A Deep Exploration of BitLocker Encryption and Security Analysis

Cheng Tan
Science and Technology on Communication Security Laboratory
Chengdu, Sichuan Province, China
e-mail: ctan2007@163.com

Lijun Zhang
Science and Technology on Communication Security Laboratory
Chengdu, Sichuan Province, China
e-mail: 41049250@qq.com

Liang Bao*
Key Lab of Information Network Security, Ministry of Public Security
Shanghai, China
e-mail: 1005886234@qq.com
*Corresponding Author

***Abstract*—Due to the popularity of Windows system, BitLocker is widely used as a built-in disk encryption tool. As a commercial application, the design of BitLocker has to consider a capability of disaster recovery, which helps a user to recover data stored on encrypted disk when a regular access is not available. In this case, it will inevitably lead to some security risks when using BitLocker. We have a deep exploration of BitLocker encryption mechanism in this paper. We present the decryption method of encrypted VMK in case of system partition encryption and non-system partition encryption, respectively. VMK is the core key in BitLocker, with which the encrypted partition or the entire disk can be further decrypted. As for security analysis on BitLocker, we firstly make a difficulty analysis of brute force cracking on BitLocker keys, and then we analyze a possible threat caused by key theft. Based on this, we propose a few countermeasures about BitLocker usage. Additionally, we give some suggestions about security enhancement of BitLocker encryption.**

Keywords-BitLocker encryption; VMK; external key; recovery key; unlock password; security analysis

I. INTRODUCTION

Most Windows users restrict others from using their personal computers (PC) by setting login passwords. An attacker can easily access various types of data stored on PC even if he has no login password. Due to the unencrypted hard disk on PC, all data appears as plaintext, which seems dangerous from the perspective of personal information security. It is an effective method to prevent others from illegally acquiring personal data to encrypt a PC hard disk.

BitLocker Driver Encryption [1], which is known as BitLocker, first appeared in Windows Vista as a newly added data protection module. A user can use BitLocker to protect personal data by system partition encryption, data partition encryption or removable storage medium encryption. With BitLocker, users are immune to threat of data leakage caused by loss, theft, or improper elimination of PC hard disk.

Researchers have proposed some attack methods to obtain key information of BitLocker encryption during past years. A kind of firewire-based authentication bypassing attack under BitLocker in Windows 7 is proposed by Benjamin [2]. In fact, only when basic mode of BitLocker operation is configured can attacker log on Windows system. That is to say, BitLocker key must have been loaded into memory before starting the attack. Cold boot attack [3] is an advanced physical attack method， by which BitLocker key can be obtained from memory. However, if a PC deployed with BitLocker is powered off， any kind of cold boot attack is helpless. BitLocker is designed to prevent offline attacks, that is, offering offline security for users. However, most of current attack methods need to ensure that BitLocker volume is mounted. Offline attacks on BitLocker mainly rely on brute force cracking of key or unlock password, and there seems to be no more feasible method about offline attack.

In this paper, we firstly present BitLocker encryption mechanism and application modes of various keys used in BitLocker. Then we will analyze the decryption process to recover Volume Master Key (VMK) in case of system partition encryption and non-system partition encryption, respectively. Finally, we will give a comprehensive security analysis on BitLocker encryption.

II. OVERVIEW OF BITLOCKER ENCRYPTION MECHANISM

In order to better understand BitLocker encryption mechanism, we introduce the application modes of different BitLocker keys in this section.

A. Full Volume Encryption Key

BitLocker uses AES (Advanced Encryption Standard) as its encryption algorithm. Full Volume Encryption Key (FVEK) is the key used for encrypting partition of a disk or the entire disk, the size of which is 64 bytes. When AES-128 is selected, the first 16 bytes in FVEK will be used as the AES encryption key, and the 33-48th byte will be used as the sector key [4]. Similarly, if AES-256 is selected, the first 32 bytes in FVEK will be used as the AES encryption key, and the remaining bytes will be used as the sector key.

Note that BitLocker encryption mechanism may be slightly different in different versions of Windows. In Windows Vista or Window 7, AES-CBC and Elephant diffuser algorithm are used [5]. Since from Windows 8, Elephant diffuser algorithm has been abandoned. In Windows 10 (1511 and later versions), AES-XTS [6] is added, and the encryption algorithm can be set separately for system partition encryption and non-system partition encryption (including data partition encryption and removable storage medium encryption) [7].

978-1-7281-8141-7/20/$31.00 ©2020 IEEE

B. *Volume Master Key*

In order to prevent an unauthorized access, FVEK is stored encrypted on disk. FVEK uses VMK of 256 bits as its encryption key, and the encryption mode is AES-CCM [8]. Therefore, once VMK is obtained, FVEK can be decrypted, and then the entire disk or partition can be decrypted.

In fact, VMK is also stored encrypted on disk. However, multiple encrypted copies of VMK are stored, each of which uses another kind of key to encrypt the VMK. These related keys include external key, TPM key, unlock password, plain key, recovery key, etc. Each key represents a kind of access way of BitLocker volume. Here we briefly introduce the application modes of these keys:

1) external key: In case of system partition encryption, the external key is stored on a USB flash drive, which is used for being inserted into a PC to start the Windows system (or return from hibernation).

2) TPM key: For system partition encryption, it is recommended to use TPM for encryption, where RSA is used for encrypting VMK. When system starting, BitLocker firstly checks the integrity of boot components, and then TPM uses its private key to decrypt the encrypted VMK.

3) unlock password: For non-system partition encryption, a BitLocker user is required to set a password to gain access permission to BitLocker volume. In addition, the password length must not be less than 8 bytes.

4) plain key: When BitLocker is disabled, a 256-bit plain key will be generated, and a copy of VMK is to be encrypted by this key. The plain key and the encrypted VMK copy are together stored in the BitLocker metadata block. Like accessing an unencrypted disk, Windows system can decrypt VMK and FVEK without any manual operation. Note that data stored on BitLocker volume is still encrypted.

5) recovery key: For system partition encryption, when the external key or TPM key is not available, BitLocker user can access Windows system by entering the recovery key. For non-system partition encryption, if BitLocker user has forgotten his unlock password, the recovery key provides another option for gaining access permission. Recovery key is included in a .txt file and consists of 48 digits.

III. Decryption Process of VMK in System Partition Encryption

For system partition encryption, VMK is decrypted by TPM key, external key or recovery key. In order to show the decryption process intuitively, we only use the external key and the recovery key to perform decryption.

A. *VMK Decryption through External Key*

When BitLocker is configured to use an external key to encrypt VMK, an additional USB flash drive will be necessary for storing the external key. Before starting，the system will search and read a .bek file on the USB flash drive, which contains the external key. Figure 1 shows the encrypted VMK entry corresponding to the external key, and contents of some fields are shown in Table I. It can be seen from Figure 1 and Table I that the size of encrypted VMK field is 44 bytes, and AES-CCM algorithm is used.

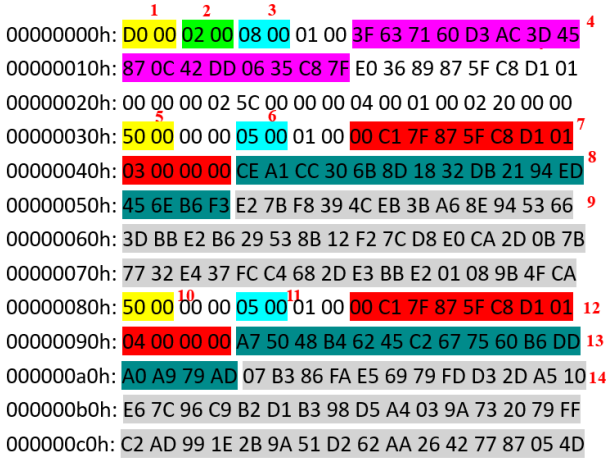

Figure 1. VMK entry corresponding to external key.

TABLE I. Contents of Fields Marked in Figure 1

Field Number	Size in Bytes	Content
1	2	Size of VMK entry
2	2	Entry type
3	2	Datum type is VMK
4	16	GUID
5	2	Size of subentry 1，in which external key is encrypted by VMK
6	2	Datum type is AES-CCM
7	12	Nonce in subentry 1, the first 8 bytes mean file time, and the remaining 4 bytes mean a counter
8	16	MAC value in subentry 1
9	44	External key encrypted by VMK
10	2	Size of subentry 2，in which VMK is encrypted by external key
11	2	Datum type is AES-CCM
12	12	Nonce in subentry 2
13	16	MAC value in subentry 2
14	44	VMK encrypted by external key

The structure of external key entry is similar to that of VMK entry. Figure 2 shows the .bek file content in hexadecimal mode. Obviously, Figure 1 and Figure 2 have the same GUID (Globally Unique Identifier), indicating that the external key in Figure 2 matches the VMK in Figure 1.

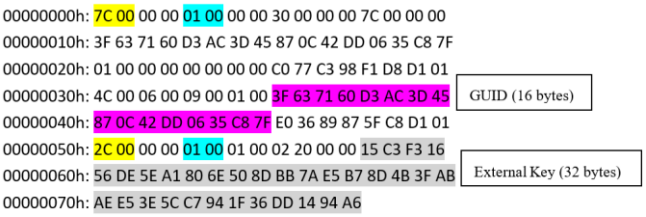

Figure 2. External key entry in .bek file.

Next, we start to decrypt the encrypted VMK with external key. As is shown in Figure 3, the 32-byte external key, the 44-byte encrypted VMK, the 12-byte nonce and the 16-byte encrypted MAC participate in decryption. A_0, A_1, A_2 and A_3 are used to generate the key stream through the external key under AES-CTR algorithm. Particularly, readers can learn how to build A_0, A_1, A_2, A_3 and B_0 from CCM protocol [8], including the flag bytes. The key stream consists of S_0, S_1, S_2 and S_3. S_1, S_2 and S_3 are concatenated to XOR with the 44-byte encrypted VMK, and the we can get B_1, B_2 and B_3, i.e., raw VMK. Then we calculate a 16-byte MAC from B_0 to B_3 under AES-CBC algorithm. After being XORed with S_0, the MAC is consistent with that in Figure 1 marked with field number of 13, which implies a correct decryption.

External Key

15 C3 F3 16 56 DE 5E A1 80 6E 50 8D BB 7A E5 B7
8D 4B 3F AB AE E5 3E 5C C7 94 1F 36 DD 14 94 A6

Encrypted VMK

07 B3 86 FA E5 69 79 FD D3 2D A5 10 E6 7C 96 C9
B2 D1 B3 98 D5 A4 03 9A 73 20 79 FF C2 AD 99 1E
2B 9A 51 D2 62 AA 26 42 77 87 05 4D

Nonce

00 C1 7F 87 5F C8 D1 01 04 00 00 00

Encrypted MAC

A7 50 48 B4 62 45 C2 67 75 60 B6 DD A0 9A 79 AD

Decrypt and Verify

00000010(00000L')

A_0: 02 00 C1 7F 87 5F C8 D1 01 04 00 00 00 00 00 00
A_1: 02 00 C1 7F 87 5F C8 D1 01 04 00 00 00 00 00 01
A_2: 02 00 C1 7F 87 5F C8 D1 01 04 00 00 00 00 00 02
A_3: 02 00 C1 7F 87 5F C8 D1 01 04 00 00 00 00 00 03
S_0: 0F BE D4 61 7B 33 A1 CA C8 B9 DF A0 6A A4 B0 D3
S_1: 2B B3 86 FA E4 69 79 FD D0 0D A5 10 1F D5 D0 88
S_2: BF B0 A7 B5 00 EA CA EB 57 CC EE A1 29 75 B1 7C
S_3: 64 54 26 08 0A B7 7E 3F 68 67 2D 2B

00111010(00M'L')

B_0: 3A 00 C1 7F 87 5F C8 D1 01 04 00 00 00 00 00 2C
B_1: 2C 00 00 00 01 00 00 00 03 20 00 00 F9 A9 46 41
B_2: 0D 61 14 2D D5 4E C9 71 24 EC 97 5E EB D8 28 82
B_3: 4F CE 77 DA 68 1D 58 7D 1F E0 28 66
MAC: A8 EE 9C D5 19 76 63 AD BD D9 69 7D CA 0D C9 7E

Figure 3. Blocks of data in VMK Decryption.

B. VMK Decryption through Recovery Key

Figure 4 shows the BitLocker recovery key in a .txt file. Obviously, the recovery key is stored as a 48-digit password. There are 8 groups in this 48-digit password, and each group consists of 6 digits. In each group, the 6-digit number is constrained by:

- The 6-digit number should be divisible by 11;
- The 6-digit number should be smaller than 720896;
- If the 6 digits in a group are represented as a, b, c, d, e and f, respectively, then we have f = (a – b + c – d + e) mod 11.

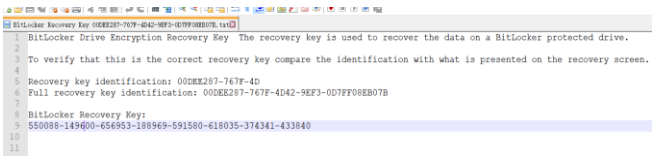

Figure 4. BitLocker recovery key in a .txt file.

Divide these 6-digit numbers of recovery key in Figure 4 by 11 respectively, and convert them to be hexadecimal. Then we get a transformed recovery key a size of 16 bytes, that is, 58C3 2035 4BE9 1B43 14D2 79DB EF84 109A.

Figure 5 shows the encrypted VMK key entry corresponding to the recovery key. Actually, the VMK entry is encrypted by an intermediate key. We display the process of generating an intermediate key in Figure 6. Firstly, we take the sha256 hash of the 16-byte recovery key as the input of BITLOCKER_HASH->hash_initial. Together with the 16-byte salt in Figure 5, we perform 0x100000 iterations of sha256 on BITLOCKER_HASH. In each iteration, the 32-byte hash result is saved as BITLOCKER_HASH->hash_update for next iteration, and BITLOCKER_HASH->iterations increases by 1. The sha256 hash in last iteration will be outputted as the intermediate key.

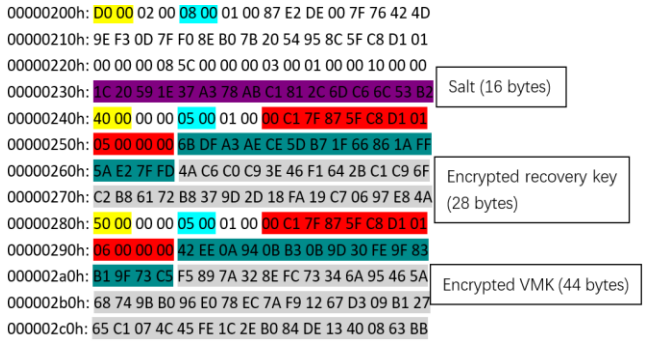

Figure 5. VMK entry corresponding to recovery key.

According to Figure 6, we calculate that the 32-byte intermediate key is 213C D303 2E4B 5FAD AAE7 01FB E390 EB9E A4F6 904D 1525 9CD3 7814 267D BAF7 DD97. With this intermediate key, we decrypt to get a same VMK as that in 3.A, which reflects a correct decryption method. Referring to VMK decryption with external key, interested readers can verify the related decryption by themselves.

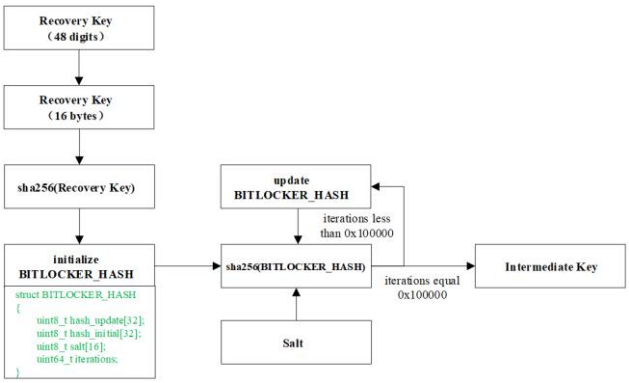

Figure 6. Generation of intermediate key.

IV. Decryption Process of VMK in Non-system Encryption

For non-system partition encryption, VMK can be decrypted through recovery key or unlock password. Since the process of decrypting VMK by recovery key in non-system partition encryption is very similar to that in system partition encryption, here we just demonstrate how to decrypt VMK with unlock password.

The process of generating password key is displayed in Figure 6. Firstly, the format of unlock password string is to be converted from utf-8 to utf-16. Then we take sha256 hash on this password string twice, and the result is used as the input of BITLOCKER_HASH->hash_initial. The remaining steps are the same as those in Figure 6. With the generated password key, we can decrypt to get VMK from the encrypted VMK entry corresponding to the unlock password. In addition, the VMK decryption method is same as that in 3.A.

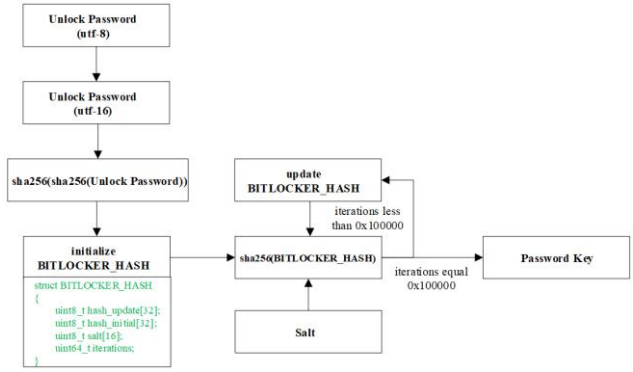

Figure 7. Generation of password key.

V. Security Analysis of BitLocker Encryption and Usage Precaution

In this section, we mainly analyze the security of BitLocker encryption mechanism. Furthermore, we give some suggestions about usage and security enhancement of BitLocker.

A. Resistance of Brute Force Cracking

The difficulty analysis of brute force cracking on BitLocker keys is showed in Table II. For system partition encryption, it is almost impossible to crack external key or recovery key directly through brute force method by now. For non-system partition encryption, it is relatively more feasible to crack unlock password by building a set of password candidates. As BitLocker requires that the minimum length of unlock password is only 8 bytes, there is a risk of many weak passwords being cracked. If the unlock password used is composed by less than 12 ASCII characters or the password structure is too simple, an experienced attacker may crack it out within a couple of days.

Hence, it is the most feasible way to decrypt VMK through password cracking from the perspective of brute force attacking. There is a lot of research on password guessing technology currently, and development of this technology has promoted the motivation of brute force cracking on password[9-11]. On one hand, a password dictionary is formed by collecting various passwords, which can be directly used for password guessing. Further, each password in a dictionary can be transformed into multiple other passwords according to certain deformation rules. On the other hand, by training on some representative passwords and mining composition structures of these passwords, a new password dictionary based on target personal information or a set of hot words will be derived. Additionally, by accelerating password cracking algorithms on FPGA, GPU, ASIC and other chips, brute force cracking becomes more feasible.

In summary, it is recommended that BitLocker users try to use full disk encryption to protect personal data on PC, so that attackers have no way to perform brute force cracking on unlock password. Secondly, if a BitLocker user is reluctant to accept full disk encryption, or BitLocker is just used for removable storage medium encryption, the unlock password should be set with a size of more than 14 bytes. A perfect password should completely contain digits, uppercase letters, lowercase letters and special characters. If supported by input method application, it is better to use a few non-ASCII characters in a password. In addition， some phrases regarding personal information and common combinations of keyboard characters should not appear in a perfect password. Finally, BitLocker unlock password should be reset at least once a month, and a new password should be quite different from the old one.

TABLE II. Difficulty Analysis of Brute Force Cracking on BitLocker Keys

Key type	Volume type	Space for brute force	Implementation feasibility
External key	System partition	2^{256}	Very difficult
Recovery key	System/Non-system partition	2^{128}	Difficult
Unlock password	Non-system partition	Depend on number of password candidates	Less difficult

B. Key Theft

Although BitLocker encrypted volume can be cracked by brute force method, the cracking work usually takes too long to be acceptable. An attacker prefers to decrypt BitLocker encrypted volume through key theft. The first way of key theft is to obtain FVEK key from memory dump file or hibernation file, and some professional forensic softwares [12, 13] can implement this function. The second way is to search a .txt file containing the recovery key. Generally, the .txt file may be stored locally on PC, on a personal USB flash drive, in personal Microsoft account, or in an email, etc. In many cases, attackers just try to find the recovery key through some special methods like file recovery technology. The third way is to search a .bek file containing the external key, provided that system partition is encrypted and external key is used. The external key is usually stored on a USB flash drive.

In order to prevent key theft, BitLocker user can disable virtual memory function and hibernation function, and shut the PC down when it is not in use. Secondly, never save the recovery key on an unencrypted partition, but in an absolutely private place. If PC is equipped with a TPM module, use TPM+PIN mode to encrypt the entire disk. If no TPM module is available and external key is used, ensure that the USB flash drive with the external key stored cannot be obtained by an attacker.

C. Compared to VeraCrypt

VeraCrypt [14] is a popular open source disk encryption software, which is an enhanced version of TrueCrypt [15]. VeraCrypt has a significant security improvement based on TrueCrypt, which increases the complexity of password cracking by 10 to about 300 times. Compared to VeraCrypt, BitLocker has its own advantages, but there are some novel ideas in VeraCrypt that can be referred to:

- Use keyfile mechanism [14, 15] to enhance security of unlock password.
- Use a hidden volume to introduce concealment on encrypted data.
- Provide a variety of optional hash functions for key derivation to introduce an unknown parameter for brute force cracking.
- Provide an option of setting the number of iterations of hash function in key derivation algorithm, and the number must not be less than a certain threshold. This also introduces an unknown parameter for brute force cracking.

VI. CONCLUSION

In this paper, we mainly focus on clarifying and analyzing BitLocker encryption mechanism. First, we have introduced the application modes of FVEK, Volume Master Key (VMK) and other related keys in BitLocker. Then we elaborated the decryption process of VMK in case of system partition encryption and non-system partition encryption. FVEK can be further decrypted with VMK, so that the encrypted data on disk can be decrypted. Finally, we comprehensively analyzed the security of BitLocker from several aspects, and gave some suggestions about usage and security enhancement of BitLocker.

ACKNOWLEDGMENT

This work is supported by project of Key Lab of Information Network Security, Ministry of Public Security, and Sichuan Province Application Fundamental Research Project (No. 2018JY0102).

REFERENCES

[1] BitLocker, https://docs.microsoft.com/powershell/module/bitlocker

[2] Böck B, Austria S B. Firewire-based physical security attacks on windows 7, efs and bitlocker[J]. Secure Business Austria Research Lab, 2009.

[3] Halderman J A, Schoen S D, Heninger N, et al. Lest we remember: cold-boot attacks on encryption keys[J]. Communications of the ACM, 2009, 52(5): 91-98.

[4] Kornblum J D. Implementing bitlocker drive encryption for forensic analysis[J]. Digital Investigation, 2009, 5(3-4): 75-84.

[5] Ferguson N. AES-CBC+ Elephant diffuser: A disk encryption algorithm for Windows Vista[J]. 2006.

[6] Martin L. XTS: A mode of AES for encrypting hard disks[J]. IEEE Security & Privacy, 2010, 8(3): 68-69.

[7] Bitlocker: AES-XTS (new encryption type), https://docs.microsoft.com/archive/blogs/dubaisec/bitlocker-aes-xts-new-encryption-type

[8] Whiting D, Housley R, Ferguson N. RFC3610: Counter with CBC-MAC (CCM)[J]. 2003.

[9] Hitaj B, Gasti P, Ateniese G, et al. Passgan: A deep learning approach for password guessing[C]//International Conference on Applied Cryptography and Network Security. Springer, Cham, 2019: 217-237.

[10] Wang D, Zhang Z, Wang P, et al. Targeted online password guessing: An underestimated threat[C]//Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016: 1242-1254.

[11] Xia Z, Yi P, Liu Y, et al. GENPass: a multi-source deep learning model for password guessing[J]. IEEE Transactions on Multimedia, 2019, 22(5): 1323-1332.

[12] Elcomsoft Forensic Disk Decryptor, https://www.elcomsoft.com/efdd.html

[13] Passware Kit Business, https://www.passware.com/kit-business

[14] VeraCrypt, https://www.veracrypt.fr/en/Home.html

[15] TrueCrypt, http://truecrypt.sourceforge.net