd.org/base/head/sys/geom/eli/g_eli_integrity.c?view=co

  - larger sector for plaintext device
  - smaller for ciphertext + auth tag

- Example (sector: 512 bytes -> 4k)



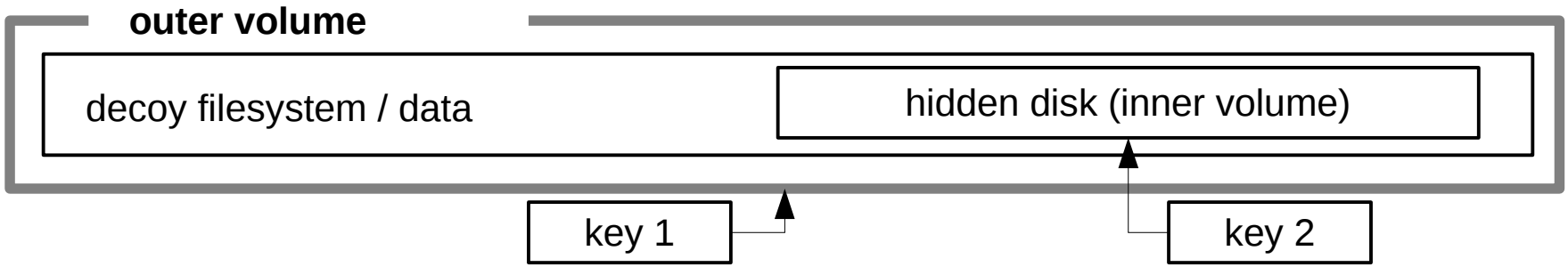


Hidden disk concept

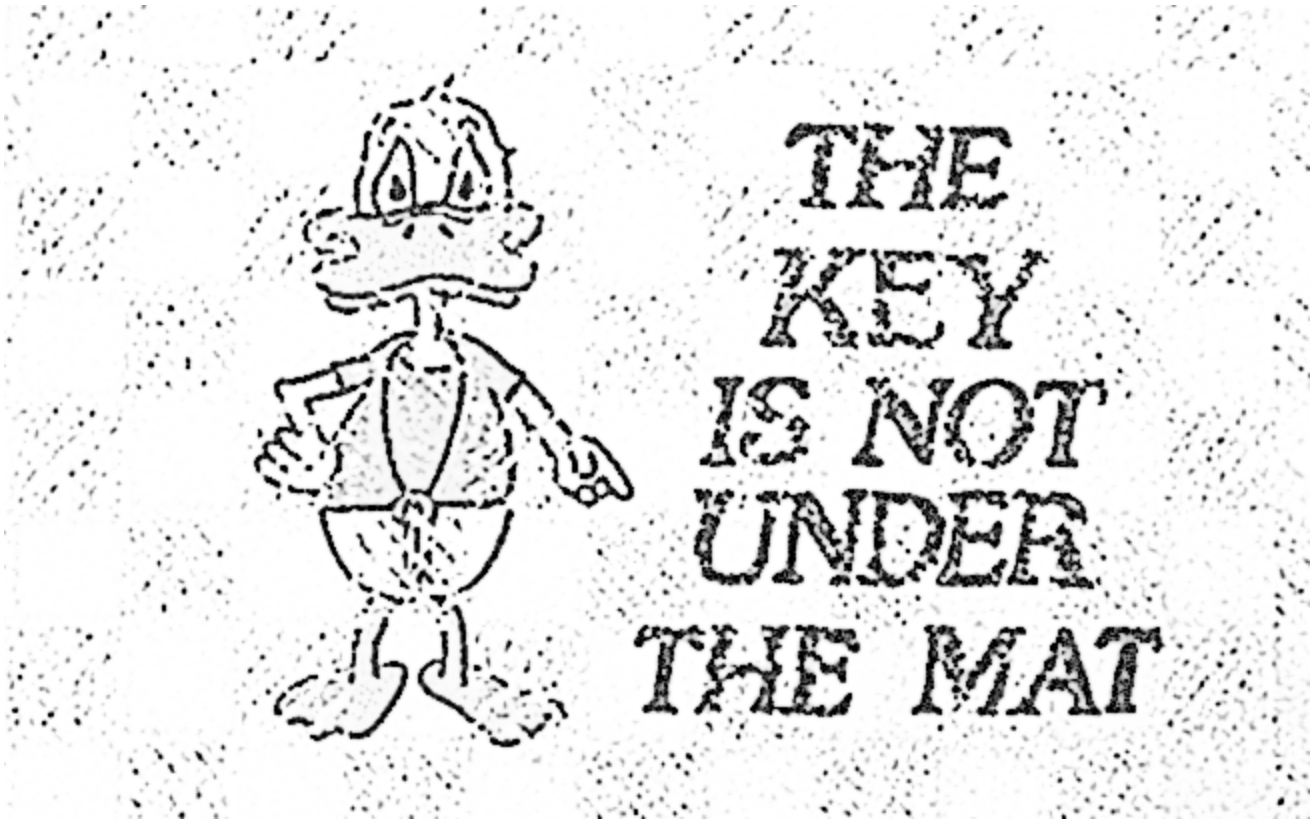
- **Plausible Deniability**
  - you can reveal the encryption key
  - real data remain secure (encrypted by another key)

- data hidden in unused (encrypted) filesystem space
  (simple steganography on sector level)

- no detectable header



- Encrypted OS, Hidden OS

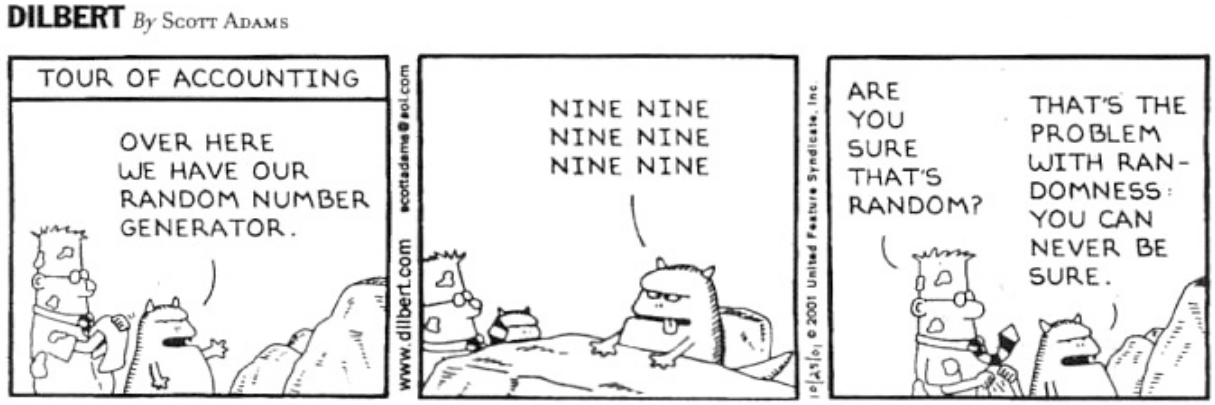


Key management ...

Key generator

- **Critical for the encryption system security**
  Note difference: encryption key / unlocking passphrase

- **Encryption key**
  random, unique
  generated by RNG (Random Number Generator)



or derived from passphrase
- i.e. PBKDF2 (Password Based Key Derivation)
- usually not desirable (~restricted in security policy)

Key storage

- **Outside of encrypted device**
  - token, SmartCard, TPM, EEPROM
  - file (protected by another encryption system)
  - (encrypted) on another disk (separation of metadata)

- **On the same disk** (with encrypted data)
  - **metadata** (header)
  - unlocking using passphrase of different key
  - brute force and dictionary attack contermeasures
    (slow down attack)
  - hw problems (e.g. firmware sector reallocation)

- **Integration with key management tools**
  - enterprise use (LDAP, Active Directory, ...)
  - dual factor authentication, TPM, ...

Key removal

- **key removal** (wipe of key storage area) **= data disposal**
  - **intended** (secure disk disposal)
  - **unintended (error)**
    - the most common problem
    - metadata overwrite – operator error
    - hw error, bad sector, controller, TPM, ...
- hidden / unintended key storage copies
  - fw reallocation
  - SSD flash translation layer (garbage collector)

(Lost) key recovery

Trade-off between security and user-friendly (availability) approach.

- disk copy (metadata)
- **Key Escrow** (key backup to diferent system)
- **duplicated metadata** on disk
- **recovery key** to regenerate encryption key

- wrongly designed user-friendly "extensions" destroys security

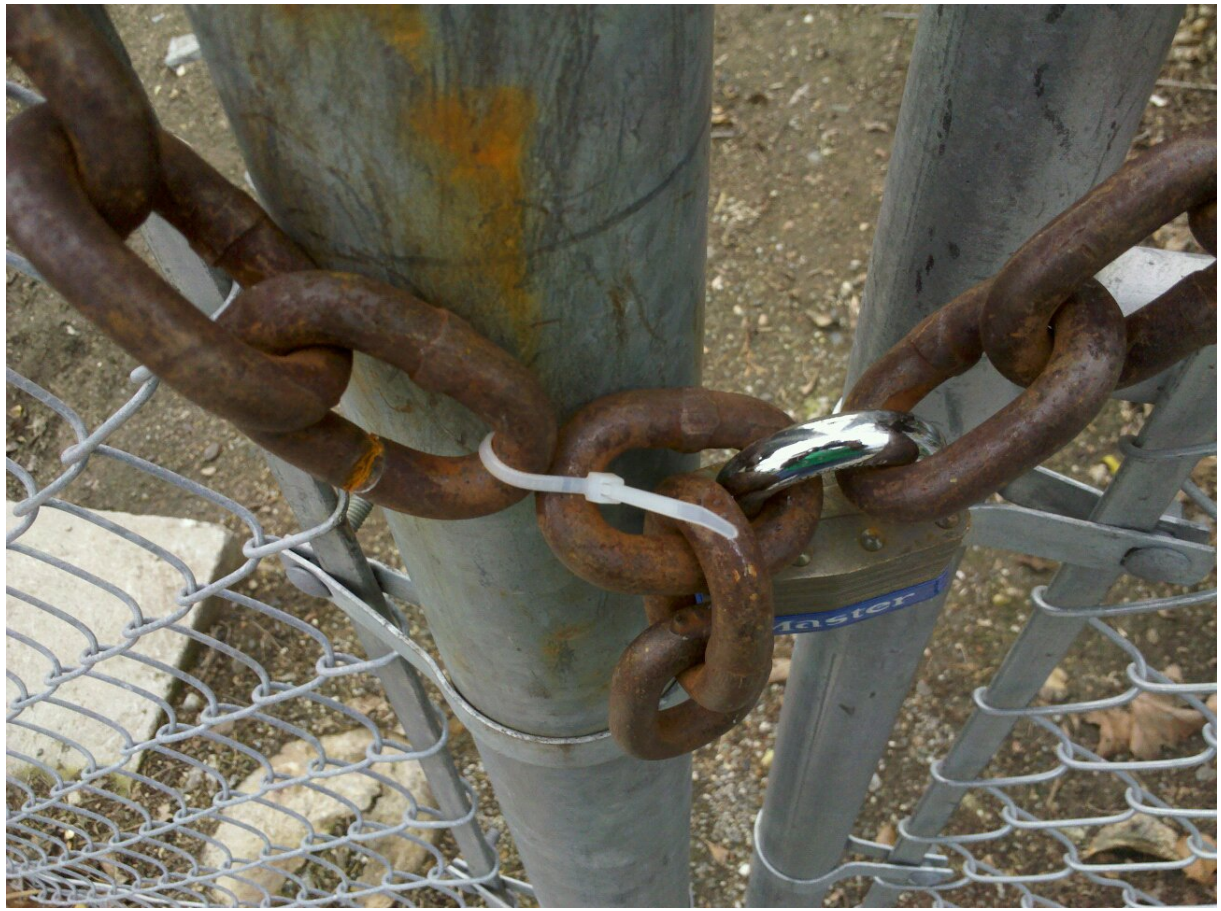
  Examples (3rd party Linux based NAS ...)

  - CVE 2009-3200 - undocumented recovery key in flash memory, allows local users decrypt the hard drive.

  CVE 2009-3278 - use the rand library to generate recovery key, brute-force attack possible.

  CVE 2008-1431 - firmware stores a partition encryption key in an unencrypted file with base64 encoding.

  ...



Attacks ...

always get better, they never get worse.

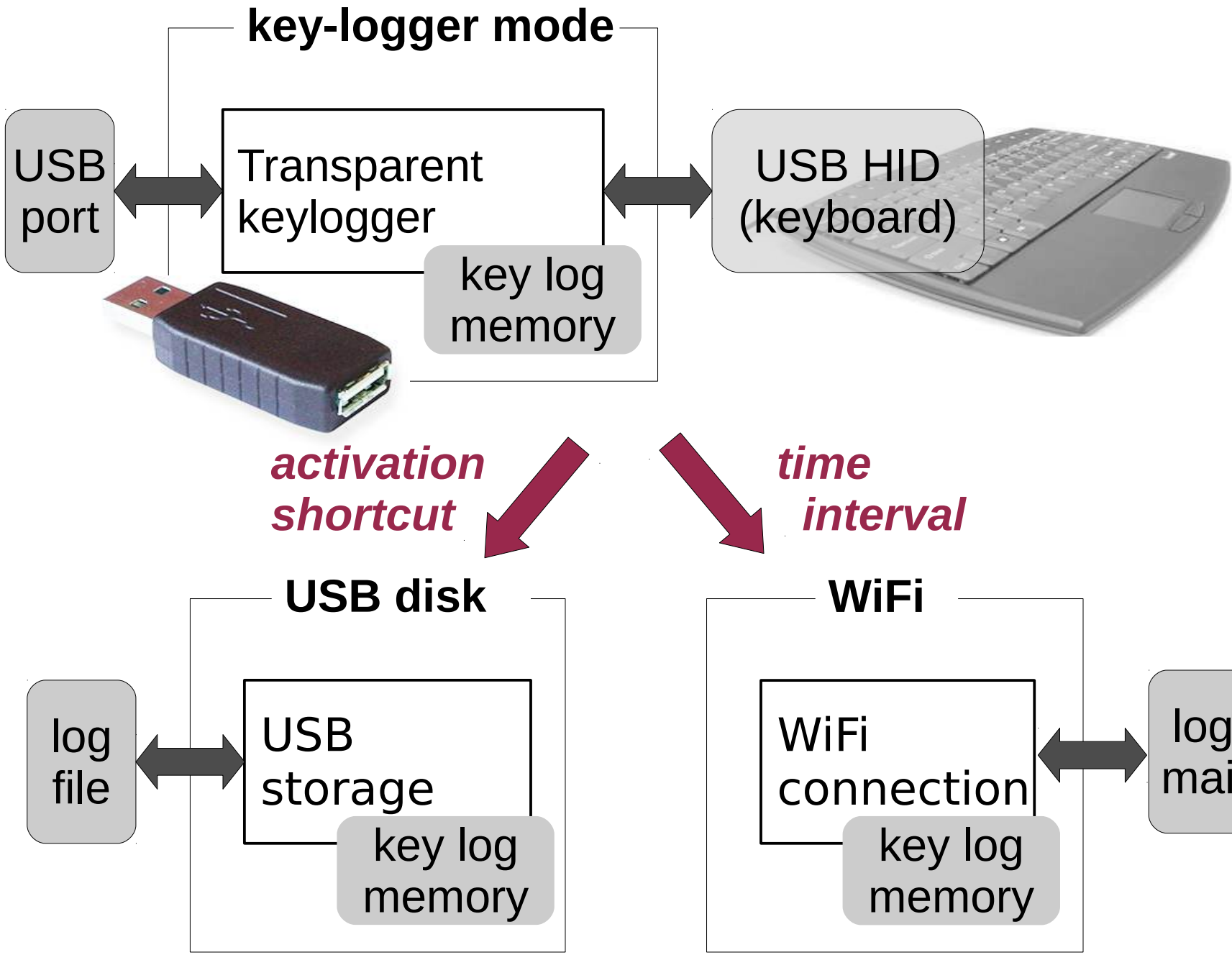
Attacks to FDE

- **Attacks to algorithm**
- **Attacks to implementation**
  e.g. side channels

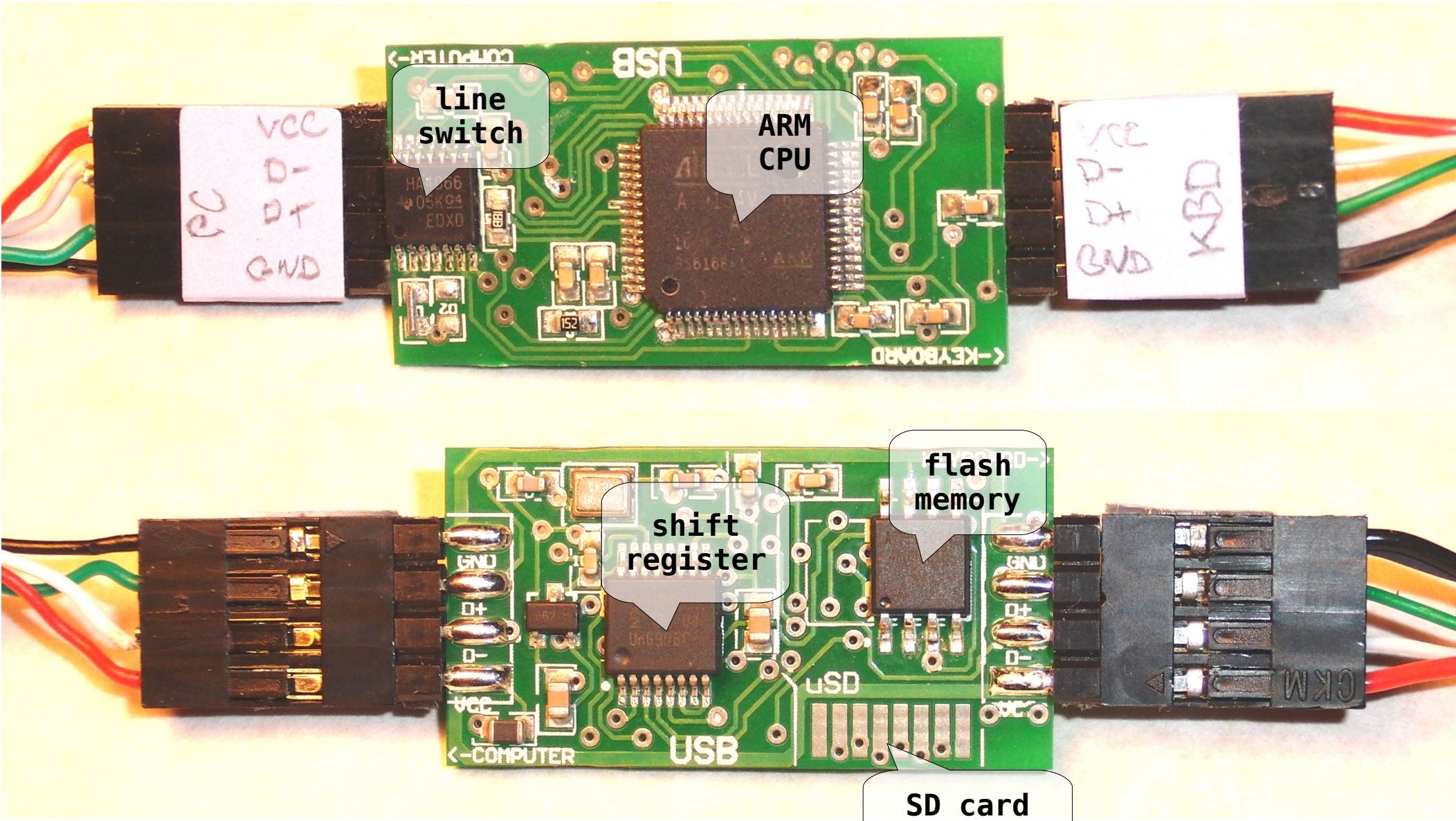
- **Obtaining key or passphrase in open form**
  - HW attack / tampering (keylogger, Cold Boot, ...)
  - Malware ("Evil Maid")
    boot / OS / hypervisor modification
  - Social engineering

If you let your machine out of your sight,
it's no longer your machine.

Simple USB hw keylogger



USB keylogger (example: KeyDaemon module, 2010)



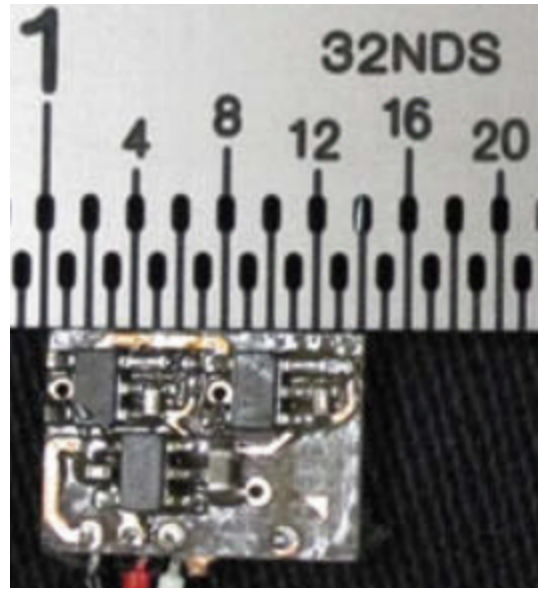
Example from NSA ANT catalog...

revealed 2013, http://en.wikipedia.org/wiki/NSA_ANT_catalog

SURLYSPAWN

• Keystroke monitor technology that can be used on remote computers that are not internet connected.

• Operates as a RF retro-reflector: returns target data modulated signal when illuminated with radar.



Cold Boot attacks

http://citp.princeton.edu/memory/
http://en.wikipedia.org/wiki/Cold_boot_attack

Lest We Remeber: Cold Boot Attacks on Encryption Keys

- Retrieving the key directly from RAM image
- DRAM content is readable seconds after power loss
  (... RAM images for VMs are readable forever ;-)
- With encryption key attacker doesn't need password
- *Physical prevention. In sw ...*
  - *CPU-based key storage (TRESOR, Loop-amnesia projects)*
    (key in special registers only, x86_64)
  - *Frozen cache*
    (key in cache, cache disabled, performance impact, x86_64)
  - *Wipe of key in RAM on suspend*

Example: Cold boot & dm-crypt/LUKS

(attacked machine) `[root@192.168.2.4]# cryptsetup luksDump /dev/sda2`

```
LUKS header information for /dev/sda2
Cipher name:    aes
Cipher mode:    xts-plain64
MK bits:        512
...

[root@192.168.2.4]# dmsetup table --showkeys
luks-...: 0 155882682 crypt aes-xts-plain64 \
ffe8b78d9f652e5eddc822885d3c2b47b3... \
75f5d220a30dbd40a506a6fdc9ad571e7b...
```

- forced machine power cycle
- boot into special PXE loader
- RAM image network copy
- scan for AES keys

(attacker machine) `$ ./pxed 192.168.2.4 >img`

```
request segment0 [base: 0x0 size: 579584]
request segment1 [base: 0x100000 size: 736034816]
request segment2 [base: 0x2bf00000 size: 1048576]

$ ./aeskey -t 50 img
ffe8b78d9f652e5eddc822885d3c2b47b3...
75f5d220a30dbd40a506a6fdc9ad571e7b...
Keyfind progress: 100%
```

Disk encryption + data channel encryption

***Example**: iSCSI exported encrypted disk, decryption on client side.*

Is there plain data on data channel? No.
*So it is secure? **No. Replay attack!***

- **FDE is offline protection** (stolen disk)
  - attacker cannot access snapshots in time
  - encryption mode designed for transparent disk access
- **Encrypted data channel**
  - Attacker can listen the whole communication
  - Attacker cannot replay data without detection.

Software based FDE ...
Examples

TrueCrypt (discontinued), CipherShed

- AES, Twofish, Serpent
- chained ciphers (e.g. AES-Twofish)
- XTS mode
  - after several iterations and vulnerable CBC implementation

- hidden disk (including disk with OS), bootloader

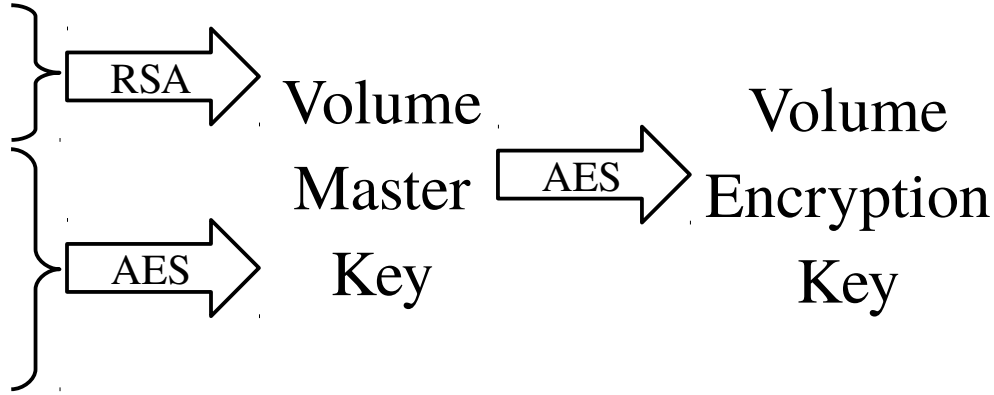
- on-disk metadata encrypted (no detectable header)
- duplicated metadata
- recovery CD
-
- CipherShed project tries to continue development

BitLocker (Windows proprietary)

Native FDE in Windows Ultimate edition, Windows 8

- many options (system policy)

- TPM
- TPM + PIN
- TPM + Startup Key
- Clear Key
- Startup/Recovery Key
- Recovery Password



- AES 128, 256 CBC
- AES 128, 256 CBC + Elephant diffuser (removed in Win 8)

- design documented, not implementations (reverse engineering)
- data, system and external devices (Bitlocker to Go)
- 3 metadata copies
- integration with MS infrastructure (keys in AD)

BitLocker (Windows)

```
C:\>manage-bde -status
BitLocker Drive Encryption: Configuration Tool version 6.1.7601
Copyright (C) Microsoft Corporation. All rights reserved.

Disk volumes that can be protected with
BitLocker Drive Encryption:
Volume C: []
[OS Volume]

    Size:                 19.71 GB
    BitLocker Version:    Windows 7
    Conversion Status:    Fully Encrypted
    Percentage Encrypted: 100%
    Encryption Method:    AES 128 with Diffuser
    Protection Status:    Protection On
    Lock Status:          Unlocked
    Identification Field: None
    Key Protectors:
        External Key
        Numerical Password
```

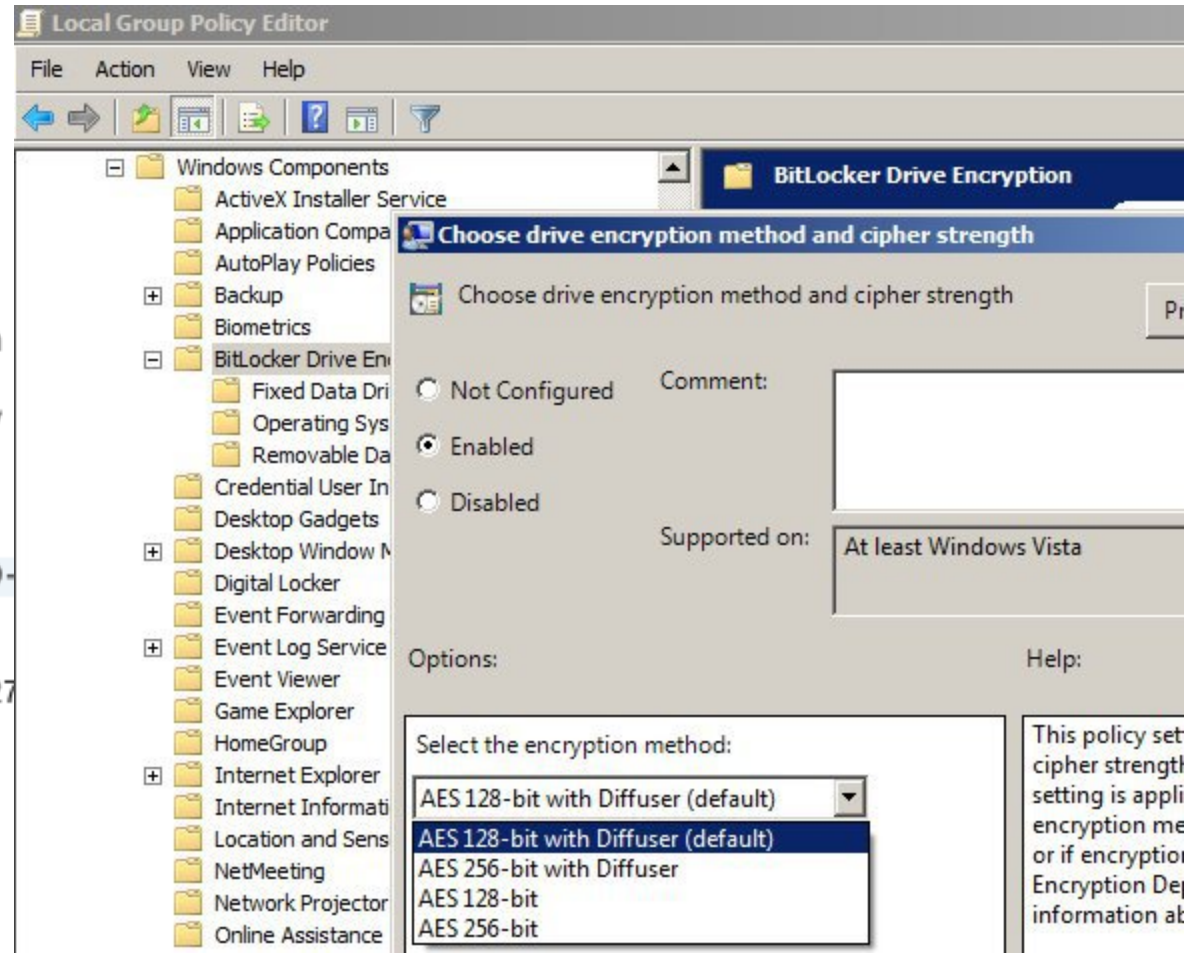
```
BitLocker Drive Encryption Recovery Key

The recovery key is used to recover the data on

To verify that this is the correct recovery key
with what is presented on the recovery screen.

Recovery key identification: B32663A4-783D-33
Full recovery key identification: B3266BA4-783D-

BitLocker Recovery Key:
728327-193815-112648-839772-108271-011233-650327
```



LUKS / dm-crypt

- **Native on Linux**
- **including support for loopAES, TrueCrypt formats**

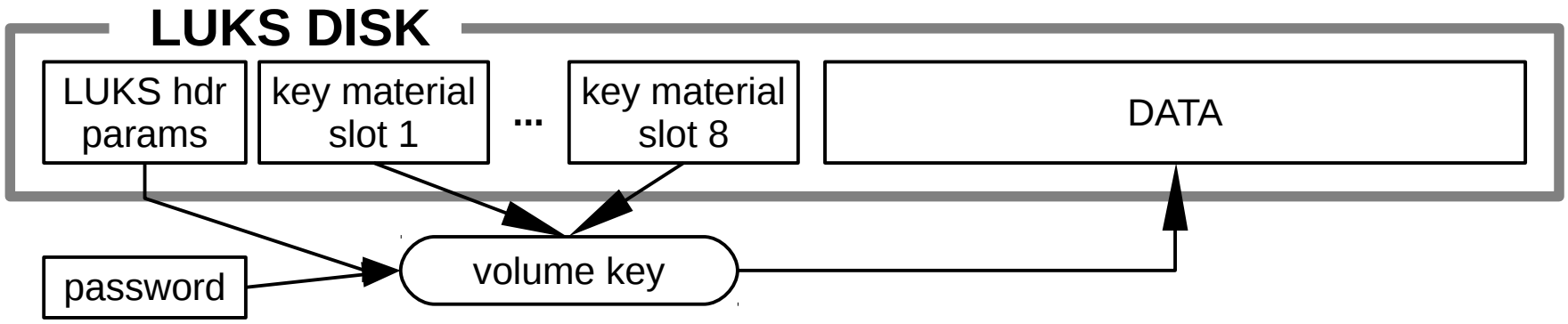
- strict separation of
  - **disk encryption engine**
    dm-crypt – device-mapper crypto target (kernel)
  - **key management (LUKS) and configuration**
    cryptsetup – userspace

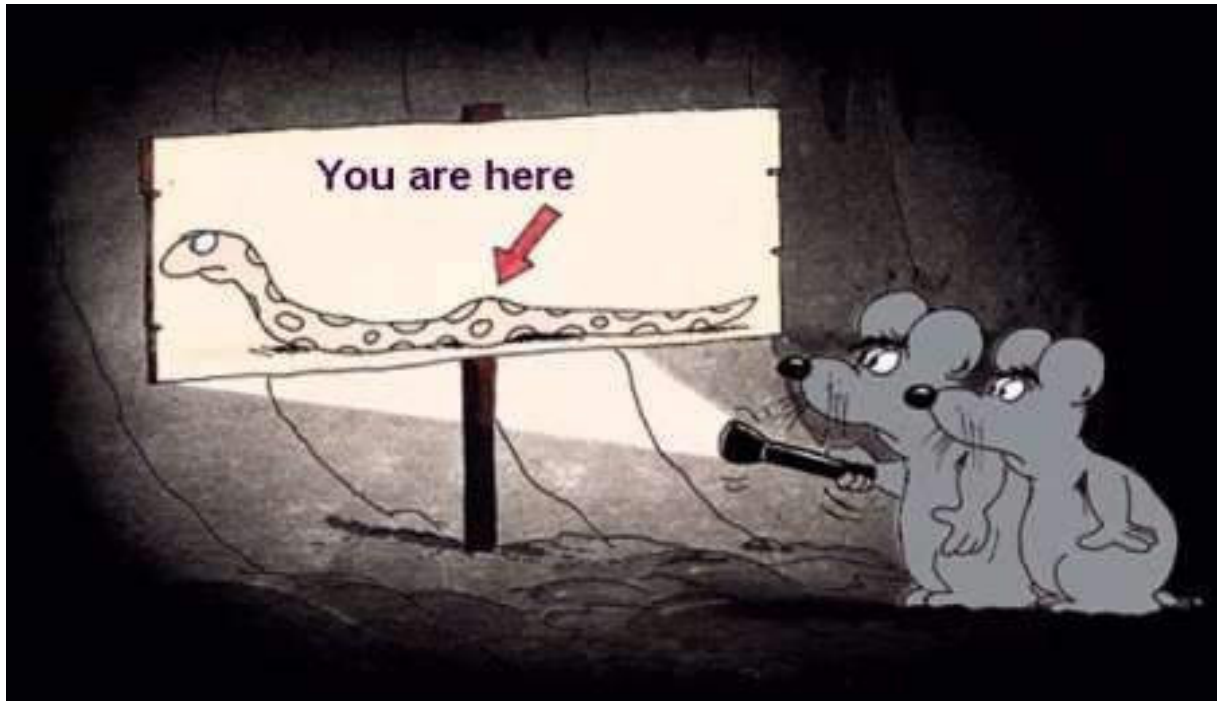
- never implements crypto primitives itself
  - kernel cryptoAPI
  - userspace crypto libraries

LUKS (Linux Unified Key Setup)

- Simple key / passphrase management system for dm-crypt

- de facto standard for disk encrytpion in Linux, portable
- **more passphrases (keyslots)**
- uses iterated PBKDF2 store key – slow down dictionary attacks
- **passphrase change** – no need to reencrypt disk
- **AF-splitter** – anti-forensic (fw sector reallocation issue)





Disk encryption ...
and what it means to forensic analysis.

Forensic analysis of encrypted drive

In theory (with proper threat model), investigator shouldn't be able to create disk duplicate with usable information at all.
(Only copy whole disk with "random-like" data.)

In reality, availability is usually more important then confidentiality.

- backdoors, master keys, admin passwords
- recovery mechanism
- predictable RNG
- legal issues with encryption (you have to provide password)
- password stored elsewhere (in file, printed on paper for recovery)
- the same password used with different service
- dictionary and bruteforce search (special hw, GPU, ASIC, ...)

Forensic analysis of encrypted drive

Forensic tools should indicate encryption is in use.

Identification of encryption – often signatures in metadata!

- Bitlocker: "-FVE-\x2dFS-" / 2D 46 56 45 2D 46 53 2D
- Safeboot: "Safeboot" / 53 61 66 65 42 6F 6F 74
- CheckPoint: "Protect" / 50 72 6F 74 65 63 74
- LUKS: "LUKS\xba\xbe" / 4C 55 4B 53 BA BE

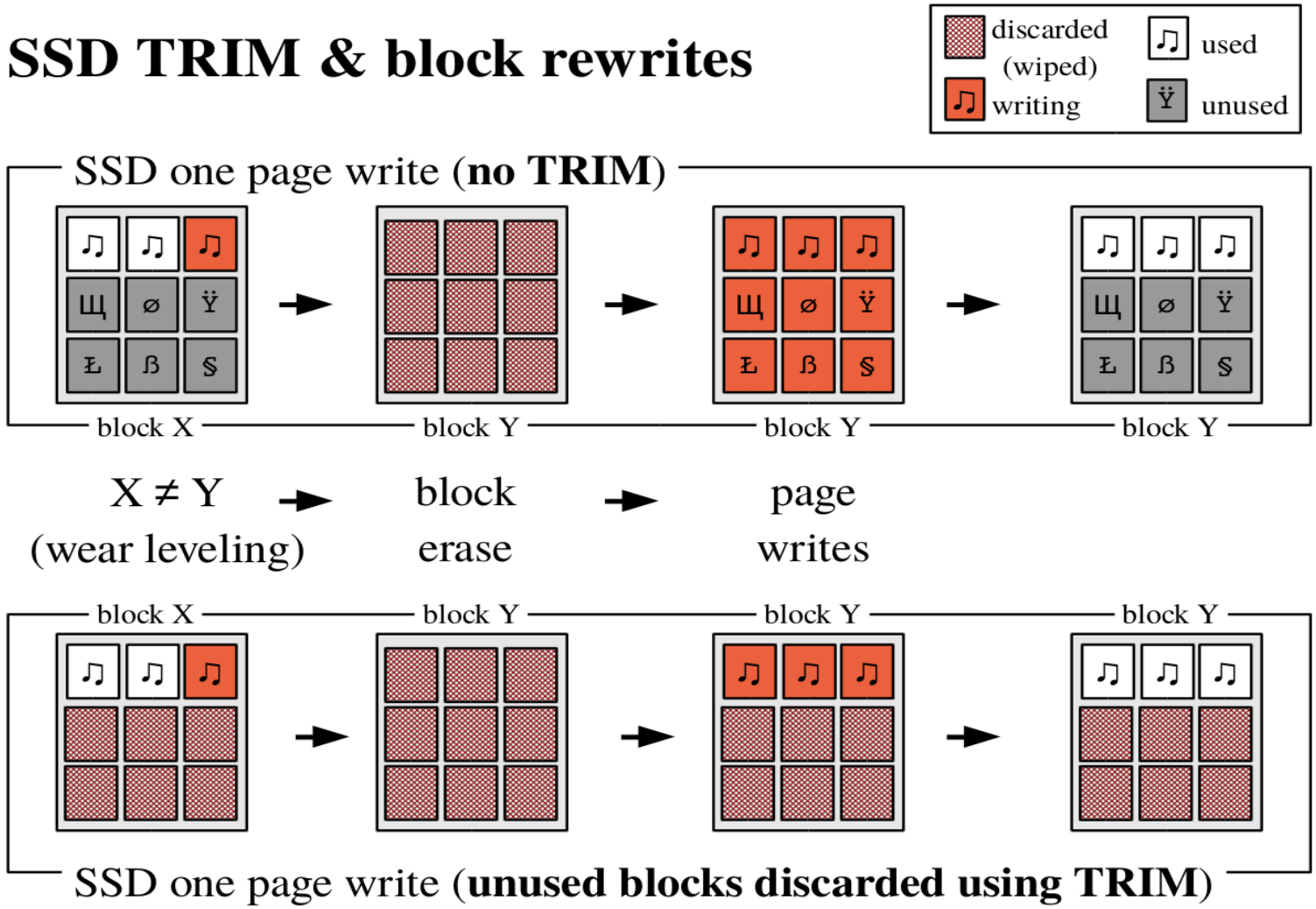
...

- TrueCrypt – none (should look like random data)
- detached metadata (LUKS)

Detection of security erase (SATA command) / TRIM / SMART logs.

SSD & TRIM

- information that certain blocks are no longer in use and can be wiped
- used in SSD (rewriting blocks is expensive), thin provisioning

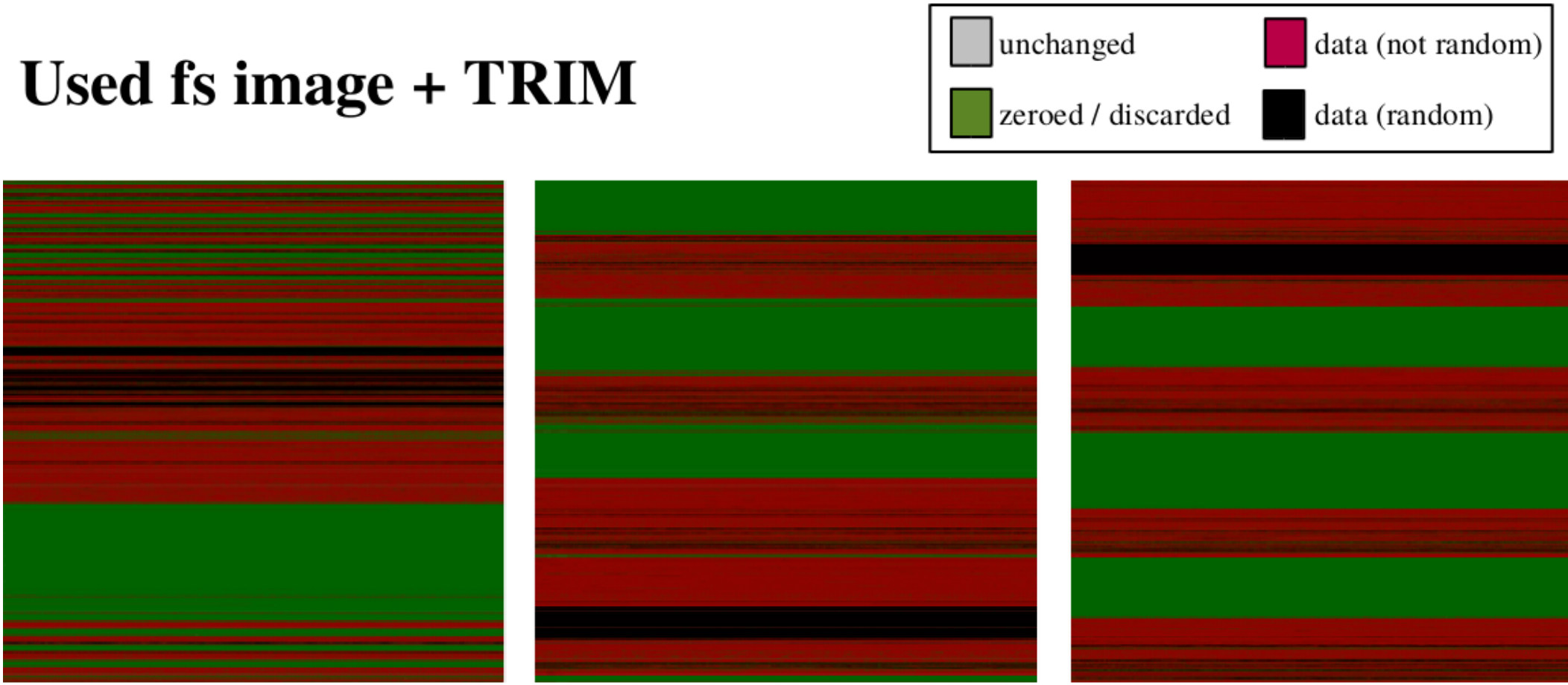


TRIM without encryption

Statistic methods – detection of random/non-random data

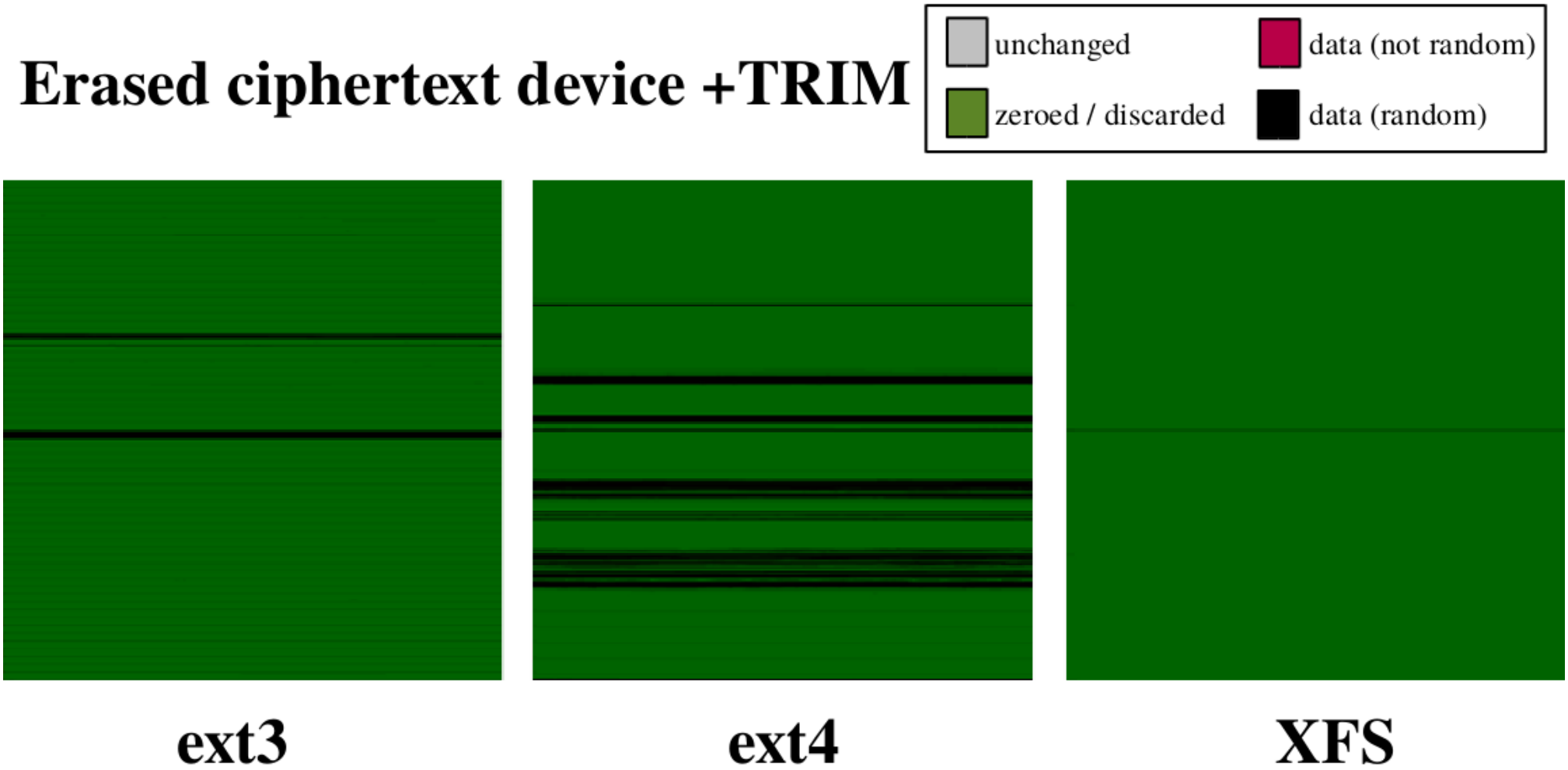
- sector is too small sample
- example of **plaintext** snapshot analysis

Used fs image + TRIM



TRIM with encryption

- trimmed block are detectable (zeroed)
- experimental proof / detecting filesystem type patterns
- here after deletion of all user data + TRIM

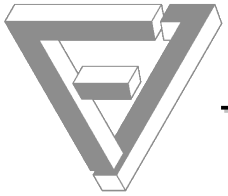


Centre for Research on
Cryptography and Security

Disk encryption ...

Thanks for your attention.

Q & A ?



Milan Brož
xbroz@fi.muni.cz

ESET

October 30, 2014