LUKS2 Disk Encryption and OpenSSL

Milan Brož
mbroz@openssl.org

Milan Brož
mbroz@openssl.org

Intro

- maintainer of cryptsetup / LUKS2 project since 2009
- researcher in storage security

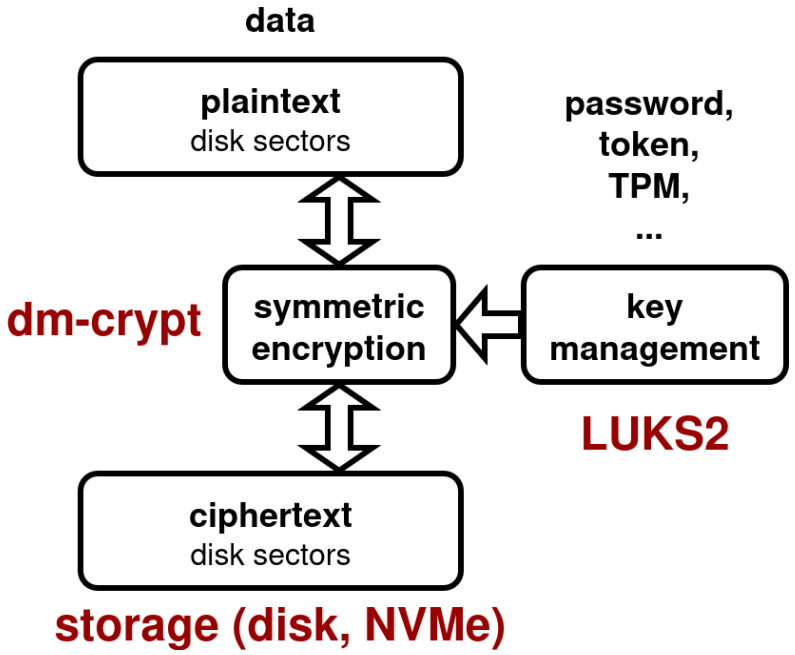
  ... and now employed by OpenSSL

This talk is about Linux open-source disk encryption that uses the OpenSSL library.

Cryptsetup is (and will be) developed independently of the OpenSSL Corporation.

Disk (sector) encryption, LUKS2, dm-crypt...

- LUKS2 (Linux Unified Key Setup)
- dm-crypt (Linux kernel driver)
  - kernel crypto API backend
- (planned) ublk alternative
  - dm-crypt replaced by userspace daemon
  - OpenSSL backend
- cryptsetup project implements LUKS1/2 (and other) formats

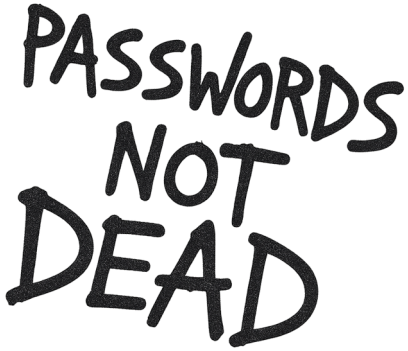


cryptsetup uses OpenSSL as default cryptographic library backend

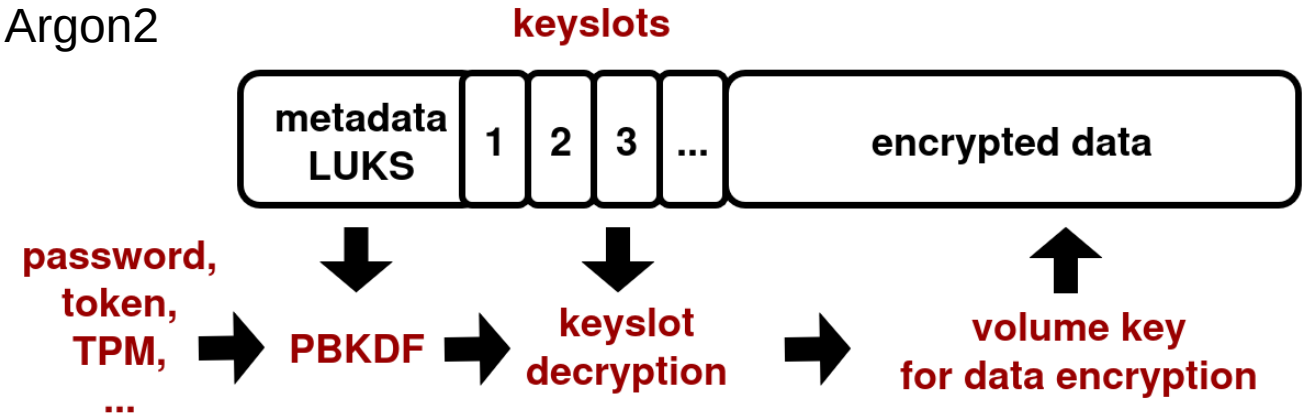
How it relates to OpenSSL?

- two areas
  1) key management (LUKS key slots handling)
     - password-based key derivation
  2) data encryption (algorithms and modes)
     - symmetric encryption
     - authenticated encryption (AEAD)
- prototyping of new algorithms
  - OpenSSL providers

LUKS2 key management



- LUKS uses key hierarchy (keyslot key, volume key)
- keyslot key is derived from “password” with PBKDF
- PBKDF - Password-Based Key Derivation
- PBKDF2, Argon2



Argon2 KDF extension for OpenSSL was initiated as part of cryptsetup development

Argon2 KDF & OpenSSL

- PBKDF2 can be highly optimized on GPUs / ASICs
  - very fast and cost-effective brute force search
- Argon2 was selected in Password-Hashing Competition (2015) as a replacement
- Argon2 is memory-hard algorithm
  - iterations, threads and required memory costs
- NIST / FIPS still do not “allow” it
  (despite mentions of memory-hard KDFs in docs)

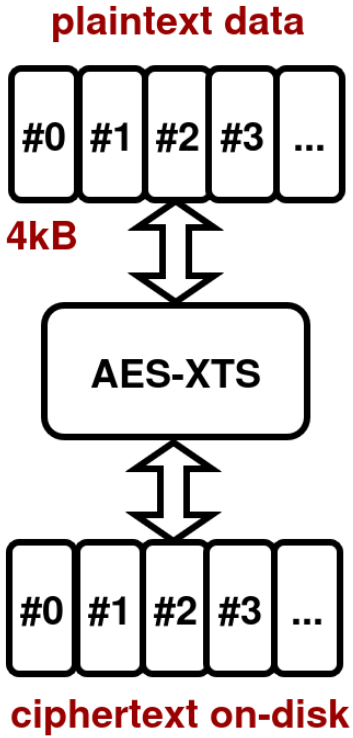
Argon2 KDF extension for OpenSSL was initiated as part of cryptsetup development

Argon2 KDF & OpenSSL II.

- Argon2 used in LUKS2 since 2014
- we started with embedded reference code
- Argon2 integration to OpenSSL was initially bachelor thesis (by Čestmír Kalina)
  - started 2019, defended 2020
  - just ~3 years of discussions
    (yeah, I know, it required adding threads :-)
  - finally released with OpenSSL 3.2 (2023)

Disk sector encryption

- “the last sort of cold storage encryption”
- disk sector encrypted independently
  - tweaked by sector number
  - length-preserving encryption
- provides confidentiality, not integrity protection
- performance is important here
- today everyone uses XTS mode (AES-XTS)



Authenticated encryption myth

- cryptography textbooks mention disk encryption must be length-preserving
- **Authenticated Encryption (with Additional Data) - AEAD is possible - we only need space for authentication tag**

- but this is exactly what enterprise NVMe drives can provide today!
- firmware for common drives ...

```
$ nvme list /dev/nvme0n1
Node             ...  Format
---------------- ...  --------------
/dev/nvme0n1     ...   4 KiB + 64 B
```

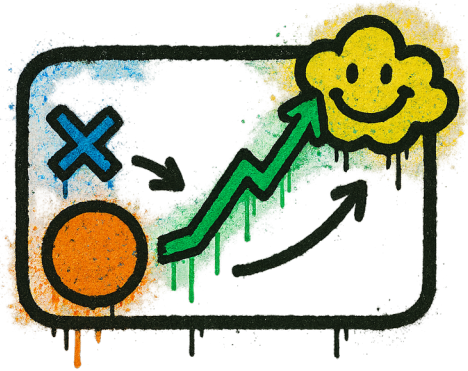
XTS encryption mode issues

- designed for performance
- used in all major disk encryption schemes today (even hardware like Opal)
- increasing storage capacity uncovers serious issues for XTS
  - increased probability of collisions, leading to new attacks
- IEEE (and NIST) is currently updating XTS specification
  - introducing key scopes (maximal amount of data for one key)
- some selected parameters remains mystery, we tried to summarize it
  XTS mode revisited: high hopes for key scopes?
  https://arxiv.org/abs/2502.18631
- there are alternatives (Adiantum, HCTR2) and
  even new research (like double-decker algorithms)

Standards, certifications & Co.

- disk encryption does not need post-quantum cryptography
- it needs more suitable symmetric encryption modes
- storage encryption is very long-term (decades)
  - we need to prepare now
- current standards are focusing on preserving existing modes (HW)
- we need try to use new secure encryption modes / algorithms
  - while keeping ***less secure*** options where certification is needed

What’s the plan?



- do not invest time in fixing unfixable
  (XTS key scopes)
- prototype use of new encryption algorithms and modes
  - using OpenSSL providers
  - revise authenticated encryption options
- hope that standards allowing new algorithms are updated
- hope that storage vendors will support sector extensions for common drives
  (we can already emulate it in software through dm-integrity)

Milan Brož
mbroz@openssl.org

Thanks for your attention!

Questions?

openSSL CORPORATION