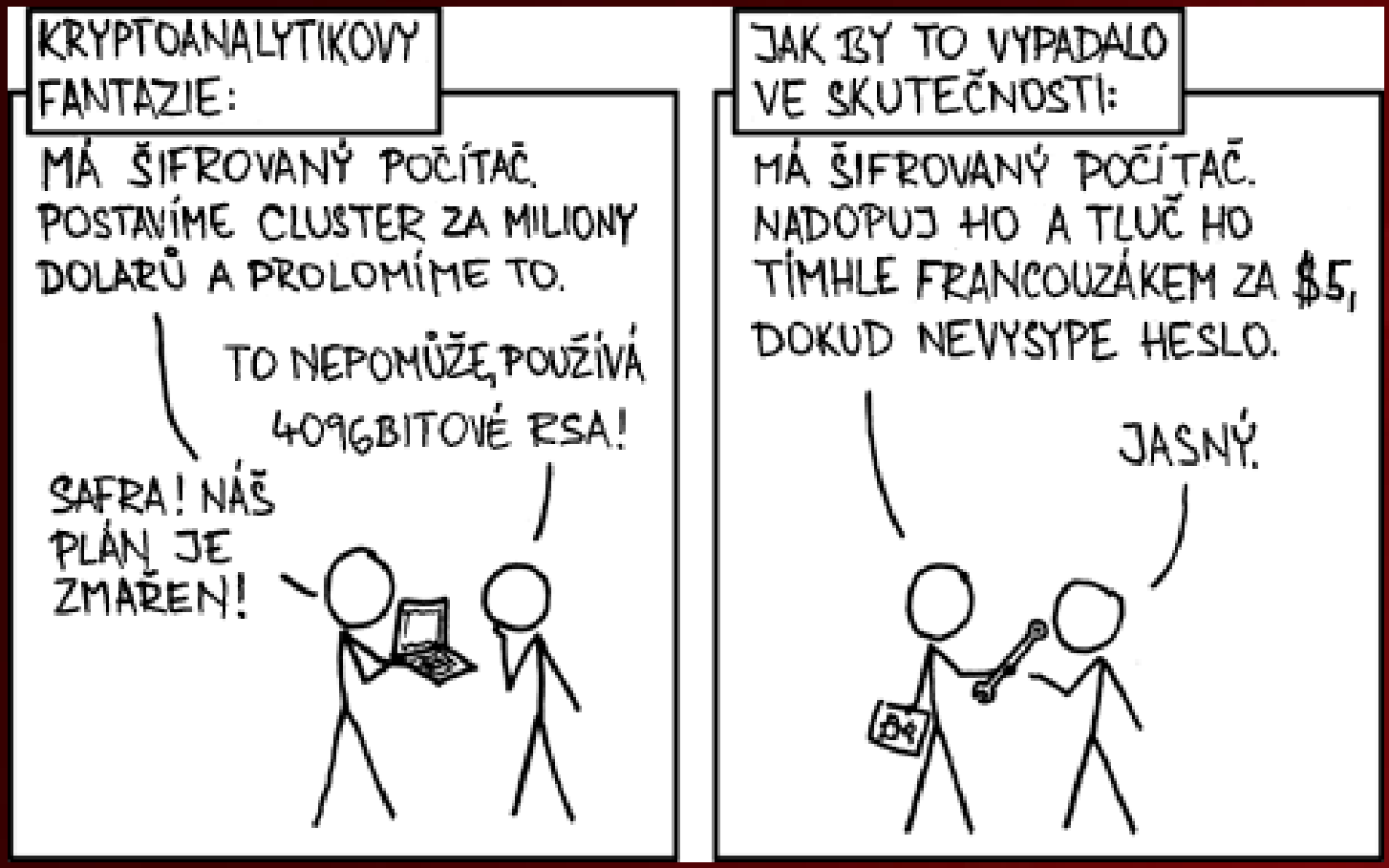
Šifrování disků ...

(nejen) v Linuxu

Milan Brož
mbroz@redhat.com

Želiv, 2011



http://www.abclinuxu.cz/images/clanky/xkcd/xkcd-538_czech.png

FDE – Full Disk Encryption (šifrování disku)

FDE - Full Disk Encryption

- šifrování na úrovni sektorů disku (každý sektor nezávisle)
- **FDE** (Full Disk Encryption) – celý disk
- **FVE** (Full Volume Encryption) – jen některé particie
- (ne)výhody?
  - + pro notebook, přenosná uložistě (offline ochrana)
  - + transparentní, není svázáno se souborovým systémem
  - + co šifrovat není na uživateli
  - + hibernace, swap
  - + odstranění klíče – likvidace dat
  - - více uživatelů – více hesel (ke všemu)
  - - únik klíče – únik všech dat
  - - některé útoky na HW lze jen omezit (Cold Boot)
  - - u sw řešení občas problém s výkonností

FDE - Full Disk Encryption

- **speciální HW** (hw based FDE)
  - HDD FDE (disk obsahuje data + key management)
  - Chipset FDE (klíč je uložen mimo disk, TPM, EEPROM, ...)

- **softwarové řešení**
  - šifrování provádí přímo hlavní CPU

- **softwarové řešení s hw akcelerací**
  - koprocesory, speciální karty
  - AES-NI instrukce, VIA padlock, ...

Blokové zařízení, sektor

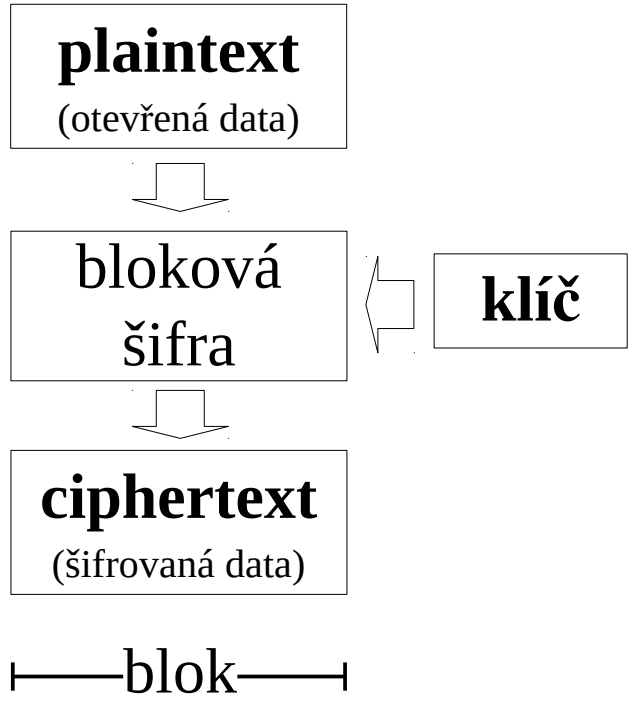
- **Sektor - atomická jednotka na disku**
  - 512 bytů, 4096 bytů
  - disk s 4k sektorem může emulovat 512b sektor – zarovnání

v Linuxu **disk = blokové zařízení**

- **sektor = blok (interně vždy 512 bytů)**
- skládání zařízení nad sebe (block device stacking)
- virtuální bloková zařízení (MD Raid, device-mapper, loop)

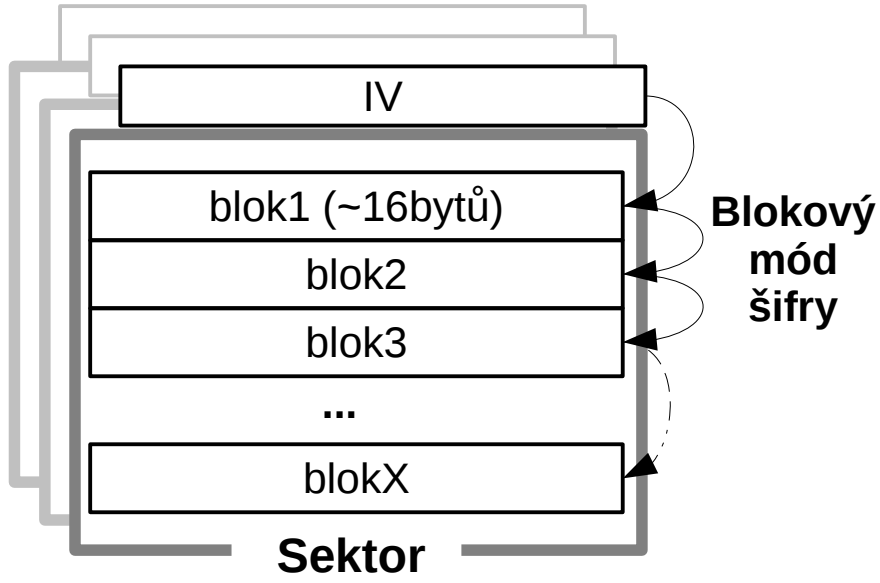
Plaintext & ciphertext

- **plaintext – otevřená** (původní) **data**
  - virtuální zařízení
- **ciphertext – šifrovaná data**
  - vlastní hw disk
- **symetrické algoritmy** (tajný klíč)
  - propustnost (~disk)
  - **blok** (obvykle 16 bytů)



Blokový mód

- **BLOK** (šifry) **< SEKTOR** (disku)
- rozdělení sektoru do bloků



- postupné/paralelní zpracování

  blokový mód

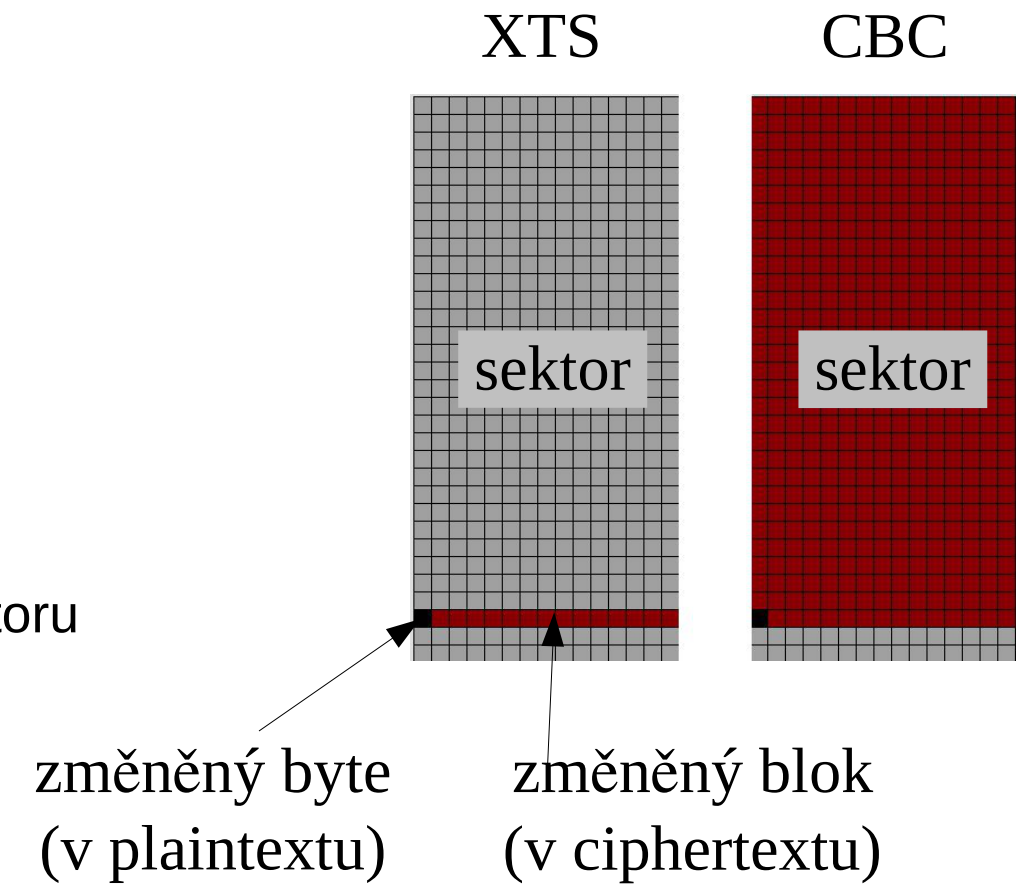
- stejná data v různých sektorech
  – různý ciphertext

  inicializační vektor IV
  (pro každý sektor jiný)

  - obvykle odvozen od čísla sektoru (případně klíče)
  - např. ESSIV – Encrypted Salt-Sector IV

Blokový mód - příklady

- sektor: jak se změna plaintextu promítne do ciphertextu?
- **CBC** – cipher block chaining
  - ciphertext XOR předchozím blokem
- **XTS / XEX** (XOR encypt XOR)
  - interně 2 klíče
    - klíč na vytvoření tweaku pro jednotlivé bloky
    - šifrovací klíč
  - IV může být přímo číslo sektoru



Blokový mód vs sektor

- ~ náhodná změna bloku ciphertextu (16 bytů) – problém?
- ideál: změna bitu (plaintext) – změna celého sektoru (ciphertext)
- **wide mód** (blok šifry = sektor)
  - složitější, náročnější na výkon (více průchodů)
  - zatížené patenty (~ volný standard EME-2)
  - prakticky se dnes nepoužívá
- **dodatečné operace**
  - příklad – **Elephant** (diffuser - Bitlocker)
    - specialní operace před aplikací CBC
    - tweak key (nezávislý)

Key Management
(správa klíčů)

Generování klíče

- **klíčové pro bezpečnost celého řešení :-)**
- **Rozdíl:** šifrovací klíč / heslo pro odemčení
- **šifrovací klíč**
  - náhodný, pro každý disk unikátní, generován nezávisle
    - nutnost kvalitního RNG (Random Number Generator)
  - odvozen od hesla
    - např. PBKDF2 (Password Based Key Derivation)
    - většinou není žádoucí (~zákázáno v bezpečnostní politice)

Uložení šifrovacího klíče

- **mimo šifrovaný disk**
  - speciální zařízení (token, SmartCard, TPM, EEPROM)
    často slabý článek – levné hw šifrované disky
  - soubor (chráněný šifrováním)
  - jiný disk (oddělená metadata)

- **na stejném disku s šifrovanými daty**
  - **metadata** (hlavička)
  - odemčení pomocí hesla či jiného klíče
  - ochrana proti útoku silou
  - ochrana proti vlastnostem hw (např. realokace sektorů)

- heslo odemyká šifrovaný klíč – změna hesla bez přešifrování disku

Odstranění klíče

- **odstranění (smazání) klíče = zničení dat**
  - **záměrné** (secure disk disposal)
  - **chybou**
    - nejčastější problém
    - přepis hlavičky – chyba administrátora
    - chyba hw, vadný sektor, řadič, TPM, ...

Obnova klíče (recovery)

- **kompromis mezi bezpečností a uživatelskou přívětivostí**

  - kopie disku (metadata), **Key Escrow** (záloha klíče)
  - **duplicitní metadata** na disku samotném
  - klíč lze vygenerovat ze záložního hesla (**recovery key**)
  - více hesel

  - problém: špatně navržená recovery strategie ničí celé řešení

Příklady aplikací pro šifrování disku

Truecrypt

Truecrypt, www.truecrypt.org

- široce používané multiplatformní řešení

- AES, Twofish, Serpent
- řetězené šifry (např. AES-Twofish)
- XTS mód

- skrytý disk (i disk s OS), bootloader
- nepoužívá TPM

- šifrované on-disk metadata
- duplicitní metadata (záloha hlavičky)
- recovery CD (při formátování)

- na Linuxu využívá dm-crypt jako backend

loop-AES

loop-AES, loop-aes.sourceforge.org

- separátní projekt mimo hlavní strom
- postaven nad loop

- AES, (volitelně Twofish, Serpent)
- modifikovaný CBC mód (IV je odvozen od sektoru, klíče a plaintextu)
- multikey – 64 klíčů (modulo sektor) + klíč pro IV

- externí uložení klíčů v GPG šifrovaném souboru

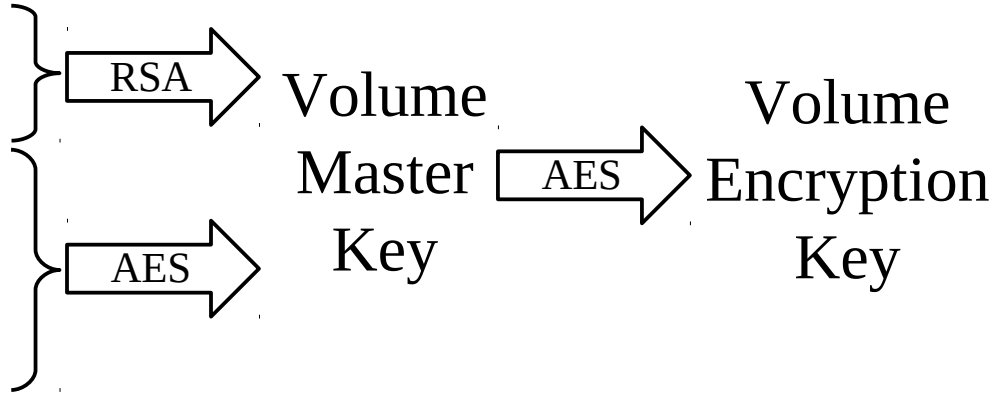
- dm-crypt / cryptsetup má kompatibilní mód

BitLocker (Windows)

Nativní řešení FDE nejvyšší řady Windows

- v budoucnu společně se "secure boot" (Windows 8)
- mnoho kombinací (lze povolit v systémové politice)

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



- AES 128 CBC
- AES 128 CBC + Elephant (diffuser)
- AES 256 CBC
- AES 256 CBC + Elephant (diffuser)

LUKS / dm-crypt

- **Nativní řešení v Linuxu**

- **striktní oddělení**
  - **implementace vlastního šifrování disku**
    dm-crypt – device-mapper crypto target (kernel modul)
  - **správy klíčů (LUKS) a konfigurace**
    (**cryptsetup** – userspace aplikace)

- Neimplementuje žádná šifrovací primitiva
  - využití kernel cryptoAPI
  - crypsetup používá knihovny (volitelně gcrypt, openssl, nettle, NSS)

dm-crypt (low-level)

- vytvoří virtuální plaintext device nad diskem
- neřeší správu klíčů

- kernel modul

- variabilita, podporovány všechny šifry a módy (cryptoAPI)
  (s výjimkou diffuseru), HW akcelerace
- podporuje stackování (~ řetězené šifry)

Příklady specifikace šifry

- aes-cbc-essiv:sha256 (AES, CBC, ESSIV)
- aes-xts-plain64 (AES, XTS, IV je číslo sektoru)
- aes:64-cbc-lmk (loop-AES kompatibilní mód – multikey)

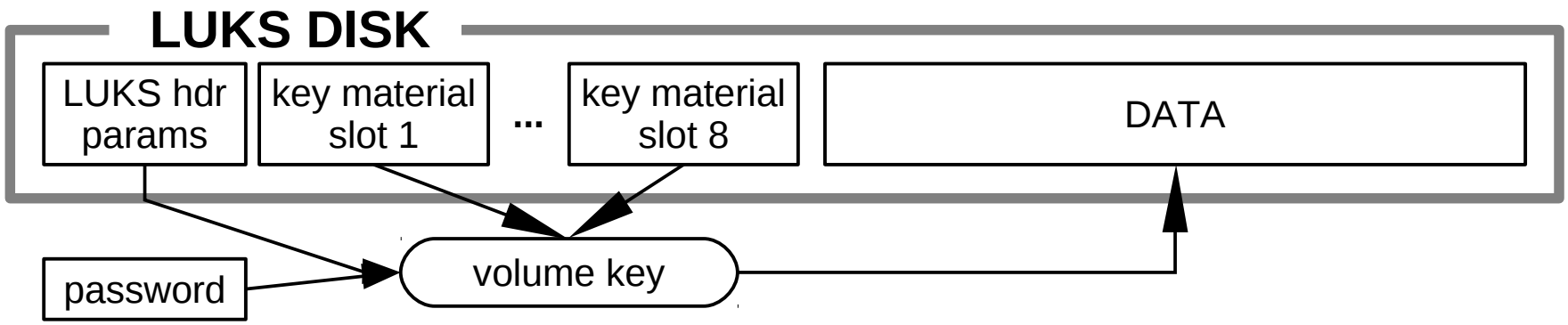
Lze nastavit i kompatibilní módy (i když nejsou bezpečné)

- twofish-ecb
- serpent-cbc-plain64

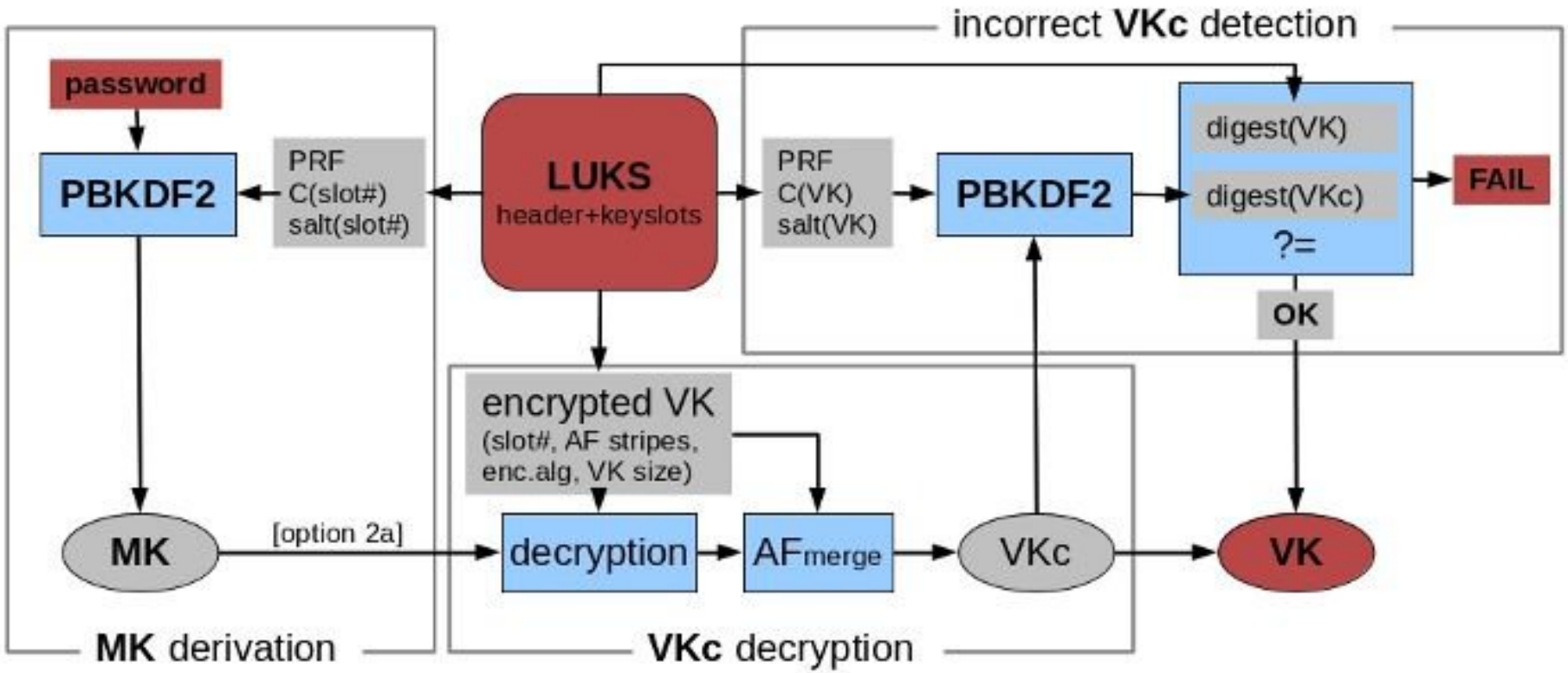
...

LUKS (Linux Unified Key Setup)

- de facto standard pro konfiguraci šifrování disku v Linuxu
- přenositelný, podporovaný i jinými OS (FreeOTFE.org)
- **podpora více hesel (slotů)** – odemyká **volume key**
- velký počet iterací (PBKDF2) – ochrana proti slovníkovému útoku
- **změna** (zneplatnění) hesla bez nutnosti přešifrovat celý disk
- **AF-splitter** – anti-forenzní ochrana
  (proti obnovení hesla z realokovaných sektorů)



LUKS volume key recovery



MK – derived master key
VKc – volume key candidate
VK – recovered volume key
AFmerge – anti-forensic merge
PRF – pseudorandom function
C – iteration count

operations
input/output data/key
internal data/key

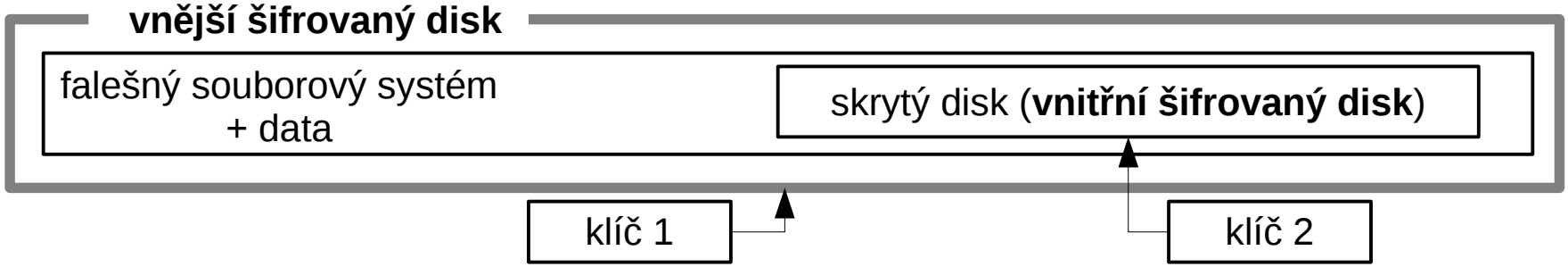
http://www.schneier.com/blog/archives/2010/01/tsa_logo_contes.html

Zajímavé problémy...

Skrytý disk (hidden disk)

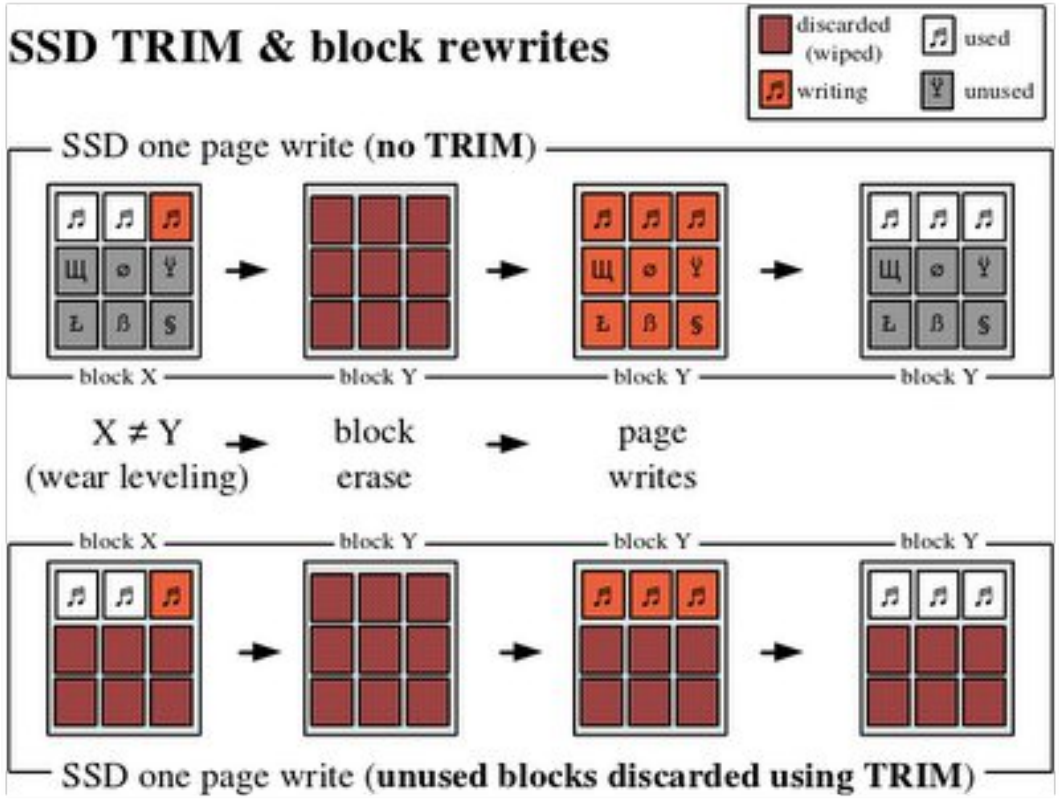
- **plausible deniability**
  schopnost „uvěřitelně“ popřít, že jsou na disku nějaká data

- data jsou ukrytá v nepoužívaném prostoru, ke kterému je nutný další klíč
- bez klíče šifrovaná data nelze rozeznat od šumu
- šifrovaná data nemají viditelnou hlavičku



SSD & TRIM

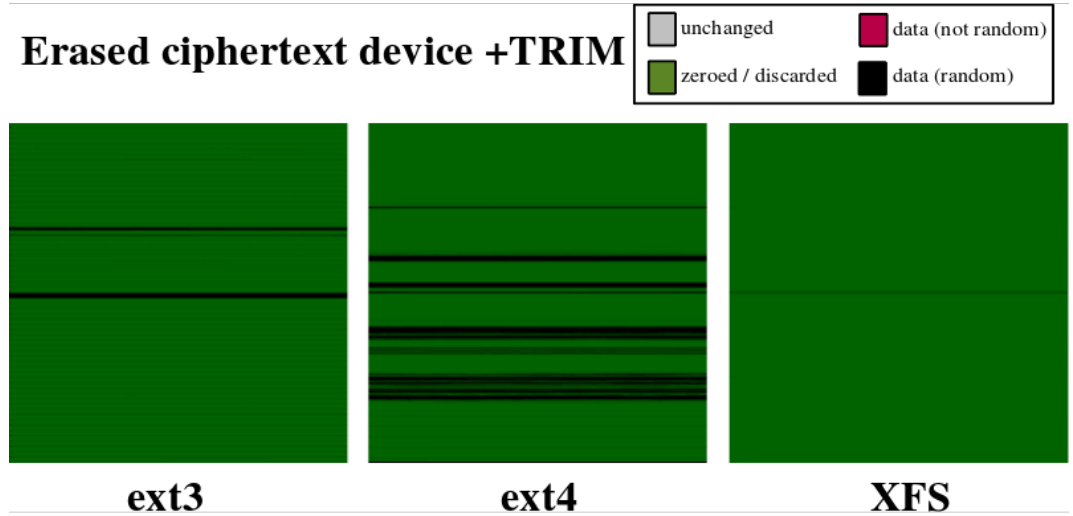
- TRIM – informace pro hw disku o uvolnění sektorů

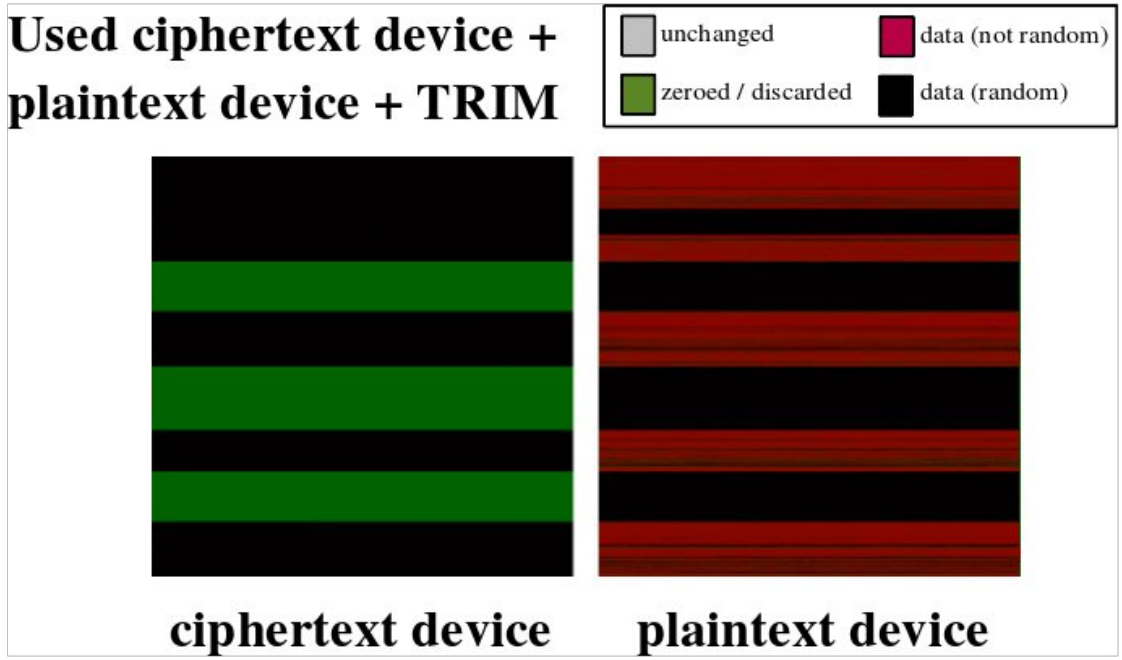


- pro dlouhodobou výkonnost SSD (Solid State Drive)

TRIM & FDE

- bloky obsahují po provedení TRIM obvykle samé nuly
- nuly se pak interpretují jako ciphertext
- ze zařízení lze detekovat volné místo
- vedlejší kanál, specifický vzor (např. je vidět typ fs)





TPM (Trusted Platform Module)

• kryptoprocesor (obvykle na základní desce)

• základní komponenta pro „trusted boot“
viz aktuálně Windows 8 / UEFI / secure boot

toto je tlustá černá čára

• při použití s FDE jde však jen o uložistě klíčů
(TPM vydá klíč pokud boot probíhá „v pořádku“)

• disk je tak svázán s daným HW.
(ale je možné Key recovery např. pomocí recovery password!)

• Po vydání klíče je šifrovací klíč přítomen v RAM.

• TPM chrání před neautorizovanou změnou v konfiguraci bootu...
... která vyžaduje fyzický nebo administrátorský přístup do systému.
(A obecně se v této situaci nelze chránit...)

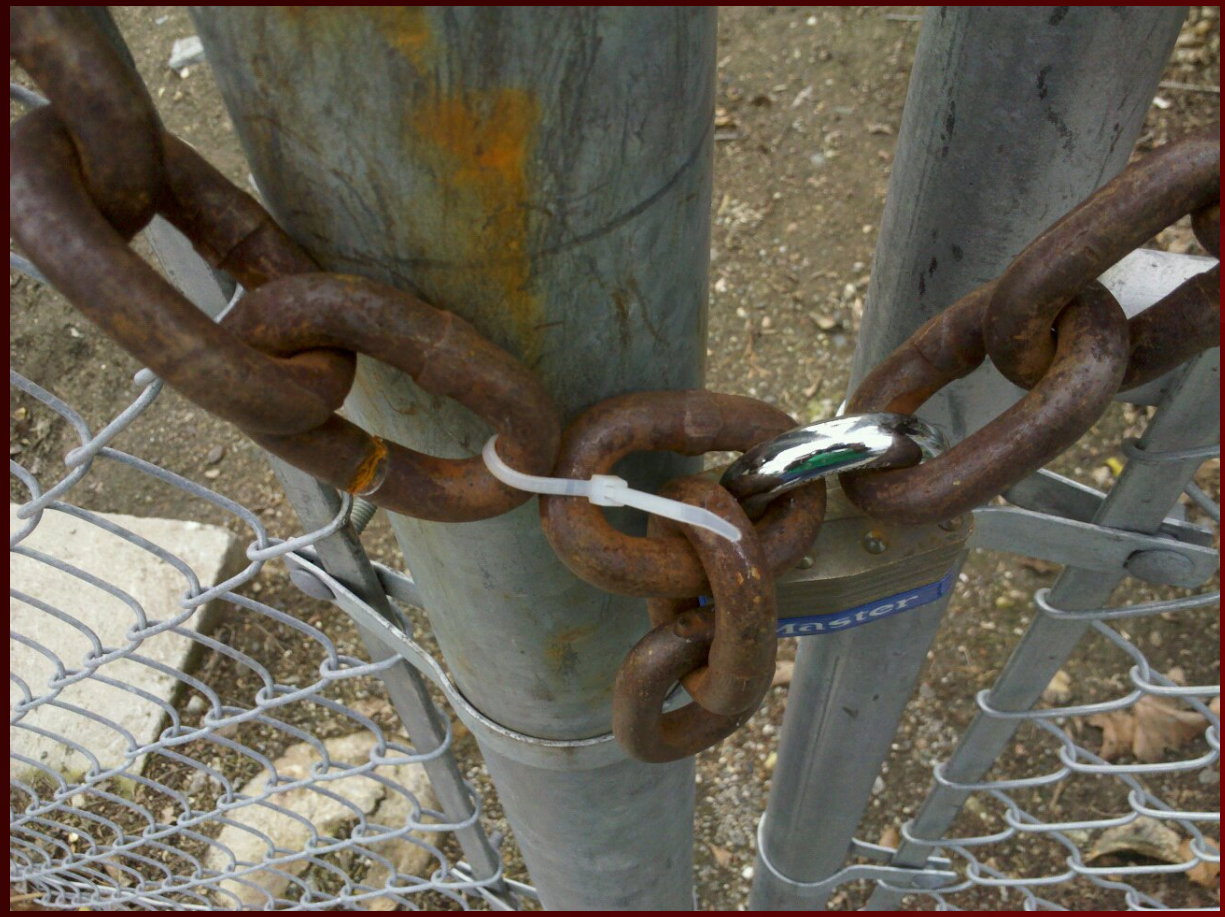
• DRM (!) tedy bude fungovat, ochrana FDE klíče už ne tak docela.

Šifrování disku + datového kanálu k disku

- *iSCSI exportovaný disk, dešifrování probíhá na klientské straně (data jsou tedy na síti šifrována).*

- Zásadní rozdíl
  - **FDE řeší offline ochranu** (ukradený disk)
    - útočník nemá k dispozici snapshoty zařízení (pokud má, musel mít opakovaný přístup k hw!)
    - blokový šifrovací mód je určen pro disk (IV je pro sektor konstantní)
  - **Šifrovaná síťová komunikace**
    - řeší problém, kdy útočník má dispozici záznam komunikace (nelze vložit jiný obsah sektoru, připadně záznam použít znovu)

- **Oba dva problémy je třeba řešit odděleně.**



Zajímavé útoky...

Attacks always get better, they never get worse.

Útoky se vždy jen vylepšují, nikdy se už nezhorší.

- **Útok na vlastní algoritmus**
- **Útok na implementaci**
  - například využití postranních kanálů
- **Získání klíče nebo hesla v otevřené podobě**
  - napadení hw (keylogger, Cold Boot)
  - malware – modifikace boot, OS, hypervisoru
  - social engineering

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

Počítač ponechaný chvíli bez dohledu už nemusí patřit jen vám.

No security product on the market today can protect you if the underlying computer has been compromised by malware with root level administrative privileges.

That said, there exists well-understood common sense defenses against “Cold Boot,” “Stoned Boot,” “Evil Maid,” and many other attacks yet to be named and publicized.

Marc Briceno, PGP Corporation
http://blog.pgp.com/index.php/2009/10/evil-maid-attack/

V současnosti neexistuje žádné bezpečnostní řešení, které vás ochrání, když v počítači je instalován zákeřný software běžící s právy administrátora systému.

Přesto existují dobře známé, rozumné a snadno pochopitelné postupy, jak se bránit proti „Cold Boot“, „Stoned Boot“, „Evil Maid“ a mnoha dalším útokům, které na své pojmenování a zveřejnění teprve čekají.

Útok Cold Boot

Lest We Remeber: Cold Boot Attacks on Encryption Keys
Princeton University, http://citp.princeton.edu/memory/

- **Paměť DRAM udrží data ještě nějakou dobu po výpadku napájení**.
- Pro šifrování, které provádí hlavní procesor,
  musí být v paměti RAM přítomen klíč.
- Pokud získáme klíč k blokové šifře, **již není třeba žadná hesla**.
- Odchyťme tedy klíč z paměti po násilném resetu systému nebo uspání.

Příklad: útok Cold Boot na dm-crypt (LUKS)

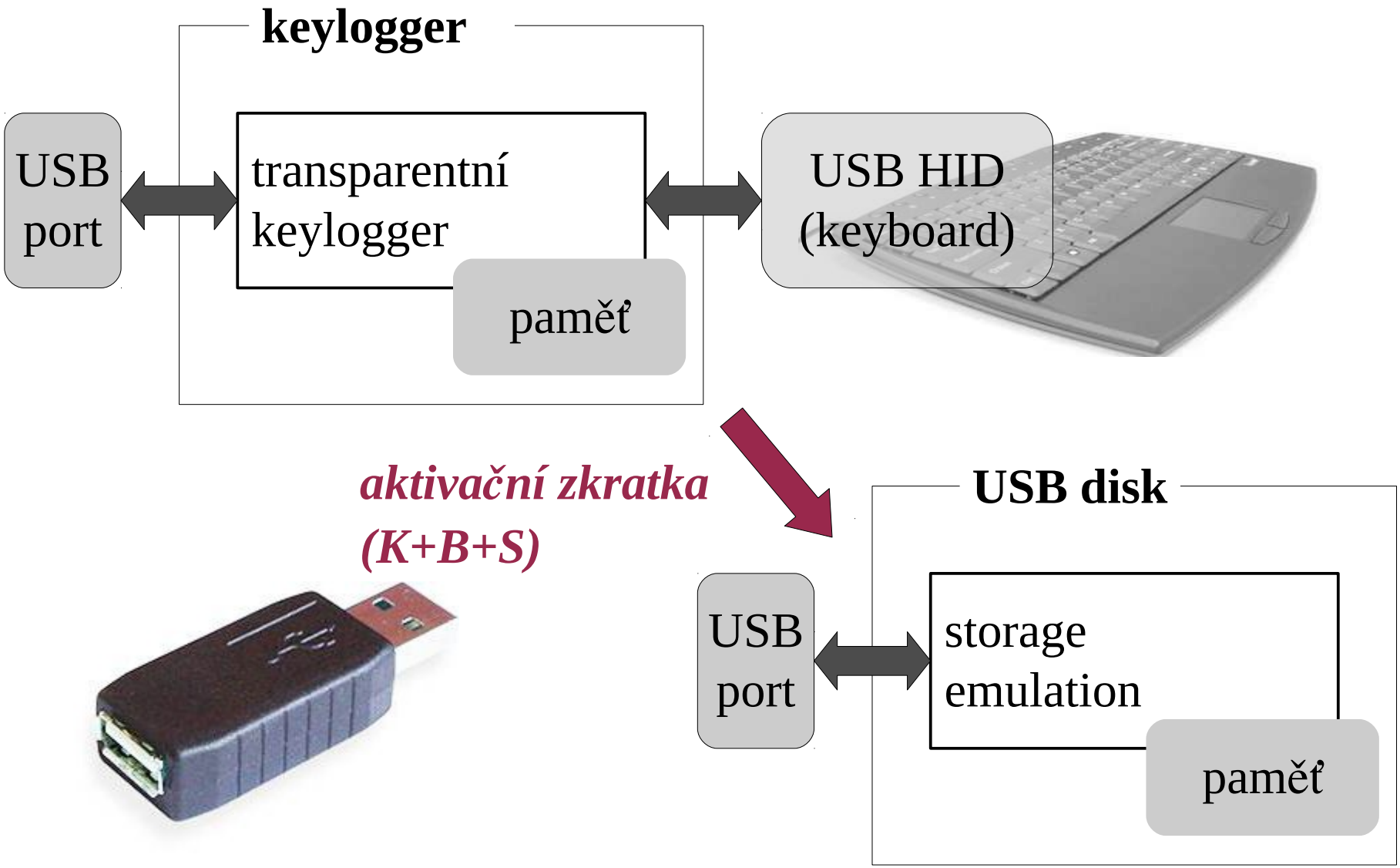
Konfigurace počítače, na který se útočí

```
[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... 0 8:2 4040
vg-lv_swap: 0 2818048 linear 253:0 153051520
vg-lv_root: 0 153051136 linear 253:0 384
```

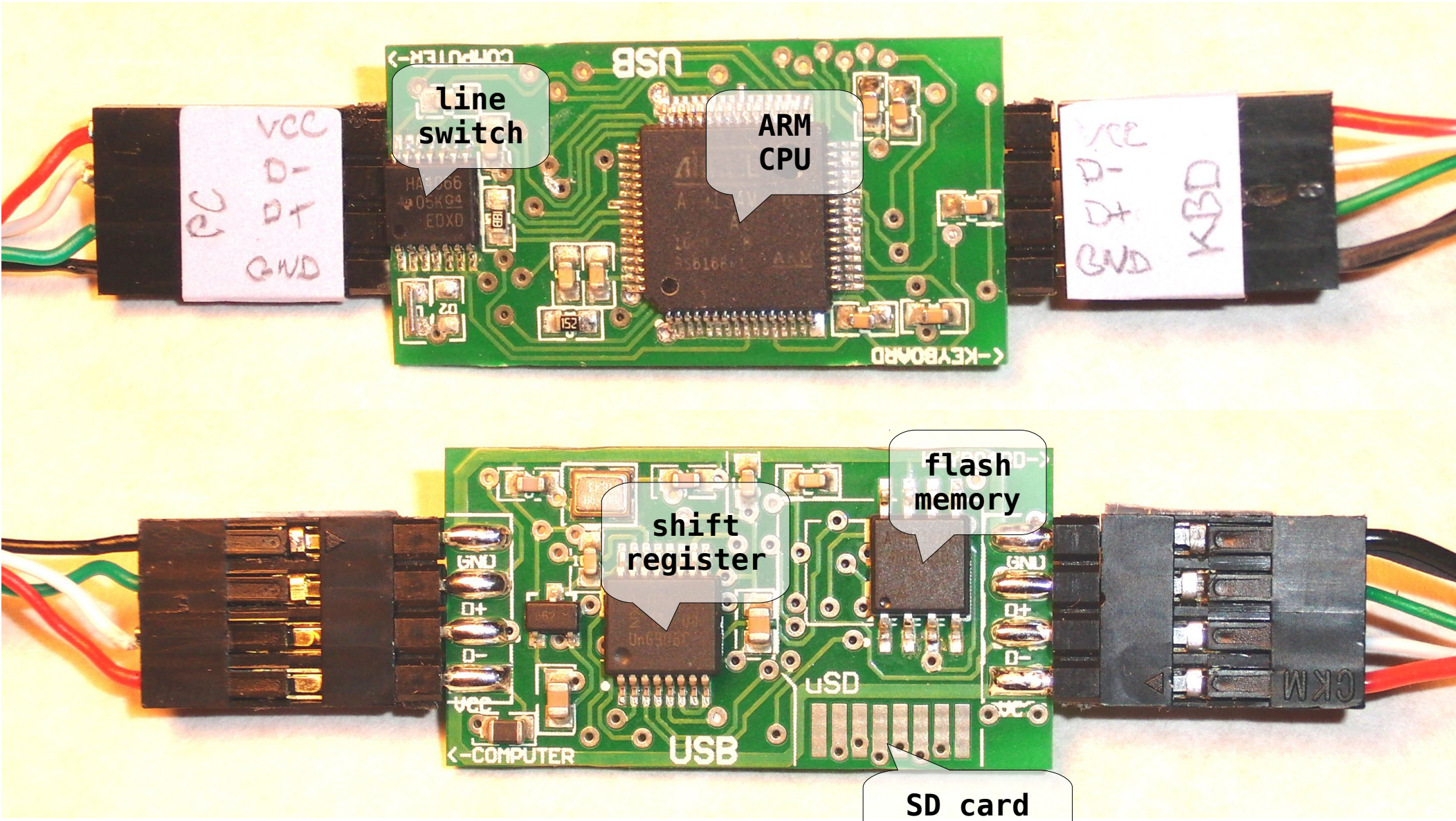
- **násilný reset**
- **nabootování PXE zavaděče**
- **přenesení RAM image přes síť**
- **sken na přítomné klíče**

```
$ ./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%
```

USB hw keylogger



USB keylogger (příklad: KeyDaemon)



děkuji za pozornost