Centre for Research on
Cryptography and Security

The Future of Disk Encryption ...
... with LUKS2

Milan Brož, Ondřej Kozina
mbroz@redhat.com, okozina@redhat.com

DevConf, Brno
February 7, 2016

Agenda

- Linux Unified Key Setup (LUKS)
- Disk Encryption Use Cases
- (Mention of) Cryptography
- LUKS2
- Online Reencryption
- "User Survey" Notes

FDE – (software) Full Disk Encryption

- **Transparent encryption on disk sector level**

  - Transparent for filesystem
  - No user decision what to encrypt
  - Encryption of hibernation and swap partitions

- **Volume key** – key used to encrypt data

- **Passphrase** – unlocks encrypted disk

Linux FDE

- **dm-crypt** (kernel module) + **cryptsetup** (control utility)
- **LUKS** (Linux Unified Key Setup)

  - On-disk format to store encrypted volume key

  - Implemented inside cryptsetup library

LUKS history

2004 dm-crypt (kernel 2.6.4) + cryptsetup 0.1 [J.Saout]

- Volume key derived from passphrase

2005 cryptsetup-luks (LUKS extension) [C.Fruhwirth]

- Key is random, encrypted in keyslots
- Compatible on-disk format
- Independent keyslots

2012+ stable libcryptsetup API

- loopAES, TrueCrypt support

LUKS Common Use Cases

- **Local encrypted disk**

  - Encrypted notebook, portable drives, ...

  - Corporate notebooks – on-demand recovery

- **Datacentre disks**

  - Different physical access policies in-place

  - Data disks (also Gluster bricks, Ceph OSDs, ...)

  - Automatic unlocking?

- ***Mobile devices***

  - Specific environment, usually non-LUKS metadata

LUKS, Threat Example

- Asset: **Confidential data on-disk**

  Threat: **Stolen disk**

  => Strong encryption with random key

  => Dictionary password attack resistance

- LUKS provides data confidentiality only

- No integrity protection

- Protection only of locked (powered-off) device

Cryptography and Disk Encryption

Key Management

- It's all about weak passwords :-)
- Password-based key derivation functions
    - PBKDF2
    - Argon2 (PHC winner, planned)
- No Trusted Platform Module (TPM) bindings
- No 2nd factors authentication.
- No secret sharing.

Disk Sectors Encryption

Block Cipher (like AES) – Encryption Modes

- Narrow modes per sector (CBC, XTS)
- No wide mode (patents!)
- No support for authenticated encryption modes
    - today just "Poor man authentication"

- Volume key change, algorithm change, ...
    - Device reencryption
    - Not possible online

LUKS2 ... OMG Why?

Lifetime of data on encrypted disk is long-term.

We have to think in this time frame.

Security Hardening

- Key derivation – PBKDF2 is not fixable in long term
    - GPU, ASIC speedup, no threads, no memory cost
- Integrity: no option for it
- Volume key or encryption upgrade (online)

LUKS2 ... OMG Why?

Missing Extensibility

- No specific key slot processing
    - Using TPM, HSM, PKCS#11 SmartCards
    - Remote key or automatic unlocking
- Independent keyslot attributes

No header metadata redundancy

No header visible metadata corruption detection

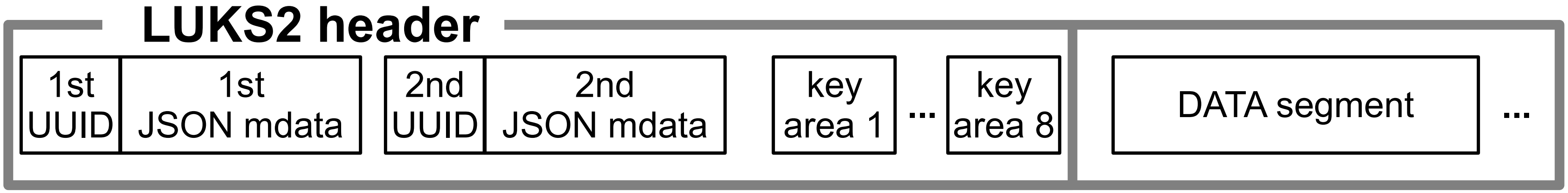
...

Note that LUKS2 is still an experiment!

LUKS2 on-disk

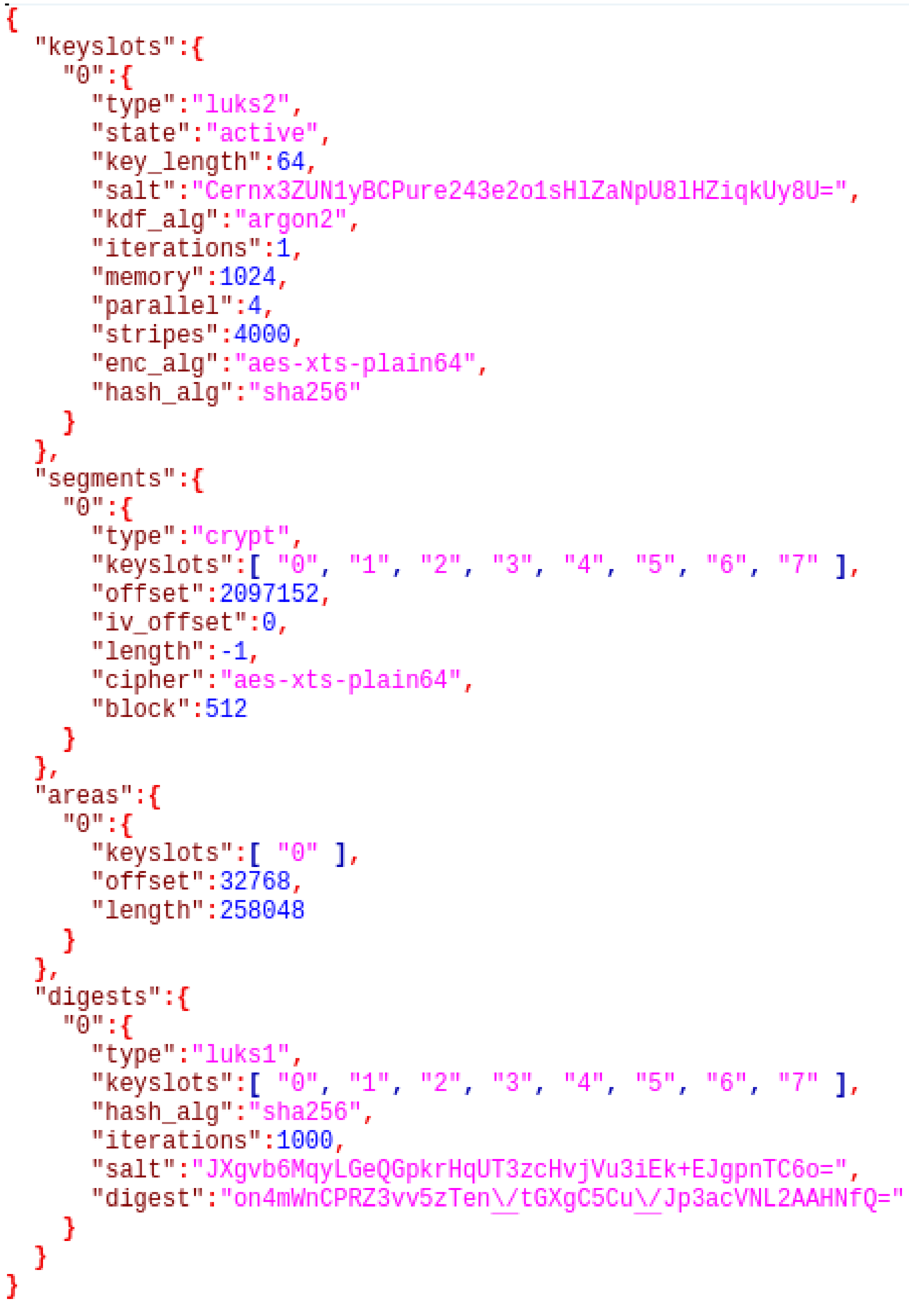
- New features without on-disk format change
- Abstraction, keyslot handlers interface ("plugins")
- In-place upgrade / downgrade from LUKS1 (partially)
- LUKS2 targets more "enterprise"
- LUKS1 remains stable, supported "forever"

LUKS2 on-disk Schema



- Redundant header (NOT redundant key data)
- Header corruption detection (checksum)
- Resistance to write on fail (epoch recovery)
- Binary part (for blkid – UUID, magic, ...)
- Extensible metadata format (JSON)

LUKS2 on-disk JSON Schema



```
{
  "keyslots":{
    "0":{
      "type":"luks2",
      "state":"active",
      "key_length":64,
      "salt":"Cernx3ZUN1yBCPure243e2o1sHlZaNpU8lHZiqkUy8U=",
      "kdf_alg":"argon2",
      "iterations":1,
      "memory":1024,
      "parallel":4,
      "stripes":4000,
      "enc_alg":"aes-xts-plain64",
      "hash_alg":"sha256"
    }
  },
  "segments":{
    "0":{
      "type":"crypt",
      "keyslots":[ "0", "1", "2", "3", "4", "5", "6", "7" ],
      "offset":2097152,
      "iv_offset":0,
      "length":-1,
      "cipher":"aes-xts-plain64",
      "block":512
    }
  },
  "areas":{
    "0":{
      "keyslots":[ "0" ],
      "offset":32768,
      "length":258048
    }
  },
  "digests":{
    "0":{
      "type":"luks1",
      "keyslots":[ "0", "1", "2", "3", "4", "5", "6", "7" ],
      "hash_alg":"sha256",
      "iterations":1000,
      "salt":"JXgvb6MqyLGeQGpkrHqUT3zcHvjVu3iEk+EJgpnTC6o=",
      "digest":"on4mWnCPRZ3vv5zTen\/tGXgC5Cu\/Jp3acVNL2AAHNfQ="
    }
  }
}
```

Keyslots

- "How a key is stored and encrypted"
- Typed (handler plugins)
- Several keyslots – the same key

Segment(s)

- "Where are the user data"
- How are encrypted
- Link to keyslots with key

Area(s)

- Binary area for keyslots (if needed)
- Non-redundant key material

Digests

- "How to check derived key validity"

Why Reencrypt?

- Different data lifetime and algorithm lifetime
- Prevent access to the data from header backup
- Mitigate risk of device snapshot replay attack
- Regular volume key change (policy)
- Offline reencrypt utility available since cryptsetup v1.5.0

Why Online?

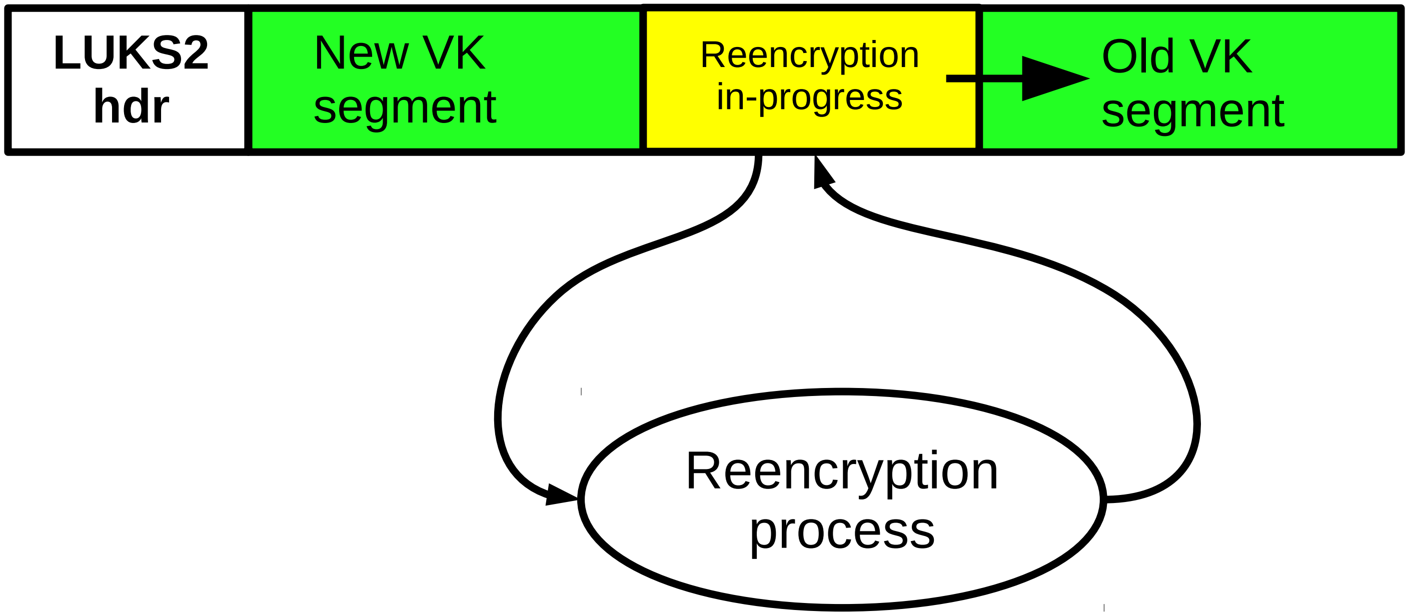
- Full disk (re)encryption may take long time
  - Not likely feasible offline with HA systems

- Complicated offline reencryption of root device
  - Limited set of tools to support error recovery
  - Interruption could make system unbootable

Online Reencryption – New Features

- Resilient reencryption metadata
  - stored in LUKS2 format (inside header)
- Device can be unlocked even after
  - Intentional interruption (SIGTERM)
  - System crash
  - Power fail
- Interrupted reencryption resume (on demand)
- Device can be unlocked even if partially reencrypted

Reencryption Progress Schema



- Sliding window
- Resistance to interruption (hash of old data)

NOW FOR SOMETHING COMPLETELY DIFFERENT...

Disk Encryption "User Survey"

- Collected lot of ideas how to [not] ask IT people :-)
- Many of us did not understand it was about "feeling"
- ... Eventually only Red Hat participated (memo-list)
- 141 completed responses
- Part of Bachelor thesis

  https://is.muni.cz/th/409782/ (not visible yet)

Disk Encryption "User Survey"

- 6% does not use encryption (despite company policy :-)
- 96% believes that encryption increases security
- 20% lost data on encrypted disk at least once
  - 59% of them lost data forever
  - 18% of them suffered corruption of encrypted disk
- 62% have backups of encrypted data
- 1% have problem with slowdown caused by encryption
  - 75% did not notice, 19% negligible slowdown

Conclusion

- We need both LUKS and LUKS2 formats
  - LUKS2 provides extensibility interface
  - Plugins will come later (with your help!)
- Integrating new strong cryptography
  - Conservative way
- Think about providing user-friendly way to
  - Setup secured systems
  - Backup and recovery integration
  - Painless upgrade path

redhat

CRoCS
Centre for Research on
Cryptography and Security

Thanks for your attention.

Q & A ?

mbroz@redhat.com
okozina@redhat.com

DevConf, Brno
February 7, 2016