

Bitlocker šifrování disku v Linuxovém prostředí

Bc. Vojtěch Trefný



*** Nascanované zadání, strana 1 ***

*** Nascanované zadání, strana 2 ***

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomové práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky. Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....

podpis autora

ABSTRAKT

Text abstraktu česky

Klíčová slova: Přehled klíčových slov

ABSTRACT

Text of the abstract

Keywords: Some keywords

Zde je místo pro případné poděkování, motto, úryvky knih, básní atp.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	8
1 BITLOCKER	10
1.1 DISKOVÝ FORMÁT.....	10
1.1.1 Hlavička	10
1.1.2 FVE metadata	12
1.1.3 FVE záznamy	13
1.2 KLÍČE.....	15
1.2.1 Full Volume Encryption Key.....	15
1.2.2 Volume Master Key	16
1.3 ŠIFROVANÁ DATA	18
1.3.1 Použité šifrovací algoritmy.....	18
1.3.2 Způsob uložení data.....	18
1.3.3 Postup při dešifrování	18
2 EXISTUJÍCÍ ŘEŠENÍ PRO PRÁCI S BITLOCKEREM V LINUXU ...	20
2.1 LIBBDE	20
2.2 DISLOCKER	20
II PROJEKTOVÁ ČÁST	20
3 NADPIS	22
3.1 PODNADPIS.....	22
ZÁVĚR	23
SEZNAM POUŽITÉ LITERATURY	24
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	25
SEZNAM OBRÁZKŮ	26
SEZNAM TABULEK	27
SEZNAM PŘÍLOH	28

ÚVOD

První odstavec pod nadpisem se neodsazuje, ostatní ano (pouze první řádek, odsazení vertikální mezy odstavci je typické pro anglickou sazbu; czech babel toto respektuje, netřeba do textu přidávat jakékoliv explicitní formátování, viz ukázka sazby tohoto textu s následujícím odstavcem).

Formátování druhého odstavce. Text text text text text text text text text text text.

I. TEORETICKÁ ČÁST

1 BitLocker

text

1.1 Diskový formát

Pro samotnou práci s BitLocker zařízením v linuxovém prostředí je nejdůležitější formát, tedy způsob, jakým jsou na disku uložena data. Protože je pomocí BitLockeru možné vytvořit šifrovaný flash disk, který lze použít na jiném počítači pouze za znalosti hesla pro jeho odemčení, je zřejmé, že někde na samotném disku jsou uložena všechna potřebná metadata pro jeho odemčení v (alespoň částečně) otevřené podobě.

16	68760	128	21440	128	22632	128	91568
Hl.	Data	FVE 1	Data	FVE 2	Data	FVE 3	Data

Obr. 1.1 Zjednodušená struktura BitLocker zařízení

Na obrázku 1.1 je nastíněna zjednodušená struktura uložení dat a metadat na zařízení šifrovaném pomocí BitLockeru. Na začátku zařízení se nachází 8 KiB velká hlavička (podrobněji popsána v části 1.1.1) obsahující základní data pro jeho identifikaci a mezi zašifrovanými daty jsou uloženy tři kopie dalších metadat, každá o velikosti 64 KiB (podrobněji popsány v části 1.1.2). Čísla nad jednotlivými „částmi“ odpovídají jejich velikosti v sektorech pro testovací zařízení o velikosti 100 MiB, které bylo vytvořeno ve Windows 10.

1.1.1 Hlavička

Stejně jako u většiny diskových formátů, je i u BitLockeru na začátku disku takzvaná hlavička, která obsahuje základní informace o použitém formátu a jeho vlastnostech a také k jeho rychlé identifikaci. BitLocker hlavička zabírá celkem 512 bajtů a je u ní patrná inspirace u souborového systému NTFS. V tabulce 1.1 jsou zobrazeny jednotlivé (známé¹) položky hlavičky BitLockeru a pro srovnání také stejné položky v hlavičce souborového systému NTFS.

Struktura NTFS hlavičky je převzata z [4], struktura BitLocker hlavičky je pak částečně převzata z [3], částečně z [5] a částečně výsledkem vlastního zkoumání.

Z pohledu identifikace BitLocker zařízení je nejdůležitější částí hlavičky 8 bajtů na offsetu 3, které se u NTFS formátu nazývají *OEM název* a které slouží pro rychlou

¹Struktura formátu BitLocker není společností Microsoft nikde veřejně zcela kompletně zdokumentována, význam jednotlivých položek tedy nemusí být vždy přesně znám.

Tab. 1.1 Porování položek hlaviček BitLocker a NTFS

offset	velikost	BitLocker	NTFS
0	3	boot kód	
3	8	OEM název (signatura)	
11	2	počet bajtů na sektor	
13	1	počet sektorů na cluster	
14	2	rezervované sektory	
16	4	nepoužito	
21	1	popisek média	
22	18	nepoužito	
40	8	počet sektorů	
48	8	adresa prvního clusteru MFT	
56	8	kopie adresy prvního clusteru MFT	
64	1	velikost MFT entry	
65	3	nepoužito	
68	1	velikost indexu	
69	3	nepoužito	
72	8	NTFS serial number	
80	4	nepoužito	
84	76	boot kód	
160	16	BitLocker GUID	boot kód
176	8	offset první kopie FVE metadat	
184	8	offset druhé kopie FVE metadat	
192	8	offset třetí kopie FVE metadat	
200	310	boot kód	
510	2	signatura (0xaa55)	

identifikace zařízení. V linuxových systémech se podobné identifikátory obvykle nazývají *signatura*. Pro BitLocker formát je (u všech verzí) signatura v ASCII podobě -FVE-FS-.

Pro další práci s BitLockerem není většina položek hlavičky zajímavá. Výjimku tvoří GUID identifikátor uložený na offsetu 160 (16 bajtů dlouhý UTF-8 string) a trojice *uint32* hodnot na offsech 176, 184 a 192, které obsahují umístění (jako relativní offset od začátku zkoumaného zařízení) tří bloků FVE metadat. Všechny tyto čtyři hodnoty jsou v BitLocker hlavičce umístěny na offsech, které jsou v NTFS součástí *bootcode*.

Umístění všech výše zmíněných „důležitých“ částí BitLocker hlavičky je zobrazeno na obrázku 1.2.

TODO:
To
by
asi
chtělo
citaci.

```

00000000 eb 58 90 2d 46 56 45 2d 46 53 2d 00 02 08 00 00 |.X.-FVE-FS-....|
00000010 00 00 00 00 00 f8 00 00 3f 00 ff 00 00 28 03 00 |.....?....(|
00000020 00 00 00 00 e0 1f 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 01 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000040 80 00 29 00 00 00 00 4e 4f 20 4e 41 4d 45 20 20 |..)....NO NAME |
00000050 20 20 46 41 54 33 32 20 20 20 33 c9 8e d1 bc f4 | FAT32 3....|
00000060 7b 8e c1 8e d9 bd 00 7c a0 fb 7d b4 7d 8b f0 ac |{.....|..}.|...|
00000070 98 40 74 0c 48 74 0e b4 0e bb 07 00 cd 10 eb ef |.@t.Ht.....|
00000080 a0 fd 7d eb e6 cd 16 cd 19 00 00 00 00 00 00 00 00 |..}.....|
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000a0 3b d6 67 49 29 2e d8 4a 83 99 f6 a3 39 e3 d0 01 |;.gI)..J....9...|
000000b0 00 50 19 02 00 00 00 00 00 d0 c1 02 00 00 00 00 |.P.....|
000000c0 00 a0 73 03 00 00 00 00 00 00 00 00 00 00 00 00 |..s.....|
000000d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000100 0d 0a 52 65 6d 6f 76 65 20 64 69 73 6b 73 20 6f |..Remove disks o|
00000110 72 20 6f 74 68 65 72 20 6d 65 64 69 61 2e ff 0d |r other media...|
00000120 0a 44 69 73 6b 20 65 72 72 6f 72 ff 0d 0a 50 72 |.Disk error...Pr|
00000130 65 73 73 20 61 6e 79 20 6b 65 79 20 74 6f 20 72 |less any key to r|
00000140 65 73 74 61 72 74 0d 0a 00 00 00 00 00 00 00 00 |estart.....|
00000150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000190 00 00 00 00 00 00 00 00 78 78 78 78 78 78 78 78 |.....xxxxxxx|
000001a0 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78 |xxxxxxxxxxxxxxxx|
*
000001e0 78 78 78 78 78 78 78 78 ff ff ff ff ff ff ff ff |xxxxxxxx.....|
000001f0 ff ff ff ff ff ff ff ff ff ff ff 00 1f 2c 55 aa |.....,U.|
00000200

```

Obr. 1.2 BitLocker hlavička se zvýrazněnou signaturou,
GUID a trojicí offsetů FVE metadat

1.1.2 FVE metadata

Samotná výše popsaná hlavička formátu BitLocker neobsahuje o samotném BitLockeru téměř žádné informace. Slouží především pro rychlou identifikaci zařízení jako zařízení šifrovaného pomocí technologie BitLocker. Všechny informace potřebné pro práci s tímto zařízením, tedy především způsob uložení dat, jejich umístění, způsob jakým jsou šifrovány a hlavně klíč pro jejich (de)šifrování je uložený na třech různých místech²⁾ definovaných v hlavičce. Jedná se o tři identické kopie³⁾ takzvaných *FVE metadat*.

FVE metadata se skládají z celkem tří částí – hlavičky FVE bloku (*FVE metadata block header*), samotné FVE hlavičky (*FVE metadata header*) a různého množství

²⁾Na offsetech přibližně ve 33 %, 44 % a 55 % u testovaných BitLocker zařízení.

³⁾Tři kopie jsou zvoleny pravděpodobně jako záloha pro případ náhodného poškození metadat. Vzhledem k tomu, že bez kompletní nepoškozené kopie těchto metadat není možné data na zařízení dešifrovat, je vícenásobná záloha na místě.

FVE záznamů (*FVE metadata entry*, které obsahují samotné klíče a další důležité informace[3]⁴⁾.

Důležité položky v obou hlavičkách, jejich velikosti a offsety (vztažené vůči začátku dané hlavičky) jsou uvedeny v tabulce 1.2. Kompletní struktura obou hlaviček je součástí přílohy .

TODO:
od-
kaz
na
přílohu

Tab. 1.2 Zjednodušená struktura FVE metadat

Hlavička FVE bloku		
offset	velikost	popis
0	8	signatura (-FVE-FS-)
10	2	verze (1 nebo 2)
32	8	offset první kopie FVE metadat
40	8	offset druhé kopie FVE metadat
48	8	offset třetí kopie FVE metadat

FVE hlavička		
0	4	velikost metadat (včetně záznamů)
16	16	GUID
36	4	šifrovací algoritmus
40	8	datum a čas vytvoření

Mezi pro nás zajímavé položky v hlavičce patří její celková velikost (včetně velikosti samotné hlavičky a velikosti za ní následujících záznamů), šifrovací algoritmus použitý pro zašifrování dat uložených na disku (možné algoritmy jsou popsány v části 1.3.1) a v některých případech může být užitečný i čas vytvoření, který je uložen ve formátu FILETIME⁵⁾.

1.1.3 FVE záznamy

Za výše uvedenou hlavičkou se nechází blíže nespécifikované množství FVE záznamů. Ty slouží v podstatě jako key-value úložiště pro jakékoli další „informace“, které jsou pro práci s BitLockerem potřebné. Tím, že není třeba předem určeno, kolik takových záznamů bude za hlavičkou uloženo, je možné přidávat nové položky při zachování zpětné kompatibility⁶⁾.

⁴⁾Toto dělení zavádí Joachim Metz v [3]. Teoreticky by se daly dvě první části metadat spojit, protože na disku se nachází vždy hned za sebou, ale rozdělení dává smysl, protože první část se týká popisu samotných metadat (signatura, verze, umístění všech tří bloků), zatímco druhá část už obsahuje samotná metadata (GUID, čas vytvoření, použitý šifrovací algoritmus).

⁵⁾FILETIME je ve skutečnosti struktura sestávající ze dvou 32bit integer hodnot, které dohromady udávají počet 100 nanosekundových intervalů, které k danému datu uplynuly od 1. ledna 1601.[1]

⁶⁾Celková největší možná velikost FVE metadat je 64 KiB (alespoň tedy tolik je pro FVE metadata vyhrazeno na vytvořených BitLocker zařízeních), teoreticky je tedy možné mít až 64 KiB - 112 B metadat.

Jelikož známe celkovou velikost FVE metadat (je uvedena v hlavičce, viz tabulka 1.2) a celková velikosti hlaviček FVE metadat je pevná (64 a 48 bajtů), pro přechzení všech záznamů stačí číst data ve smyčce, dokud nedojdeme na konec metadat, nebo dokud následující záznam nemá nulovou velikost.

Struktura FVE je relativně jednoduchá a je popsána v tabulce 1.3. Důležitou součástí je velikost záznamu, protože podle svého typu může mít různou délku.

Tab. 1.3 Struktura FVE záznamu

offset	velikost	popis
0	2	velikost záznamu
2	2	typ záznamu
4	2	typ hodnoty záznamu
6	2	verze (1)
8		data

Typ a hodnota označují, co je v daném záznamu uloženo. Známé typy a hodnoty jsou popsány v tabulce 1.4. U typů se typicky jedná buď o klíč (FVEK, VMK) nebo obecnou property, hodnota pak dále specifikuje, jak je daný typ uložen (zašifrovaný klíč, unicode string).

Způsob uložení dat záleží na tom, jaká konkrétní data jsou v záznamu uložena. U „jednoduchých“ záznamů, jako je například popis je v datech uložen textový řetězec uložený v kódování UTF-16, u „složitějších“ záznamů, jako jsou například klíče, mají data vlastní strukturu včetně dalších záznamů.

Tab. 1.4 Známé typy FVE záznamů

Typy		Hodnoty	
typ	popis	typ	popis
0	property	0	smazáno
1	VMK	1	klíč
2	FVEK	2	string
7	popisek	5	AES-CCM šifrovaný klíč
15	hlavička disku ⁷⁾	6	TPM klíč
		8	VMK
		15	offset a velikost

Příklad „jednoduchého“ záznamu je uveden na obrázku 1.3, kde vidíme záznam typu *description*. Ten v podstatě obsahuje jméno počítače, na kterém bylo dané BitLocker zařízení vytvořeno a také datum vytvoření. Můžeme tedy vidět, že toto konkrétní

⁷⁾Umístění a velikost NTFS hlavičky otevřeného zařízení. Odpovídá hodnotě 15. Podrobnější informace o umístění NTFS hlavičky na šifrovaném zařízení jsou v části 1.3.2.

BitLocker zařízení bylo vytvořeno na počítači DESKTOP-NPM7RCA a to 3. února 2019. Tato informace je uložena jako standardní string v kódování UTF-16. Kromě tohoto stringu jsou pak na obrázku zvýrazněny i další údaje: velikost celého záznamu (64 bajtů), jeho typ (7 – popisek) a hodnota (2 – string) a verze (1).

```
02195070  40 00 07 00 02 00 01 00 44 00 45 00 53 00 4b 00 |@.....D.E.S.K.|
02195080  54 00 4f 00 50 00 2d 00 4e 00 50 00 4d 00 37 00 |T.O.P.-.N.P.M.7.|
02195090  52 00 43 00 41 00 20 00 47 00 3a 00 20 00 32 00 |R.C.A. .G.:. .2.|
021950a0  2f 00 33 00 2f 00 32 00 30 00 31 00 39 00 00 00 |/.3./2.0.1.9...|
```

Obr. 1.3 Příklad FVE záznamu typu „description“
(popisek)

U jednoduchého zařízení — v našem případě USB flash disku — se bude obvykle vyskytovat pouze pět záznamů a to již výše zmíněný popisek, dvojice záznamů typu VMK, jeden záznam typu FVEK (o obou více v části 1.2 a jeden záznam obsahující informace o umístění hlavičky disku (o tomto záznamu více v části 1.3.2)).

1.2 Klíče

Pravděpodobně nejdůležitější součástí BitLocker hlavičky jsou šifrovací klíče. Ve FVE hlavičce nalezneme celkem dva typy klíčů — Full Volume Encryption Key, neboli FVEK, a Volume Master Key, neboli VMK⁸⁾. Uloženy jsou v metadatových záznamech odpovídajících typů a to samozřejmě nikoli v otevřené podobě, ale zašifrované.

TODO:
link
na
část
o
odlišnosti
ve
starších
verzích

1.2.1 Full Volume Encryption Key

Full Volume Encryption Key (dále jen „FVEK“) je nejdůležitějším klíčem pro celý BitLocker. Pomocí tohoto klíče jsou totiž zašifrovaná data uložená na disku. FVEK samotný nejde změnit⁹⁾ a v případě jeho poškození nebo náhodného smazání není možné uložená data nijak dešifrovat.

FVEK je v metadatach uložen v záznamu typu *FVEK* s hodnotou *AES-CCM šifrovaný klíč* a je, jak hodnota naznačuje, zašifrován pomocí šifry AES-CCM (o této šifře a módu více v části 1.3.1), kdy je jako klíč použit VMK a jako inicializační vektor 0.

Struktura dat pro FVEK v metadatovém záznamu je popsána v tabulce 1.5. Kromě samotného klíče obsahují datum a čas jeho vytvoření a nonce.

TODO:
na-
jít
definici
a
citaci

⁸⁾Původní varianta BitLockeru má ještě jeden klíč — TWEAK, ten je podrobněji popsán v části .

⁹⁾Bez kompletního přešifrování všech dat.

¹⁰⁾Velikost šifrovaného klíče závisí na použité šifře — 12 bajtů vždy připadne na informace o klíči a 32 bajtů v tomto případě připadá na samotný klíč, jelikož je použit 128bit AES.

Tab. 1.5 Způsob uložení FVEK v metadatech

offset	velikost	popis
0	8	datum a čas vytvoření (jako FILETIME)
8	4	nonce
12	16	MAC tag
28	44 ¹⁰⁾	šifrovaný klíč

Samotná zašifrovaná část klíče obsahuje kromě samotného klíče také další data o klíči samotném — velikost, verze a šifrovací metoda použitá pro data zašifrovaná pomocí FVEK. Jejich struktura je popsána v tabulce 1.6.

Tab. 1.6 Obsah FVEK po dešifrování

offset	velikost	popis
0	4	velikost
4	4	verze (1) ¹¹⁾
8	4	šifrovací metoda
12	32	klíč

Na obrázku 1.4 je pak vidět příklad dešifrovaného FVEK. Zvýrazněny jsou jeho celková velikost (44 bajtů), verze (1), šifrovací metoda (hex kód 0x8004 v tomto případě znamená 128bit AES-XTS) a následně samotný 128bit klíč.

```

00000000  2c 00 00 00 01 00 00 00 04 80 00 00 a4 d0 11 64 |,.....d|
00000010  0c a0 df ec b2 4d a2 39 b1 4e 4a b7 62 56 f2 e3 |....M.9.NJ.bV..|
00000020  b2 27 54 40 91 21 0e 98 aa 84 5f 52          |.'T@.!...._R|

```

Obr. 1.4 Dešifrovaný FVEK

1.2.2 Volume Master Key

Jak již bylo řečeno výše, FVEK je na disku uložen zašifrován pomocí Volume Master Key (dále jen „VMK“). Ten je uložen také v metadatových záznamech v hlavičce a je také zašifrován. Narozdíl od FVEK, který je vždy uložen v metadatech v jediné kopii, VMK může být v metadatech uložen vícekrát, pokaždé „jinak“ zašifrovaný.

Tento systém umožňuje, aby byl FVEK, jakožto hlavní a nejdůležitější klíč, uložen na disku pouze v jedné kopii, ale zároveň existovala možnost, jak mít pro jedno zařízení

¹¹⁾Některé zdroje [3] uvádějí verzi pouze jako 2 bajtovou a následující 2 bajty jako „neznámé“. Vzhledem k tomu, že v jiných hlavičkách je verze v některých případech 4 bajtová a v některých 2 bajtová a že na testovacích zařízeních byly tyto dva bajty vždy nulové, domnívám se, že je pravděpodobnější, že verze je zde 4 bajtová.

více různých hesel (respektive více různých způsobů odemčení daného zařízení). Pro přidání „nového“ hesla tak teoreticky stačí jednoduše znát alespoň jedno již existující, pomocí kterého se VMK dešifruje a následně uloží zašifrovaný pomocí nového hesla. Analogicky tak lze také snadno změnit heslo — jak již bylo zmíněno výše, FVEK nejde změnit bez přešifrování celého zařízení, ale změna hesla díky tomuto systému znamená pouhé uložení nově zašifrovaného VMK.

V odstavci výše je několikrát zmíněno *heslo*, ale VMK může být chráněn více různými způsoby. Dokumentace BitLockeru [2] zmiňuje celkem deset možných typů *protektorů* klíčů v BitLockeru, které jsou zapsány v tabulce 1.7.

Tab. 1.7 Možnosti ochrany VMK

hodnota	popis
0	neznámý/jiný
1	TPM
2	externí klíč
3	číselné heslo
4	TPM a PIN
5	TPM klíč
6	TPM, PIN a klíč
7	veřejný klíč
8	heslo
9	TPM certifikát
10	CryptoAPI Next Generation (CNG)

Z pohledu této práce je nejobvyklejším protektorem právě heslo, protože použití BitLocker zařízení v linuxovém prostředí se dá předpokládat primárně u flash disků, u nichž se používá ochrana heslem¹²⁾.

Pro každé vytvořené BitLocker se kromě „primární“ ochrany (v našem případě typicky hesla) vytváří ještě jeden VMK chráněný záložním heslem. Způsob ochrany je u něj stejný jako u VMK, který je chráněný heslem, rozdíl je v tom, že heslo zadává uživatel, kdežto záložní heslo je vygenerované a uživateli je při vytváření „předáno“ v podobě souboru, který obsahuje 48 čísel.

Struktura VMK, naznačená v tabulce 1.8, je ovlivněna tím, že samotný klíč může být chráněn různými způsoby a je pro něj tedy třeba ukládat různá metadata a i samotný klíč může být třeba v některých případech ukládat v různých podobách.

První část VMK struktury je v celku běžná — obsahuje identifikátor klíče (GUID), čas vytvoření a typ ochrany. Další metadata jsou pak uložena jako záznamy, stejně jako

¹²⁾Ochrana pomocí TPM nedává u přenosných disků smysl, protože TPM čipy jsou nedělitelnou součástí hardwaru.

TODO:
citace

u samotné FVE hlavičky (podrobněji v části 1.1.3 a tabulce 1.3).

Tab. 1.8 Struktura VMK

offset	velikost	popis
0	16	GUID
16	9	datum a čas vytvoření
24	2	neznámé
26	2	typ ochrany
28		metadatové záznamy

Kompletní VMK klíč chráněný záložním heslem je zobrazen na obrázku 1.5. Zvýrazněno je GUID, typ ochrany (8 — heslo) a dva „připojené“ záznamy, oba typu property, první obsahující sůl potřebnou pro odvození klíče potřebného pro dešifrování VMK ze záložního hesla (funkcionalita odvození klíče z hesla je popsána v části) a druhá obsahující samotný klíč (textový výpis je debugovacím výstupem z nástroje vytvořeného v rámci praktické části).

TODO:
od-
kaz
na
část
popisu-
jící
KDF

1.3 Šifrovaná data

1.3.1 Použité šifrovací algoritmy

1.3.2 Způsob uložení data

1.3.3 Postup při dešifrování

```

00000000 c1 56 2e 01 d6 4e 27 45 8a bf 7a 9f 29 e0 b5 21 |.V...N'E..z.)...!|
00000010 40 c5 cd 54 a0 bb d4 01 00 00 00 08 ac 00 00 00 |@..T.....|
00000020 03 00 01 00 00 10 00 00 46 ee b7 10 0e 43 4d d4 |.....F....CM.|
00000030 f1 84 a5 ab eb c6 21 f4 40 00 12 00 05 00 01 00 |....!...@.....|
00000040 40 7d e5 52 a0 bb d4 01 04 00 00 00 72 b0 71 f4 |@}.R.....r.q.|
00000050 20 9e c9 8e b7 1b 5e 42 71 b5 bc 21 c6 57 9b 29 | .....^Bq...!W.)|
00000060 56 2c 92 ad db d7 73 75 a9 78 c2 94 c5 a5 07 d1 |V,....su.x.....|
00000070 62 61 0c 56 d8 ca 9d ac 50 00 13 00 05 00 01 00 |ba.V....P.....|
00000080 40 7d e5 52 a0 bb d4 01 05 00 00 00 3e d9 ac 58 |@}.R.....>..X|
00000090 e6 86 ba ac 05 48 ea 0b 64 ee 77 7a b4 77 ba cb |....H..d.wz.w..|
000000a0 c0 83 83 b0 7b ab 52 c7 0d 9e 8f 62 d7 cb a3 90 |....{.R....b....|
000000b0 cc b8 8e 39 a4 be 8a 0a 5c 16 86 62 c9 64 81 4d |...9....\..b.d.M|
000000c0 91 9d 27 24 3a 8e a3 7c 50 00 00 00 05 00 01 00 |..'...|P.....|
000000d0 40 7d e5 52 a0 bb d4 01 06 00 00 00 97 18 2f d6 |@}.R...../.|
000000e0 83 de e7 63 0a fa 57 48 44 2b 66 90 91 a0 ad e9 |...c..WHD+f.....|
000000f0 0c 08 e8 1e 3d 2f 7d 3b cc 9f ba e4 ed b5 6b c2 |....=/};.....k.|
00000100 e1 a4 53 cf c5 60 2a 92 2d c8 1d 85 10 b7 99 87 |..S..'*.-----|
00000110 9d 1d 1e 36 46 40 6b e7 |...6F@k. |

```

VMK

```

Identifier: 012e56c1-4ed6-4527-8abf-7a9f29e0b521
Type: VMK protected with recovery password
Salt: 46 ee b7 10 0e 43 4d d4 a5 ab eb c6 21 f4 f1 84
AES-CCM encrypted key
  Nonce data: 2019-02-03 09:10:36.052000
  Nonce counter: 6
  Key: 91 a0 ad e9 0c 08 ... 1d 1e 36 46 40 6b e7

```

Obr. 1.5 VMK chráněný záložním heslem

2 Existující řešení pro práci s BitLockerem v Linuxu

2.1 libbde

2.2 Dislocker

II. PROJEKTOVÁ ČÁST

3 Nadpis

3.1 Podnadpis

ZÁVĚR

Text závěru

SEZNAM POUŽITÉ LITERATURY

- [1] Programming reference for Windows API. Dostupné z: <https://docs.microsoft.com/en-us/windows/desktop/api/minwinbase/ns-minwinbase-filetime>
- [2] Security WMI Providers. Dostupné z: <https://docs.microsoft.com/en-us/windows/desktop/secprov/bitlocker-drive-encryption-provider>
- [3] Library and tools to access the BitLocker Drive Encryption (BDE) encrypted volumes. 2018. Dostupné z: <https://github.com/libyal/libbde>
- [4] Carrier, B.: *File system forensic analysis*. London: Addison-Wesley, první vydání, 2005, ISBN 978-0321268174.
- [5] Ferguson, N.: AES-CBC + Elephant diffuser. 2006.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
CNG	CryptoAPI Next Generation
FS	File System
FVE	Full Volume Encryption
FVEK	Full Volume Encryption Key
GUID	Globally Unique Identifier
MAC	Message Authentication Code
MFT	Master File Table
NTFS	New Technology File System
OEM	Original Equipment Manufacturer
PIN	Personal Identification Number
TPM	Trusted Platform Module
USB	Universal Serial Bus
UTF	Unicode Transformation Format
VMK	Volume Master Key
XEX	Xor-Encrypt-Xor
XTS	XEX-based Tweaked-codebook with Ciphertext Stealing

SEZNAM OBRÁZKŮ

Obr. 1.1	Zjednodušená struktura BitLocker zařízení	10
Obr. 1.2	BitLocker hlavička se zvýrazněnou signaturou, GUID a trojicí offsetů FVE metadat	12
Obr. 1.3	Příklad FVE záznamu typu „description“ (popisek)	15
Obr. 1.4	Dešifrovaný FVEK	16
Obr. 1.5	VMK chráněný záložním heslem	19

SEZNAM TABULEK

Tab. 1.1	Porování položek hlaviček BitLocker a NTFS	11
Tab. 1.2	Zjednodušená struktura FVE metadat	13
Tab. 1.3	Struktura FVE záznamu	14
Tab. 1.4	Znamé typy FVE záznamů	14
Tab. 1.5	Způsob uložení FVEK v metadatech	16
Tab. 1.6	Obsah FVEK po dešifrování	16
Tab. 1.7	Možnosti ochrany VMK	17
Tab. 1.8	Struktura VMK	18

SEZNAM PŘÍLOH

P I.	Název přílohy
------	---------------

PŘÍLOHA P I. NÁZEV PŘÍLOHY

Obsah přílohy