# Forensic Analysis of Encrypted Volumes Using Hibernation File

Sasa Mrdovic, Alvin Huseinovic, Faculty of Electrical Engineering Sarajevo

*Abstract* — **Nowadays, software tools are commonly used to encrypt data on hard disk. Those tools keep encryption keys in system memory to provide the user easy access to plain text of encrypted files. Key possesion enables data decryption. A procedure that includes usage of hibernation file as a source of memory content is described. Publicly available tools are used to perform the procedure. The procedure is successfully tested on a system that uses current encryption program.**

*Keywords* — **encrypted storage, encryption keys, static analysis, live analysis, hibernation file**

## I. INTRODUCTION

COMPUTER forensics is an important part of information security and legal protection. Humans, nowadays, tend to use computers for everything they do. Anything done on a computer leaves digital traces. Those traces help investigators reconstruct actions performed using the computer. Finding and collecting traces is not a simple task. There are several factors that complicate that process. Data, and meta-data, in digital form can easily be changed. Such a change during collection would render collected data inadmissible as evidence in a court of law. Therefore strict procedures are developed that ensure evidence preservation.

Standard approach to forensic analysis of a computer is called static analysis. Hard disks and other nonvolatile memory devices, of powered off computer, are cloned bit by bit. Such forensically correct copies are analyzed, usually in read only mode to ensure data integrity, even on, cloned memory devices. Static analysis enables investigator to thoroughly search stored data and find relevant evidence. There are number of documents that describe procedure and number of tools to support and automate search and analysis.

Static analysis has certain limitations. There are practical issues regarding amount of data that needs to be analyzed on contemporary disks with standard sizes in hundreds of gigabytes. More important for evidence collection is loss of data from volatile memory. RAM of running computer holds data on everything that is going on. These data may contain important evidence that is inevitably lost when computer is powered off for static analysis. In addition, data on hard disk might be encrypted and therefore unreadable for investigator.

Different, or more correctly complementary, approach is called live analysis. Volatile memory of a running computer is analyzed, or more often dumped to a file for later offline analysis. Live analysis enables collection of data from RAM but has its own issues. Any action performed on the running computer inevitably changes its state. The same applies for RAM analysis or dumping. The change might make collected data unacceptable as evidence. Additional issue is that this procedure of data collection is not verifiable and repeatable. Furthermore live environment from which content of volatile memory is collected is not reliable. Tools and OS available on computer being examined might have been tampered with.

Usage of hibernation feature, available on all modern OS, is more recent approach [1, 2]. This power management feature enables saving computer state to disk. In this way content of RAM is dumped to a file that can be analyzed offline. Static analysis methods can be used to extract hibernation file. There are tools available that can create memory dump from hibernation file. And there are tools that analyze memory dump.

This paper explores possibilities of using data available in hibernation file to empower static hard disk analysis. The focus is on decryption of encrypted data found on hard disk. Data needed for decryption and its availability in hibernation file will be discussed. It will be shown that such a data is available. A procedure for decryption of encrypted files will be presented. The procedure uses publicly available tools. In addition, availability of data being edited in encrypted files will be discussed.

The rest of the paper is organized as follows. Related work is mentioned in section 2. Section 3 explains approach to storage encryption used by current tools. Approach to decryption keys extraction memory and proposal for simple forensic procedure for decryption is given in section 4. Section 5 presents practical implementation of proposed procedure. Conclusion and discussion on directions for future research work are in section 6.

## II. RELATED WORK

Principles of computer forensics are given in [3]. More

S. Mrdovic is with the Faculty for Electrical Engineering, University of Sarajevo, Zmaja od Bosne bb, 71000 Sarajevo, Bosnia and Herzegovina (corresponding author; phone: +387(33)250-753; fax: +387(33)250-725; e-mail: sasa.mrdovic@etf.unsa.ba).

A. Huseinovic is with the Faculty for Electrical Engineering, University of Sarajevo, 71000 Sarajevo, Bosnia and Herzegovina (e-mail: ahuseinovic@etf.unsa.ba).

recent overview of current state of digital forensics is given in [4]. Papers [5, 6] present area of live analysis and its challenges. Offline volatile memory analysis and its advantages and issues are subject of [7]. Hibernation file as source of volatile memory was proposed in [1]. [2] presents usage of hibernation file for combining static and live analysis.

Identification of cryptographic key in memory was started by [8] some time ago. Practical solutions are more recent. Extraction of encryption keys from Linux memory dump is subject of [9]. Hypothetical attack on `TrueCrypt` using keys found in memory is described in [10]. Linear scan of memory is proposed as a method for extraction of cryptographic keys from memory in [11]. [12] presents successful attack on encryption using boot kit and hibernation file. Various issues of forensic approach to identification and extraction of cryptographic keys from memory and new practical method are subject of [13]. This is an area of active research. This paper is oriented toward practical implementation and presents procedure that forensic investigators could follow when dealing with encrypted volumes.

## III. ENCRYPTED STORAGE ENCRYPTION

### A. Encryption

Data encryption is process in which data in its readable form, called plain text, is transformed using, usually publicly known, algorithm with secret parameter, called a key. The result of this transformation, called cipher text, cannot be read (transformed back to plain text) without the key. Data encryption can be done in hardware or software. Software encryption is of interest in this case. It is assumed that computer being examined is standard PC without special hardware for encryption. There are a number of software tools for data encryption. These tools offer encryption of various parts of file system. A single file, a folder or an entire disk could be encrypted. Also, so called virtual drive encryption is available. In this form of encryption virtual drive is created and stored as a single file on hard disk. Any file stored on this virtual drive is encrypted. The aim of all tools is to make content of encrypted files unavailable to anyone who does not have the key for decryption. This should hold even if the encrypted data is analyzed offline in different environment.

Encryption tools are used by humans. Humans are not good at remembering (and even entering) long numbers and encryption keys are long numbers. Therefore, these programs enable users to enter passwords or passphrases that are internally transformed to encryption keys. This means that encrypted storage can be decrypted: by using the program used for encryption and user supplied password or by any program that implements encryption/decryption algorithm used and encryption key. Encryption programs are available, for purchase or for free, and encryption algorithms publicly known. For successful decryption one needs to have either password or key used for encryption.

Encryption programs, usually for user's convenience, do not ask users to enter password each time they want to access an encrypted file. Users enter password once, usually when they mount encrypted storage, and use the file as any other regular file in file system. In order to achieve this encryption program needs to store decryption key in memory. Key might be protected or obfuscated but still needs to be somewhere in RAM. This fact is the weak point and can be used for decryption. If content of RAM is available for analysis decryption key, or keys, could be extracted. Any protection of keys provided by encryption program or operating systems will be useless if memory content is analyzed offline.

## IV. ENCRYPTED STORAGE DECRYPTION

### A. Decryption keys availability and extraction

As RAM content is required for decryption it needs to be obtained. This is a subject of live analysis. Memory dump can be made using any of the methods used for this purpose. Since this paper concentrates on usage of hibernation file it is assumed that this file is available. As it was previously proposed [2] computer being taken for examination should be sent to hibernation, instead of powered off. In this way memory content will be preserved in hibernation file. It is a fact that running computer can be used to access data that is in encrypted files but this would change evidence and is unacceptable. It will be shown later that even if the computer was not running when it was taken there are chances that decryption keys might be in a hibernation file. This would happen if the computer was, at some point in past, sent to hibernation while encrypted storage was mounted.

Different encryption programs store decryption keys in RAM in different ways. Even the same programs do this differently among different versions. There are methods developed by various researchers that can locate keys in memory dump and decrypt encrypted storage [11, 12]. There are already commercial tools that can automatically perform such key recovery and decryption procedure [14]. Decryption keys availability and extraction.

### B. Procedure For Dealing With Encrypted Data Storage

Here is a simple procedure that forensic investigators might use when dealing with encrypted storage. The procedure was tested with a particular encryption program and set of tools. The tests and results are provided in the next section. The procedure should be general for current encryption programs.

A computer being taken for forensic investigation should be sent to hibernation. An image of hard disk of the computer should be created. This image should include saved hibernation file. The image could be mounted using appropriate tool and examined for encrypted storage. If such storage is found it needs to be decrypted. Tools for decryption key extraction from memory dump and decryption of encrypted storage require memory dump and encrypted container. Memory image is available in

hibernation file. Hibernation file should be copied from hard disk image. Depending on decryption tool hibernation file could be used directly or converted to memory dump. Decryption tools can decrypt storage. Decrypted storage could be mounted with appropriate tool. Encrypted files should be available in decrypted form for further analysis by investigator.

Even in case when it was not possible to send computer to hibernation, before it was taken for examination, hard disk image should be examined for various memory dumps. There might be an older hibernation file. If the file was created while encrypted storage was mounted it is very possible that it contains decryption keys. Memory, or parts of it, might be stored in page and dump files. There are tools that can transform these files into memory dumps that can be fed to decryption tool and enable decryption.

This is an active area of research and tool development. There are different tools being developed for different operating systems and different encryption programs. Tools and encryption programs will inevitably change but principle and procedure should remain the same. Encryption programs will need to store keys in memory, for user's convenience. As long as keys are stored in RAM and RAM content can be dumped for offline analysis it will be possible to extract them and decrypt encrypted storage. It must be mentioned that encryption programs take measures to protect memory keys from being written to hibernation file [15] but with limited success.

As a proof of concept, next section presents encrypted storage decryption procedure on concrete test system. It presents current set of tools used in each step.

## V. TESTING

Test system for investigation was Windows XP SP3 OS with 10 GB hard disk and 256 MB RAM. Hard disk and RAM size are smaller than current standards but big enough to present procedure. Encryption program `TrueCrypt` 7.0a was installed on a system. `TrueCrypt` volume (virtual drive) with size of 10 MB was created. Encryption algorithm used was AES with 256 bit key size (Fig 1).

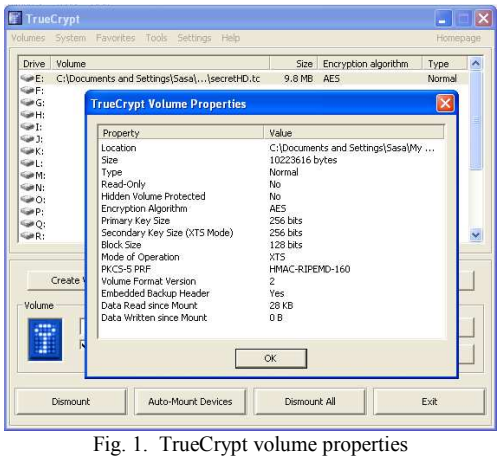


Fig. 1. TrueCrypt volume properties

### A. Scenario 1

Two text files, "`Saved secret.txt`" and "`Edited secret.txt`", were created on encrypted volume. Both files had just one line of text. Both files were saved (Fig. 2).

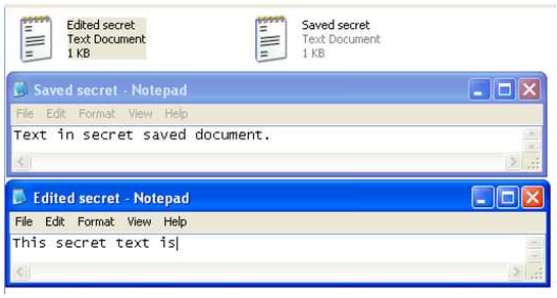


Fig. 2. Files on encrypted volume

The second file, "`Edited secret.txt`", was opened and few words " `being edited ...`" were added to it. At this point the system was sent to hibernation without saving edited file.

Image of hard disk was created using dd tool. The image was mounted using `OSFMount` tool [16]. `TrueCrypt` volume container file "`secretHD.tc`" was found. It was copied to examiners computer for decryption. Then hibernation file `hiberfile.sys` was found on mounted image and copied to examiners computer. With encrypted storage and memory dump in hibernation file everything needed for decryption was available.

Encrypted `TrueCrypt` volume file and hibernation file were given to decryption tool `Passware Password Recovery Kit Forensic` [14] (Fig. 3). Decryption took only 12 seconds. It was not possible to recover user password used in `TrueCrypt`, but decryption key was found in memory dump and used for decryption.


Fig. 3. Decryption tool

Decrypted image of encrypted `TrueCrypt` volume file was stored on examiners computer. The image was mounted using `OSFMount` tool. Mounted volume had both `TrueCrypt` encrypted files in their decrypted form as they were saved (Fig. 4). Desired decryption was achieved.

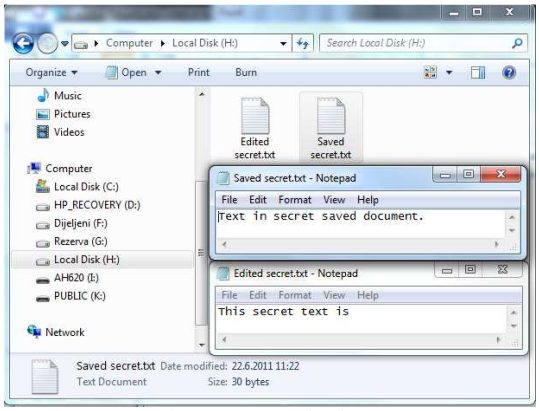

Fig. 4. Decrypted volume

Since hibernation file was available next step was to use it to try to find out if encrypted files were edited when system was sent to hibernation. Aim was to find the most recent versions of encrypted files that might not have been saved to encrypted volume. For this purpose hibernation file was converted to Microsoft crash dump file using MoonSols tool `hibr2dmp` [17] .

Using Microsft Windows crash dump file debugger `WinDbg` [18] list of running processes was extracted from created crash dump file. It was found that `notepad.exe` was running. List of loaded dlls showed that notepad was used to open document "`E:\Edited secret.txt`". Memory dump was searched for last three words of saved encrypted "`Edited secret.txt`" document: "`secret text is`". It was found that this text continues with words "`being edited ...`". Document located on `TrueCrypt` volume that was edited was available in its unencrypted form in memory dump.

### *B. Scenario 2*

In the second scenario test system was awaken from hibernation. Edited document "`Edited secret.txt`" was saved and closed. `TrueCrypt` volume was unmounted. System was turned off, not hibernated. Now, idea was to test if encrypted `TrueCrypt` volume could be decrypted without system being hibernated.

Hard disk image was created and mounted. `TrueCrypt` volume container file was again copied to examiners computer to a different location than the first one. Old hibernation file, remained from previous hibernation, was found. This file was also copied to examiners computer to the same location as encrypted volume file. The same decryption tool, `Passware Password Recovery Kit Forensic`, was used to try to decrypt encrypted `TrueCrypt` volume file. Only this time it was given hibernation file that was not created while `TrueCrypt` volume was mounted, but the older one found on imaged hard disk

Enrypted `TrueCrypt` volume file was successfully decrypted to a new unprotected file. This decrypted file was mounted. Mounted volume had both `TrueCrypt` encrypted files in their decrypted form as they were saved. Desired decryption was achieved even with older hibernation file.

## VI. CONCLUSION

Computer forensic analysis could be difficult, or even impossible, if data on hard disks being analyzed is encrypted. Availability of memory dump might help with decryption. Software encryption tools keep encryption keys in memory. This memory might be saved to a disk by hibernation. Tools are available that can decrypt encrypted storage if memory dump is available. This is rather straight forward procedure and is presented here. Even old hibernation file might be used as a source of encryption keys in some cases.

Future research could be oriented towards encryption key extraction from memory traces in various files on hard disk, like page or old hibernation files. Changes of encryption programs oriented towards better protection of keys in memory should be followed. It remain to be seen if it will be possible to have decrypted files available to users and prevent extraction of decryption keys from memory. This is the weakest point from encryption perspective, and backdoor to encrypted file for forensic investigators.

## REFERENCES

[1] M. Suiche, "Windows hibernation file for fun 'n' profit," Black Hat, USA, 2008.
[2] S. Mrdovic, A. Huseinovic, E. Zajko, "Combining static and live digital forensic analysis in virtual environment," Information, Communication and Automation Technologies, ICAT 2009., pp.1-6
[3] M.M. Pollitt, "Principles, practices, and procedures: an approach to standards in computer forensics," Second International Conference on Computer Evidence, 1995, pp. 10-15.
[4] B.D. Carrier, "Digital Forensics Works," IEEE Security and Privacy, vol. 7, Mar. 2009, pp. 26-29.
[5] F. Adelstein, "Live forensics: diagnosing your system without killing it first," Commun. ACM, vol. 49, 2006, pp. 63-66.
[6] B. Hay, M. Bishop, and K. Nance, "Live Analysis: Progress and Challenges," IEEE Security and Privacy, vol. 7, Mar. 2009, pp. 30-37.
[7] C. Waits, J.A. Akinyele, R. Nolan, and L. Rogers, Computer Forensics: Results of Live Response Inquiry vs. Memory Image Analysis, CERT, 2008.
[8] A. Shamir, N. van Someren, "Playing hide and seek with stored keys", Financial cryptography (LNCS 1648), Springer-Verlag, 1998. p. 118–24.
[9] T. Pettersson, "Cryptographic key recovery from linux memory dumps", Chaos Communication Camp, 2007.
[10] A. Walters, N.L. Petroni Jr, and I. Komoku, "Volatools: integrating volatile memory forensics into the digital investigation process," Black Hat DC, 2007.
[11] C. Hargreaves, H. Chivers, "Recovery Of Encryption Keys From Memory Using A Linear Scan", The International Workshop On Digital Forensics, Barcelona, Spain, 2008.
[12] P. Kleissner, "Stoned Bootkit", Black Hat, USA, 2009.
[13] C. Moe, S. Thorkildsen, A. Arnes, "The persistence of memory: Forensic identification and extraction of cryptographic keys", Digital Investigation 6, Elsevier, 2009., pp s132-s140.
[14] Passware, "Passware Kit Forensic 10.5", http://www.lostpassword.com/kit-forensic.htm (accessed on June 30, 2011)
[15] `TrueCrypt`,"Hibernation File", http://www.truecrypt.org/docs/?s=hibernation-file (accessed on June 30, 2011)
[16] PassMark Software, "`OSFMount`", http://www.osforensics.com/tools/mount-disk-images.html (accessed on June 30, 2011)
[17] `MoonSols`, "MoonSols Developer Network", http://www.msuiche.net/msdn/ (accessed on June 30, 2011)
[18] Microsoft, "`WinDbg`", http://msdn.microsoft.com/en-us/library/ff561300(v=vs.85).aspx (accessed on June 30, 2011)