IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-9, Sept. 2018 E-ISSN: 2347-2693

Towards Enablement Of Efficient Forensics Of Encrypted Storage Devices Such As HDDs and SSDs

Jay Parag Mehta[1], Digvijaysinh Rathod[2*]

[1, 2]Institute of Forensic Science, Gujarat Forensic Sciences University, Gandhinagar, India

[*]*Corresponding Author: digvijay.rathod@gfsu.edu.in*

Available online at: www.ijcseonline.org

Accepted: 21/Sep/2018, Published: 30/Sept./2018

***Abstract*—** Today, encryption is considered as a basic security measure to ensure protection of sensitive data contained within storage devices from external physical threats (such as people on-site) as well as network threats (such as malicious users over the internet or intranet). Today, since encryption techniques are freely and commercially available at ease to computer users all over the world, they have far reaching effects when utilized by malicious users to hide their data for the purpose of avoiding to get caught by lawful authorities. This research work essentially takes the case of encrypted disks/volumes that could cause problems in digital forensic investigations, since they provide criminal suspects with a range of opportunities for deceptive anti-forensics and a countermeasure to legislation written to force suspects to reveal decryption keys. This research work also covers techniques using which decryption keys could be found out so that encrypted data could be obtained in decrypted form to uncover artifacts of evidentiary value. This could also help the lawful authorities to bring cyber-criminals to justice and aid digital forensic analysts with a technique in their hands for retrieving data out of encrypted storage devices especially HDDs and SSDs.

***Keywords*—**Attack vector, BitLocker, Decryption, Disk, Encryption, Forensics, Hackers, HDD, Lawful, Malicious, SSD, Volume

I. INTRODUCTION

Disk/volume encryption ensures that files are always stored on the storage devices such as HDDs/SSDs in an encrypted form. Files only become available to the operating system and applications in readable form while the system is running and unlocked by a trusted user. An unauthorized person looking at the disk/volume contents directly, would only find garbled random-looking data instead of actual files. For example, this could prevent unauthorized viewing of data when computer or storage device is located in a place to which non-trusted people might gain access while the actual user is away, gets lost or stolen (in case of portable devices), being serviced in care center, discarded after its end-of-life, etc.

In addition, disk/volume encryption could also be used to add some security against unauthorized attempts to tamper with the operating system (e.g., installation of key-loggers or any type of malware by attackers who could gain physical access to a system while its actual user is away. A very strong disk encryption setup (e.g., full-disk encryption with authenticity checking and no plain-text boot partition) could stand a chance against professional attackers who are able to tamper with system before its actual user using it.

This research work could be classified under the domain of "Digital Forensics" and within that, the classification could be further taken down to the sub-domain of "Application of Cryptographic Fundamentals". The reason for such kind of a classification is because data retrieval comes under the domain of forensics which forms the core idea for this research work but, since encryption/decryption processes are at the primary focus of this research work, and these processes depend on the fundamentals of cryptography, hence the further classification could be deemed justified.

Rest of the paper is organized as follows, Section II contains brief summary about various types of encryption used on disks/volumes, Section III speaks about the motivation behind this research work, Section IV contains literature survey performed to garner understanding of BitLocker Drive Encryption Technology, Section V explains the relevant work done by various field researchers, Section VI provides more insight into the proposed work and problem statement, Section VII states the research work objective, Section VIII informs the scope for this research work, Section IX describes the assumptions in this research work, Section X states the constraints for this research work, Section XI explains the methodology used for this research work, Section XII describe the implementation details along

with expected scenarios, Section XIII presents the results and finally Section XIV concludes research work with future directions.

II. BACKGROUND

The basic types of encryption used on disks/volumes are:

II.I Full-Disk Encryption (FDE): When the entire drive is encrypted, you have good defence against data loss due to theft since you don't need to worry about whether or not a given file is encrypted. On the other hand, when a system with FDE is in active use, i.e., when one is logged in, the entire contents of the disk are exposed to theft of data by various means.

II.II Volume-Level Encryption: Here, one creates a "volume" - a virtual directory space - whose contents are always encrypted. By only mounting the encrypted volume when you need to work with the sensitive material it contains, you reduce the risk of data loss due to various means, as well as enjoy good protection if the computer is stolen. On the other hand, you need to take care that sensitive data is always kept in an encrypted volume.

II.III File-Level Encryption: Encryption at the level of individual files affords a good level of protection against data loss due to various reasons. On the other hand, this is where you need to invest the most effort in file management. (Note that there are two flavours of file-level encryption: 1. File is decrypted only when it is in use, typically the case with application-based encryption. 2. File isn't automatically re-encrypted after one is done viewing or editing it, as that happens to be the case with stand-alone encryption utilities.).

II.IV Hybrid Encryption (combination of file-level encryption and volume-level encryption strategies): This type of encryption actually falls in the category of file-level encryption but since it employs the principle of encrypting data considering the file as a volume is the reason why it needs to be referenced in a separate category altogether. This type of encryption employs encryption principles from file-level encryption and volume-level encryption techniques, i.e., a "container" file is created on any volume such that, that container file could contain multiple files within it. All the files present on that volume would remain unencrypted but only the files present within the container on that volume would remain encrypted.

III. MOTIVATION

Encryption technologies are designed to protect data. Full-disk encryption (FDE) is used to secure data at rest. Increasingly, companies are turning to FDE as a solution for protecting mobile data. Perhaps the most common use case for FDE is protecting any organization's portable devices. From what is seen, many companies have requirements that at least some systems be encrypted - especially those that are used by individuals that might have access to sensitive and/or proprietary information or those who are always on the move for business purposes. Taking things a step further, an increasing number of organizations are moving towards a model where all of their corporate owned portable devices are encrypted by default. Some organizations also deploy FDE on desktop systems, but this application is far less common. Due to the ease at which a hard-disk drive could be removed from a standard business desktop and the lack of physical security that would prevent a drive from being removed from a facility, there are certainly strong arguments for deploying FDE on a wider scale. When full-disk encryption is deployed, it is assumed to be a black-box panacea for all data security issues. All too often, the simple presence of FDE is considered protection enough to secure a system.

This research work examines the difficulties encountered due to encrypted disks and volumes during digital forensic investigations. Now-a-days, the most important and sensitive assets of business, people and organization are their computer data or digital information. Number of portable devices like PDAs, smart-phones and portable computers has increased as the dependency on computing increased. With these, chances of intrusion activities, data theft and system compromises have increased to a far greater extent. In majority of the cases the actual information/data is more important and valuable than the hardware it is stored on and the unauthorized access of that data could be very harmful. The portable devices come under the area of highest threat of data theft and intrusion activities as such devices regularly travel in unsecured public places due to which they become very much prone/vulnerable to attacks.

IV. LITERATURE SURVEY : OVERVIEW OF BITLOCKER DRIVE ENCRYPTION TECHNOLOGY

Disks/volumes encrypted with BitLocker have a different signature than the standard FAT/NTFS header. Instead, in the first sector of their volume header, they have: 2D 46 56 45 2D 46 53 2D (in HEX format) or, -FVE-FS- (in ASCII format). These volumes can be identified by the BitLocker GUID/UUID (for e.g. 4967d63b2e294ad8-8399-f6a339e3d00). The actual data on the encrypted disk/volume is protected with either 128-bit or 256-bit AES encryption algorithm or optionally diffused using an algorithm called Elephant. The key used to do the encryption, i.e., the Full Volume Encryption Key (FVEK) and/or TWEAK key, is stored in the BitLocker metadata on the protected volume. The FVEK and/or TWEAK keys are encrypted using another key, namely the Volume Master Key (VMK). Several copies of the VMK are also stored in the metadata. Each copy of the VMK is encrypted using another key; also known as the key-

protector key. Some of the key-protectors are: TPM (Trusted Platform Module), Smart Card, Recovery key/password, Start-up key, User password. BitLocker also has the support for partially encrypted volumes.

The BitLocker key management system uses a series of keys to protect the data at rest. The various types of BitLocker keys along with their roles and functions during the encryption process are as mentioned below:

IV.I Full Volume Encryption Key (FVEK) [10]
The key used to protect the data, i.e., the sector data, is the Full Volume Encryption Key (FVEK). It is stored on the protected volume and is stored in an encrypted form. To prevent unauthorized access, the FVEK is encrypted with the Volume Master Key (VMK). The size of the FVEK is dependent on the encryption method used, i.e., FVEK is 128-bit of size for AES 128-bit and FVEK is 256-bit of size for AES 256-bit.

IV.II Volume Master Key (VMK) [10]
The key used to encrypt the FVEK is the Volume Master Key (VMK). It is also stored on the protected volume. The VMK is 256-bit. In fact several copies of the VMK are stored on the protected volume. Each copy of the VMK is encrypted using a different key such as the recovery key, external key or the TPM. If the volume is BitLocked using both external key as well as the recovery password, then there will be two metadata entries for VMK where each metadata entry stores the VMK, one encrypted with the recovery key and the other encrypted with the external key respectively. When decrypted, both the VMK will be the same. If the VMK differ, then it means that the decryption has failed. It is also possible that the VMK is stored unencrypted which is referred to as the clear key.

IV.III TWEAK Key [10]
The TWEAK Key is part of the FVEK stored encrypted with the Volume Master Key (VMK). The size of the TWEAK Key depends on the encryption method used. The key is 128-bit for AES 128-bit and the key is 256-bit for AES 256-bit.
The TWEAK key is present only when the Elephant Diffuser is enabled. The TWEAK Key is stored in the metadata entry that holds the FVEK which is always 512-bit. The first 256-bits are reserved for the FVEK and the other 256-bits are reserved for the TWEAK key. Only 128-bit of the 256-bits are used when the encryption method is AES 128-bit, i.e., when the Elephant Diffuser is disabled.

IV.IV Recovery Key [10]
BitLocker stores a recovery (or numerical) password to unlock the VMK. This recovery password is stored in a \%GUID \%\.txt file. For example, recovery password can be: 471207-278498-422125-177177-561902537405-468006-693451. The recovery password is valid only if it consists of 48 digits where every 6 numbers are grouped into a block, thus, consisting of 8 blocks. Here, each block should be divisible by 11 yielding a remainder 0. The result of a division by 11 of a block is a 16-bit value. The individual 16-bit values make up a 128-bit key.

IV.V External Key [10]
The external key is stored in a \%GUID \%\.BEK file. The GUID in the file-name equals the key identifier in the BitLocker metadata entry. The BEK file contains the external key identifier and a 32 byte external key. The different keys allow different mechanisms to be used to access the stored data. Each access mechanism can be used to decrypt a copy of the VMK which in turn is used to decrypt the FVEK which in turn is used to decrypt the protected data. The various structures where these keys along with their corresponding data and values get stored in certain pre-defined formats (as defined by Microsoft) are: FVE Metadata Block: 1. FVE Metadata Block Header. 2. FVE Metadata Header. 3. FVE Metadata Entry: (a) FVE Metadata Entry Types. (b) FVE Metadata Value Types.

V. RELATED WORK

Adi Shamir, Nicko van Someren [1] had theoretically discussed about the various procedures and techniques that could be employed, primarily to obtain the RSA keys from the huge amount of data. They had also theoretically discussed a case study where RSA algorithm is used for the process of encryption. They had also tried to locate the RSA keys based on entropy (randomness) of the data.

Limitations: Adi Shamir, Nicko van Someren [1] had just provided theoretical overview about the idea presented by them which does not provide proper knowledge on the implementation aspects of any tool, technique or procedure. They had taken into account only a specific scenario based on RSA encryption algorithm but no highlights or mention of any other algorithm is provided. They had taken into account lunch-time attack only, and no focus is put on any other attack which probably would have been possible.

Brian Kaplan, Matthew Geiger [2] had discussed about the various procedures and techniques that could be employed, primarily to obtain the keys directly from the RAM. They had also described a few attack techniques such as: Key-schedule attack, Brute-force attack, which could be utilized in finding the keys from the RAM. They had developed their own tool named “Disk Decryptor” for obtaining the keys. They had also suggested an interesting technique as to which memory locations within the RAM should be looked upon primarily to search for the keys as per the case scenario.

Limitations: Brian Kaplan, Matthew Geiger [2] had described the techniques and procedures taking into account

that the system is powered on, which might not always be the case. The attacks were described taking into account that the key might be stored in plain-text form in the RAM, which might not always be the case.

Eoghan Casey, Gerasimos J. Stellatos [3] demonstrated acquisition and mounting of clones/images of completely encrypted disks and then tried to decrypt them if possible, to gather all the crucial data that could be obtained from it. The authors had used software tools such as: 1. ImageMASSter Solo-III 2. FTK Imager 3. LiveView in VMware. They had discussed some arbitrary case studies and suggested more tools in reference to the same.

Limitations: Eoghan Casey, Gerasimos J. Stellatos [3] had just provided overview but neither described any proper procedure nor shared any implementation procedure details, which could help to gain better insight into the idea that they are trying to present in their research paper.

Sarah Lowman [4] had discussed about the various procedures and techniques that could be employed, primarily to obtain the keys to decrypt the encrypted data. The author had also discussed the "File Investigator File Find" software tool by Forensic Innovations, Inc., used to detect the presence of hidden encrypted volumes and tested it with sample data.

Limitations: Sarah Lowman [4] had mostly provided theoretical overview about the presented idea, which lacks proper knowledge on the implementation aspects of any tool, technique or procedure. Sarah Lowman [4] had also mentioned case studies to support the idea presented but, that did not provide proper know-how about the technical procedures that were utilized to find the case solutions. Good theoretical information from various reference papers was cited in Sarah Lowman's [4] research paper but, the information was not elaborated properly to the extent required.

Christopher Hargreaves, Howard Chivers [5] had discussed various procedures and techniques, such as "Volume Shadow Copy" in Microsoft's Windows operating system, and how they could be used to detect the presence of hidden encrypted volumes. They had taken into account the case study of "TrueCrypt" software tool for this purpose, since; it has the feature to make hidden encrypted volumes. They had also demonstrated the changes in data which could be visible with the help of visualization techniques.

Limitations: Christopher Hargreaves, Howard Chivers [5] had demonstrated the changes in data which are visible, taking into account only the FAT file-system without providing highlights on other file-systems. They had mostly described the techniques for Microsoft's Windows operating system and the special case of "TrueCrypt" software tool, and no focus was put on other operating systems and tools.

Sasa Mrdovic, Alvin Huseinovic [6] had discussed about the various procedures and techniques that could be used to obtain decrypted data from encrypted data. They had taken into account the case study of "TrueCrypt" software tool for this purpose. They had set up a test environment on Microsoft's Windows operating system and also described the use of other tools such as: Passware Password Recovery Key Forensic for obtaining decryption key, OSFMount tool for mounting the image. They had specifically used the Windows' hibernation system file for analysis.

Limitations: Sasa Mrdovic, Alvin Huseinovic [6] had demonstrated their problem statement well, but no highlight or mention of file-systems was provided, which may probably have had an impact on the procedure in obtaining the decrypted data. They had specifically demonstrated for Microsoft's Windows operating system along with "TrueCrypt" software tool for encryption and "Passware Password Recovery Key Forensic" for decryption, but no focus was put on other operating systems and tools.

Eoghan Casey, Geoff Fellows, Matthew Geiger, Gerasimos Stellatos [7] had highlighted the legal procedures for acquiring/gathering required information as per the case scenarios and also described the usage of tools such as: 1. EnCase Enterprise (for analyzing physical memory dump to find the encryption pass-phrase) 2. HBGary Fastdump Pro (for acquiring live contents of RAM and page-file) 3. FTK Imager Lite (for acquiring an image of an encrypted volume).

Limitations: Eoghan Casey, Geoff Fellows, Matthew Geiger, Gerasimos Stellatos [7] had just provided theoretical overview about their presented idea, which lacked proper knowledge on the implementation aspects of any tool, technique or procedure. They had specifically demonstrated the tools for Microsoft's Windows operating system, but no focus was put on other operating systems. They had also mentioned case studies to support their presented idea but, that did not provide sufficient know-how about the technical procedures that would had been utilized to find the case solutions.

Adedayo M. Balogun, Shao Ying Zhu [8] had theoretically discussed about the various procedures and techniques that could be employed, primarily to obtain the keys to decrypt the encrypted storage devices. They had also theoretically discussed a case study on "TrueCrypt" encryption tool and challenges faced because of it.

Limitations: Adedayo M. Balogun, Shao Ying Zhu [8] had just provided theoretical overview about the presented idea, which lacked proper knowledge on the implementation

aspects of any tool, technique or procedure. They had provided good theoretical information from various reference papers cited in their research paper but, did not elaborated properly to the extent required.

Mario Piccinelli, Paolo Gubian [9] had discussed and demonstrated the use of "NIST Statistical Test Suite" for detecting encrypted volumes. They had also described 15 different tests from the "NIST Statistical Test Suite", which they had used for presenting their idea. They had taken into consideration "TrueCrypt" encrypted files for that purpose.

Limitations: Mario Piccinelli, Paolo Gubian [9] had taken into consideration the specific case of "TrueCrypt" encrypted files, but no focus was put on other tools. They had not highlighted or mentioned the type of file-system used.

P. Shabana Subair, C. Balan, S. Dija, K.L. Thomas [10] had demonstrated the presented idea on a FAT32 BitLocked volume. They had described the techniques to decrypt the VMK by using: Recovery Password, Startup Key. They had later on obtained the FVEK by using the previously obtained VMK and later on, decrypted the entire encrypted volume using the FVEK.

Limitations: P. Shabana Subair, C. Balan, S. Dija, K.L. Thomas [10] had not highlighted the programming language which they might had used to develop the code. They had also not highlighted that whether any automated tool was used to perform any of the tasks presented in their research paper.

VI. PROPOSED WORK

Today's scenario where disks and volumes are encrypted for the purpose of security, is a good methodology used to keep the data at rest secure, but the same methodology is also used by cyber-criminals for escaping from the hands of law enforcers and judicial systems. This is actually the point of motivation to find and develop techniques such that encrypted disks/volumes could be decrypted, or encrypted data could some-how be looked into, such that it could help to obtain crucial data whenever required.

There are many encryption techniques and algorithms available out of which the case for BitLocker Drive Encryption is considered as it is one of the most widely used, since Microsoft Windows is the most commonly used operating system by most of the users and organizations around the world.

The problem statement could hence, be further simplified and stated as an attempt to obtain data from disks/volumes that are encrypted using BitLocker Drive Encryption Technology.

VII. OBJECTIVE

Objective of this research work is: The BitLocker key should be found from the raw RAM dump as it would help to obtain data as well as aid to develop techniques that could assist the digital forensic analysts to obtain data from live systems whose disks/volumes are encrypted by BitLocker Drive Encryption Technology.

VIII. SCOPE

Scope for this research work is: The hard-disk capacity is set to a few gigabytes in size, the RAM capacity is set to a few gigabytes in size, the test environment is limited to the virtual machine, BitLocker Drive Encryption Technology is to be explored.

IX. ASSUMPTIONS

Assumptions for this research work are: Whenever the encrypted disk/volume is accessed using the password chosen before encrypting that disk/volume, the key which gets unlocked to decrypt the encrypted data must compulsorily be getting stored locally at some memory location in the RAM. This is because the complete disk/volume does not get decrypted in an instant, but only the required data (i.e., files and processes) which are actively accessed by the system or in current use get decrypted since it follows OTF (on-the-fly) principle. Various types of RAM analysis tools (open-source or proprietary) would be used to analyze the raw RAM dump.

X. CONSTRAINTS

Constraints for this research work are: All types of RAM analysis tools could not be used to analyze the raw RAM dump as it is a time-consuming process. Freely available RAM analysis tools would have to be used as licenses of proprietary tools are too costly. Availability of RAM analysis tools which could specifically find BitLocker decryption keys from raw RAM dump is less in number, so manual analysis of the raw RAM dump may also have to be done which is a time-consuming process. Scripts which could help in the analysis of raw RAM dump could not be developed for the current scenario.

XI. METHODOLOGY

XI.I Steps carried out to acquire and process, interpret inputs for this research work are: 1. Process of simulating the necessary test environment for FDE/FVE in a virtual machine was performed. 2. One of the most commonly used FDE/FVE techniques, i.e., BitLocker Drive Encryption Technology was selected to be used for this research work. 3. The other inputs which would be required were: Raw RAM

dump, Recovery key, Decryption key (VMK). 4. After the successful completion of the encryption process, the recovery key would be kept saved for a backup purpose. 5. Next, the raw RAM dump would be analyzed using various tools and the required BitLocker decryption key (VMK) would be extracted from it. 6. The extracted decryption key (VMK) from the raw RAM dump would be reverse-engineered to obtain the FVEK.

XI.II Steps carried out for research work are: 1. VMware Workstation v11 (build 2305329) was installed on the host operating system. 2. A virtual machine was created with the specifications as mentioned in the sub-section "Experimental Setup" from the section "Implementation:" of this research paper. 3. The following tools were installed in the guest operating system within the virtual machine: Belkasoft RAM Capturer, Elcomsoft Forensic Disk Decryptor and HxD HEX Editor. 4. The implementation tasks of the research work were carried out. 5. Documentation stating various parameters and information related to the research work was prepared in a standard format.

XII. IMPLEMENTATION

XII.I Experimental Setup: To simulate the case of FDE/FVE, a virtual machine was created in VMware Workstation v11 (build 2305329) with the following specifications: Processor: Dual-Core, RAM: 1 GB, Hard-disk Storage Capacity: 25 GB, Operating System: Microsoft Windows 7 Ultimate Edition, Encryption Technology Used: BitLocker Drive Encryption Technology from Microsoft (already bundled with this edition of the operating system)

XII.II Implementation Procedure: 1. After an instance of virtual machine was set up, BitLocker Drive Encryption Technology, was utilized for the purpose of encrypting the disk/volume. 2. After the complete encryption of this disk/volume, the system was restarted to ensure the disk/volume gets locked automatically. 3. After the system got rebooted for the first time, the disk/volume was unlocked by entering the password chosen before the encryption process, so that the BitLocker decryption key (i.e., VMK) gets loaded in the RAM. 4. As per the assumptions mentioned in the section "Assumptions" of this research paper, raw RAM dump of the system was taken and again the system was restarted for the second time to wipe the decryption key (i.e., VMK) from the RAM and then, after the system got rebooted for the second time, the raw RAM dump was analyzed using various RAM analysis tools and HEX file editors to check whether the desired BitLocker decryption key (i.e., VMK) was present in the raw RAM dump (either in plain-text form or in cipher-text form). 5. The tool used here for raw RAM dump analysis was Elcomsoft Forensic Disk Decryptor. 6. The key finding process from the raw RAM dump may give rise to various scenarios as mentioned within the sub-sub-sections of the next sub-section of this research paper.

XII.III Expected Scenarios:

XII.III.I Encryption scenario: The 25 GB hard-disk space allocated to the virtual machine would be containing the operating system itself. This system volume would be encrypted using BitLocker Drive Encryption which would simulate the case of FDE. If the case of FDE could not be simulated, then another logical volume would be created by shrinking the system volume and making a new volume of size 1 GB from the existing space allocated for the hard-disk drive within the virtual machine itself. This logical volume would then be encrypted using BitLocker Drive Encryption which would simulate the case of FVE.

XII.III.IIRaw RAM dump analysis result scenario: The analysis performed on the raw RAM dump file using various RAM analysis tools (open-source or proprietary), would help to obtain the BitLocker decryption key (i.e., VMK) which later on, helps to decrypt the contents of the drive/volume OTF (on-the-fly). If no such key is obtained as output from the raw RAM dump file after using various RAM analysis tools (open-source or proprietary), then the raw RAM dump file would be analyzed using HEX file editors to find and retrieve the desired decryption key (i.e., VMK).

XIII. RESULTS AND DISCUSSION

After following the procedure mentioned in the sub-section "Implementation Procedure" from the section "Implementation:" of this research paper, it was observed that the VMK key used by BitLocker for encryption was found using Elcomsoft Forensic Disk Decryptor (refer Figure 1).

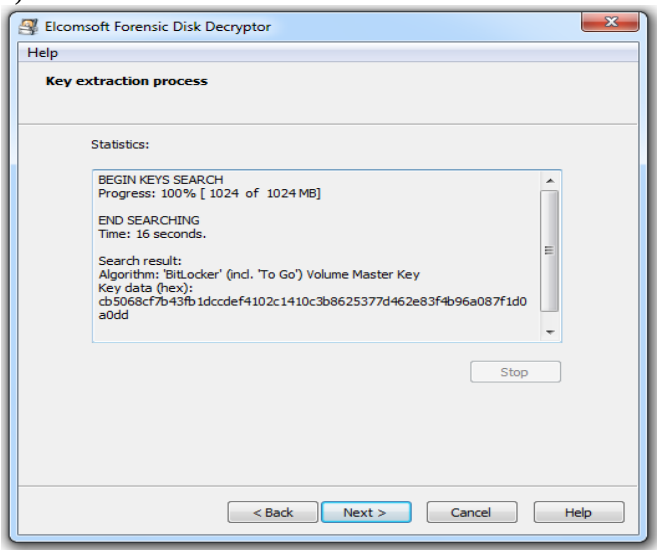

Figure 1: VMK obtained using Elcomsoft Forensic Disk Decryptor.

Also, upon searching the raw RAM dump file with the hash value obtained (in HEX form) using Elcomsoft Forensic Disk Decryptor, that key was found in its corresponding ASCII form also (refer Figure 2).

This implies that the BitLocker decryption key (i.e., VMK) is stored in the RAM in encrypted form and decrypts data OTF.

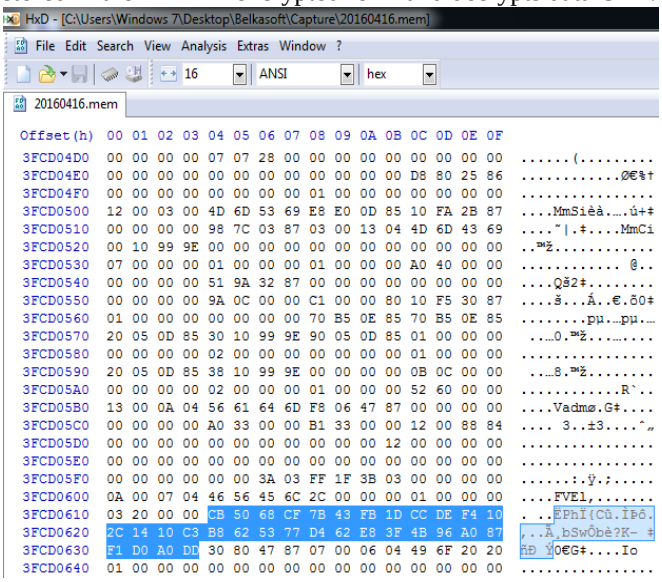

Figure 2: VMK observed in RAM dump file in encrypted form.

XIV. CONCLUSION AND FUTURE SCOPE

Through this research work, an attempt at retrieving the BitLocker decryption key (i.e., VMK) from raw RAM dump especially in systems where "NTFS" file-system volumes are encrypted, has resulted in a fair amount of success. The purpose of this is to aid the digital forensic analysts to overcome the barrier of encryption while retrieving data from encrypted HDDs/SSDs in certain case scenarios, which to some extent seems to be achievable.

The initial findings obtained from experimental setup have proved to be beneficial at this stage of the research work.
The case of FVE is taken into consideration for this research work, whereas it can be expanded to include the case of FDE also.

Techniques should be developed to identify the following parameters of encrypted HDDs/SSDs: 1. Which encryption algorithm is used? 2. Based on the signature present in the headers, what/which type of encryption tool/technology is used to encrypt the device? 3. Which portions on these storage devices are left unencrypted?

Also, RAM analysis and it's forensics should be considered at a higher priority in retrieving unencrypted data (i.e., files and keys) from encrypted data since, it is mandatory for the key to be available while decrypting data (i.e., files and processes) OTF and the only place where the decryption keys get stored currently, is in the RAM.

Also, SSD analysis and its forensics should be taken as the next step of improvement. Tools and techniques should be developed to obtain unencrypted data from encrypted data on SSDs, since the rate of adoption of SSDs in the form of modern day storage media is increasing all across the globe.

REFERENCES

[1] Adi Shamir, Nicko van Someren, "Playing Hide and Seek with Stored Keys", Proceeding, FC '99 Proceedings of the Third International Conference on Financial Cryptography, Springer-Verlag , pp.118-124, 1998.
[2] Brian Kaplan, Matthew Geiger, "RAM is Key: Extracting Disk Encryption Keys From Volatile Memory", pp.1-29, 2007.
[3] Eoghan Casey, Gerasimos J. Stellatos, "The Impact of Full Disk Encryption on Digital Forensics", pp.93-98, 2008.
[4] Sarah Lowman, "The Effect of File and Disk Encryption on Computer Forensics", pp.1-14, 2010.
[5] Christopher Hargreaves, Howard Chivers, "Detecting Hidden Encrypted Volumes", IFIP International Conference on Communications and Multimedia Security, pp.233-244, 2010.
[6] Sasa Mrdovic, Alvin Huseinovic, "Forensic Analysis of Encrypted Volumes Using Hibernation File", 19th Telecommunications Forum (TELFOR) Proceedings of Papers, pp.22-24, 2011.
[7] Eoghan Casey, Geoff Fellows, Matthew Geiger, Gerasimos Stellatos, "The growing impact of full disk encryption on digital forensics", DIGITAL INVESTIGATION 8, pp.129-134, 2011.
[8] Adedayo M. Balogun, Shao Ying Zhu, "Privacy Impacts of Data Encryption on the Efficiency of Digital Forensics Technology", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 4, Issue.5, pp.36-40, 2013.
[9] Mario Piccinelli, Paolo Gubian, "Detecting Hidden Encrypted Volume Files via Statistical Analysis", International Journal of Cyber-Security and Digital Forensics (IJCSDF), Vol. 3, Issue.1, pp.30-37, 2014.
[10] P. Shabana Subair, C. Balan, S. Dija, K.L. Thomas, "Forensic Decryption of FAT BitLocker Volumes", Digital Forensics and Cyber Crime: Fifth International Conference, ICDF2C,pp.17-29, 2014.

Authors Profile

Mr. Jay Parag Mehta completed Bachelor of Engineering from University of Pune, India in year 2012 and Master of Science from Gujarat Forensic Sciences University, India in year 2016. He is currently working as a professional in the Digital Forensics and Cyber Security industry. He is a member of the CyberAttack Community since 2014. He has 5 years of overall Industry Experience and 2 years of Research Experience.

Dr. Digvijaysinh M. Rathod completed Ph.D. in composion of RESTful web service composition from Ganpat University, Gujarat (India) and currently working as Assistant Professor (Cyber Security and Digital Fornsics) in Institute of Forensics Science, Gujarat Forensics Sciences University since 2014. He has published more than 25 research papers in reputed international journals and conferences including IEEE. His main research work focuses on Web Application Security, Cloud Security , IOT Security, Darkweb Forensics and Block Chain. He has 15 years of teaching and research experience.

