BitLocker disk encryption on Linux

Vojtěch Trefný
mail@vojtechtrefny.cz

DevConf CZ, 25. 1. 2020

twitter.com/vojtechtrefny
github.com/vojtechtrefny
gitlab.com/vtrefny

BitLocker

BitLocker

- Native full disk encryption for Microsoft Windows.
- First introduced in 2006 in Windows Vista.[1]
  - A new version of on-disk metadata was introduced in Windows 7.
  - New algorithms for the data encryption introduced in Windows 8 (AES-CBC) and Windows 10 (AES-XTS).
- Supports encryption of both system drive and removable devices (BitLocker ToGo).
- The on-disk metadata format is not open but there is enough public information and we have existing opensource implementations for Linux[2].

[1] FERGUSON, Niels. AES-CBC + Elephant diffuser: A Disk Encryption Algorithm for Windows Vista.

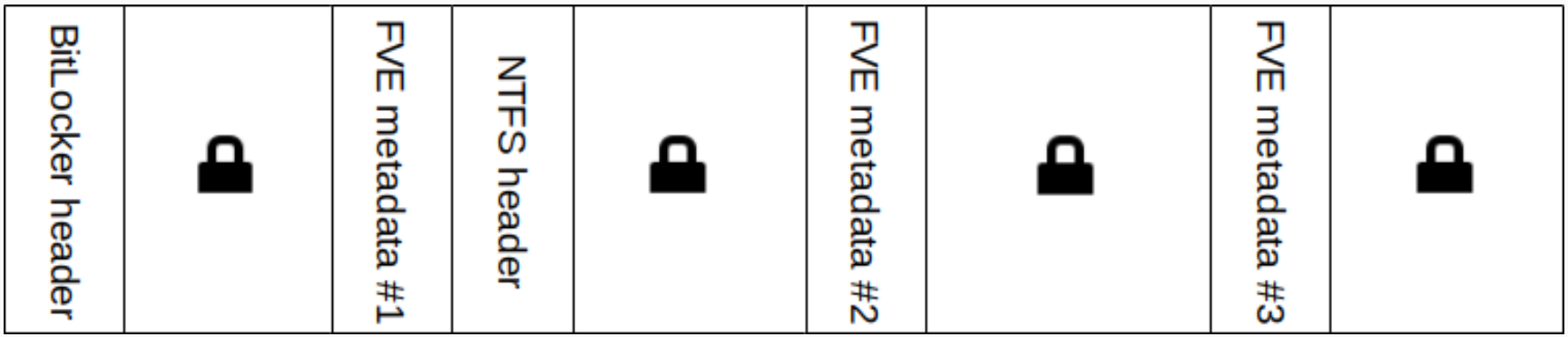
[2] Detailed description of the metadata by Joachim Metz is available in the libbde documentation.

Why BitLocker?

- There currently isn't a technology for full disk encryption that would work seamlessly, without installing additional tools, in Microsoft Windows, GNU/Linux or both.
- Existing tools for Linux are not very user-friendly and use FUSE and custom implementations of cryptographic functions.
- Ideally, BitLocker devices would be automatically recognized and presented to the user in the same way native encrypted devices are.

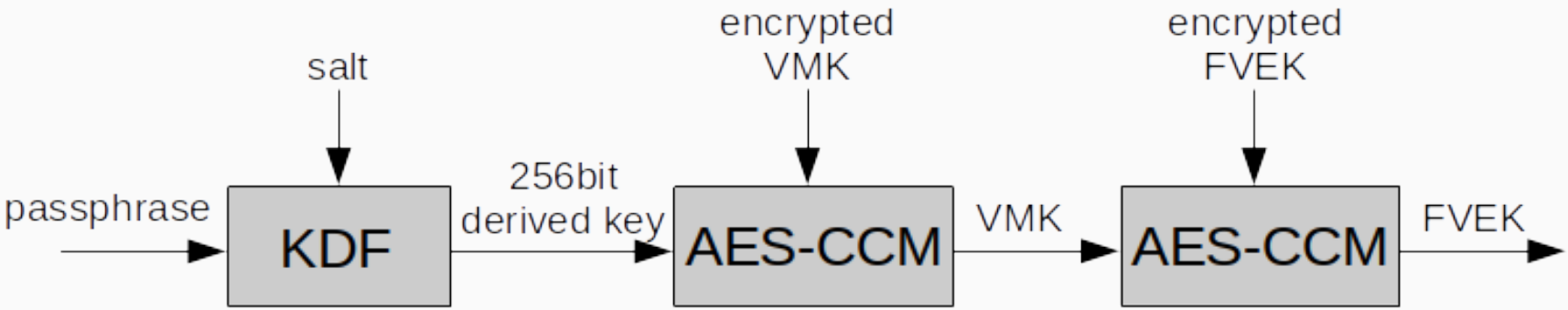
BitLocker device structure

- **Header** – format identification and FVE metadata offsets.
- **FVE metadata** – BitLocker configuration and keys.
- **NTFS header** – encrypted header for the open device.
- **Encrypted data**.



Keys

- BitLocker metadata contain two types of keys:
  - **FVEK** is a 128 or 256 bit key used for data encryption and
  - **VMK** is used to decrypt FVEK. Multiple encrypted copies of the VMK are stored in the metadata with different types of protectors.



Disk encryption on Linux

Device Mapper and LUKS

Device Mapper

- Kernel module for creating “mapped” virtual block devices.
- Can be used to “partition” disks to smaller block devices or to concatenate multiple disks to one volume.
- Multiple *targets* provide additional features that include encryption, caching, mirroring etc.

dm-crypt

- Crypt target provides transparent disk encryption.
- Data written to a dm-crypt device are encrypted with provided key and cipher specification before writing them to the underlying block device.

LUKS

Using dm-crypt directly is not very user-friendly

```
# dmsetup create x --table "0 204800 crypt aes-xts-plain64
9d3...d5c 0 /dev/sdb1 0 0"
```

LUKS

- Linux Unified Key Setup
- Defines a standardized format for storing metadata and key materials.
- Allows simple and user-friendly way of creating and managing of encrypted devices.

```
# cryptsetup luksOpen /dev/sdb1 x
Enter passphrase for /dev/sdb1: ***
```

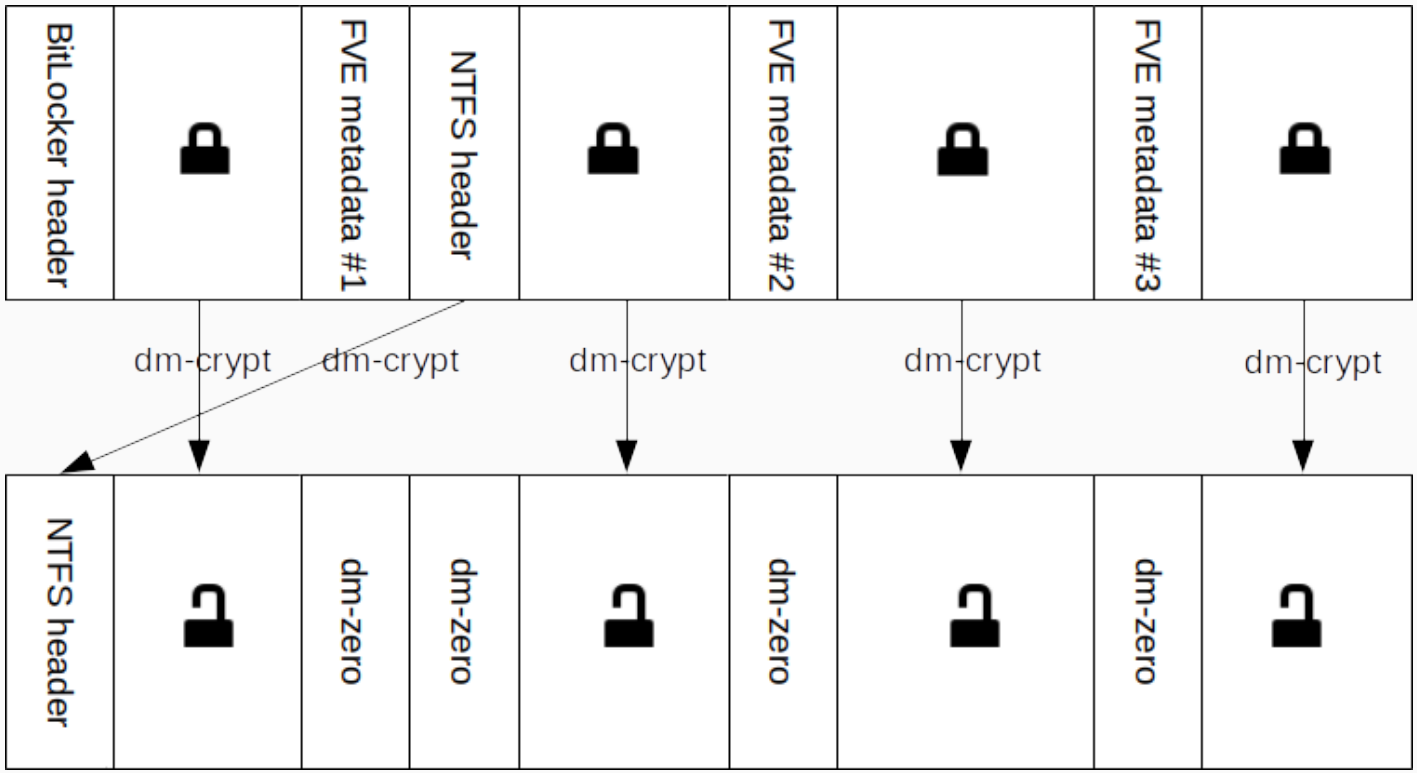
BitLocker vs. LUKS

BitLocker on Linux

BitLocker and Device Mapper

Device Mapper needs to know:

- cipher (AES-XTS for Windows 10),
- initialization vector (sector number),
- key and
- location (offset) of the encrypted data.



Using Device Mapper directly is not very user-friendly

```
# dmsetup table --showkeys
x: 0 16 crypt aes-xts-plain64 cc4...d66 68904 7:0 68904
x: 16 68760 crypt aes-xts-plain64 cc4...d66 16 7:0 16
x: 68776 128 zero
x: 68904 16 zero
x: 68920 21424 crypt aes-xts-plain64 cc4...d66 68920 7:0 68920
x: 90344 128 zero
x: 90472 22632 crypt aes-xts-plain64 cc4...d66 90472 7:0 90472
x: 113104 128 zero
x: 113232 91568 crypt aes-xts-plain64 cc4...d66 113232 7:0 \
        113232
```

BitLocker in cryptsetup

- Support for BITLK (BitLocker compatible) devices was added in cryptsetup 2.3.0[3].
- Cryptsetup can now parse BitLocker metadata, extract and decrypt (password protected) keys and construct the multi segment device mapper device.

```
# cryptsetup bitlkOpen /dev/sdb2 x
Enter passphrase for /dev/sdb2: ***
```

[5] cryptsetup 2.3.0-rc0 was released on Jan 12, 2020.

BitLocker in cryptsetup

```
# cryptsetup bitlkDump /dev/sdb2
Info for BITLK device /dev/sdb2.
Version:        2
GUID:           8f595209-f5b9-49a0-85d4-cb8f80258c27
Created:        Thu Jul  4 09:01:55 2019
Description:    DESKTOP-NPM7RCA H: 7/4/2019
Cipher name:    aes
Cipher mode:    xts-plain64
Cipher key:     128 bits

Keyslots:
 0: VMK
        GUID:           3e55195c-8811-4d9b-97b4-2b9e5f8f5384
        Protection:     VMK protected with passphrase
        Salt:           8d7637cc5d885d5ff4f748dbc8440d2e
        Key data size:  44 [bytes]
...
```

Supported features

Protectors

- *Supported:* passphrase, recovery passphrase
- *Unsupported:* TPM, smart cards, startup key...

Encryption

- *AES-XTS* (Windows 10): supported in all versions
- *AES-CBC* (Windows 7-10): Linux 5.3
- *AES-CBC + Elephant diffuser* (Windows Vista): Linux 5.6

Metadata

- Only version 2 (Windows 7+) is supported.

BitLocker in UDisks

- UDisks is a daemon for accessing and manipulating with disks and storage devices.
- It's used to mount and open removable devices in most graphical environments.
- BitLocker devices are identified by udev (using libblkid[4]).
- UDisks provides the `Encrypted` DBus interface for BitLocker devices and `Unlock` and `Lock` functions for (un)locking these devices.
- No further changes are needed in the GUI tools and daemons like GVfs to support BitLocker.
- Support for BitLocker will be available in UDisks 2.9.0.

[6] Detection of BitLocker devices was added in util-linux v2.33.

BitLocker in UDisks

```
/org/freedesktop/UDisks2/block_devices/sdb2:
  org.freedesktop.UDisks2.Block:
...
    Id:
    IdLabel:
    IdType:                          BitLocker
    IdUUID:                          1f8bf933-8323-4c97-...
    IdUsage:                         crypto
  org.freedesktop.UDisks2.Encrypted:
    ChildConfiguration:              []
    CleartextDevice:                 '/'
    HintEncryptionType:               BitLocker
```

Summary

Questions

Thank you for your attention.

Please test BITLK support in cryptsetup and report all bugs at
gitlab.com/cryptsetup/cryptsetup/issues