GRACEWIPE: SAFE AND SECURE FILE UNDER COERCION

R.Meenaskhi Sundaram[1], N.Karthikeyan[2], R.Lokesh[3], S.Santhosh[4]

[1]*Student, Computer Science and Engineering, New Prince Shri Bhavani Collage of Engineering and Technology, Tamilnadu, India.*
[2]*Student, Computer Science and Engineering, New Prince Shri Bhavani Collage of Engineering and Technology, Tamilnadu, India.*
[3]*Student, Computer Science and Engineering, New Prince Shri Bhavani Collage of Engineering and Technology, Tamilnadu, India.*
[4]*Assistent Professor, Computer Science and Engineering, New Prince Shri Bhavani Collage of Engineering and Technology, Tamilnadu, Chennai.*

ABSTRACT

The information on research is safely stored as documents in a web-based page which can be accessed by scientific team page which has a Admin. The encryption and decryption technique is based on the AES Algorithm which is latest and more preferable. The encryption is based on the cipher concept. And the project implements the Gracewipe technique which is more securable in which the secured files cannot be accessed by coercion (threatening the user). For secure interaction the user is given with a different combination password by automatic generator. The files can also be decrypted when required by the user. Files can also be shared among the users and Admin for better interaction of projects when the user specifically shares the documents. And there is a fake document which can be accessed when there is a wrong combination of real password by the Gracewipe technique. Hence the secured files are safer from coercion and can be accessed later. If there is a full wrong combination of real password the Gracewipe redirects to the same page, we designed the Gracewipe in such a way it can delete the real password under coercion and a new password is generated and sent to the registered mail-Id.

Keyword: - *Gracewipe technique, adversary, coercion, wrong combination, fake document.*

1. INTRODUCTION

1.1 Introduction to Information security

Information security shortened to InfoSec is the practice of preventing unauthorized access, use, disclosure, disruption, inspection, modification or destruction of information. Sometimes referred to as computer security, information technology security is information security applied to technology. It is note that a computer does not necessarily mean a home desktop. A computer is any devices with a processor and some memory. Such devices can range from non-networked standalone devices as simple as calculators, to network mobile computing devices such as Smartphone's and tablet computers. IT security specialists are almost always found in any major establishment due to the nature and value of the data within the country larger businesses. They are responsible for keeping all of the technology within the company secure from malicious cyber attacks that is attempt to breach into critical private information or gain control of the internal systems.

1.1 Threats

Information security threats come in many different forms. Some of the most common threads today are software attacks, theft of equipment or information, sabotage, information extortion. Most people have experienced software attacks of some sort. Viruses, phishing attacks and Trojan horses are some common examples of software attacks. The theft of intellectual property has also been an extensive issue for many businesses in the IT field.

Identity theft is the attempts to act as someone else usually to obtain the person’s personal information are to take advantage of their access to vital information. Governments, military, corporations, financial institutions, hospitals and private businesses amass a great deal of confidential information about their staffs, customers. Products, research and financial status. Most of the information’s now collected, processed and stored on computers and transmitted across networks to other computers.

1.2 Information Assurance

The act of providing trust of the information, that the confidentiality, Integrity and Availability of the information are not violated, for e.g. ensuring that data is not lost when critical issues arise. These issues include, but are not limited to natural disasters, computer or server malfunction or physical theft. Since most information is stored on computers in our modern era, information assurance is typically dealt with by IT security specialists. A common method of providing information assurance is to have an off-site backup of the data in case one of the mentioned issues is arises.

1.3 TrueCrypt

The TrueCrypt on-the-fly full-disk encryption (FDE) utility is possibly the most popular choice in its kind. It supports plausible deniable encryption (PDE) in the form of a hidden volume, which appears as free space of another volume. In the regular mode, an encrypted volume is explicitly mounted through TrueCrypt, on demand, after the OS is already booted up. We use its PDE-FDE mode (available only in Windows), where the OS volume is also encrypted and the original Windows MBR is replaced with the TrueCrypt MBR, which prompts for a password and loads the next 40-60 sectors (termed TrueCrypt modules) to decrypt the system volume. Self-Encryption Drives (SEDs). SEDs offer hardware-based FDE as opposed to software-only FDE solutions. A major benefit of an SED is its on-device encryption engine, which always keeps disk data encrypted.

2 PROBLEMS IN EXISTING SYSTEM

In the existing system, the user can not experience the Gracewipe concept more securable which makes date unsecure. Attackers can break the security easily. Since the analysis of the encrypted data can be viewed. And the encryption used is also less secure. It has poor user interface environment. The encryption of the data is not as secured and the technique used is an old technique.

3 PROPOSED SYSEM

The main objective of this project is to provide the facilities for the user to have secure data storage from External factors. It makes the user data more securable. And the user-interface is also improved. Any changes in the regular existing pattern are observed keenly. This observation leads to the major change with the security issues of the project. The duplicate data can also be stored with user’s wish which is similar like original data. Files can also be shared among the users and Admin for better interaction of projects when the user specifically shares the documents.

3.1 PROCESS IN GRACEWIPE

Overview of how Gracewipe goals are achieved. For goal (1), we introduce P D that retrieves KN but at the same time deletes KH from TPM (fig.1). Thus, if either the user/adversary enters a P D, the hidden data will become inaccessible and unrecoverable (due to deletion of K H). P N, P H and P Ds should be in a usual situation, the user can use either P H or P N to boot the corresponding system. If the user is under duress and forced to enter P H, they may input a P D instead, and Gracewipe will immediately delete KH (so that next time P H only outputs a null string).

Under duress, she can reveal P N/P Ds, but must refrain from exposing P H. The use of any P D at any time(emergency or otherwise), will delete KH the same way, and thus goal(2) can be achieved. Goal (3) can be achieved by a chained trust model and deterministic output of Gracewipe. The trusted environment is established by running the deletion operation via DRTM, e.g., using Intel TXT through tboot [24]. We assume that Gracewipe’s functionality is publicly known and its measurement (in the form of values in TPM PCRs) is available for the target environment, so that the adversary can match the content in PCRs with the known values, e.g., through a TPM quote operation. Gracewipe prints a hexadecimal representation of the quote value, and also stores it in TPM NVRAM for further verification.

A confirmation message is also displayed after the deletion (e.g., “A deletion password has been entered and the hidden system is now \permanently inaccessible!”). For goal (4), we use TPM’s sealing feature, to force the adversary to use a genuine version of Gracewipe for password guessing. Sealing also stops the adversary from modifying Gracewipe in such a way that it does not trigger key deletion, even when a P D is used. We use long random keys (e.g., 128/256-bit AES keys) for actual data encryption to thwart offline attacks directly on the keys. A side-effect of goal(4) is that, if a Gracewipe-enabled device (e.g., a laptop) with sensitive data is lost or stolen, the attacker is still restricted to password guessing with the risk of key deletion.

The following diagram shows about the phases involved in the process of Gracewipe. This process is iterative in nature.

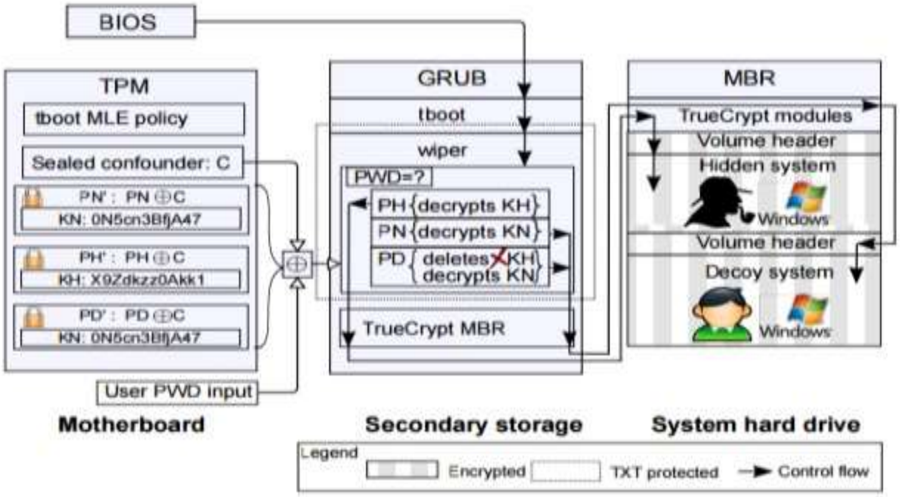

Fig.1 Process in Gracewipe

3.2 SYSTEM ARCHITECTURE

The core part of Gracewipe’s functionality includes bridging its components, unlocking appropriate TPM-stored keys, and deletion of hidden volume key. In fig.2 we term this part as the wiper, which is implemented as a module securely loaded with tboot. It prompts for user password, and its behavior is determined by the entered password (or more precisely, by the data retrieved from TPM with the password). Namely, if it appears otherwise (as designed by deletion indicator) to have a control block for deletion, the wiper performs the deletion and passes the decoy key KN to TrueCrypt. We modified TrueCrypt to directly accept input from the wiper (i.e., the original TrueCrypt password prompt in bypassed), and boot one of the encrypted systems. As the wiper must operate at an early stage of system boot and still provide support for relatively complex functionality, it must meet several design considerations, including: 1) it must be bootable by tboot, as we need tboot for the measured launch of the wiper.

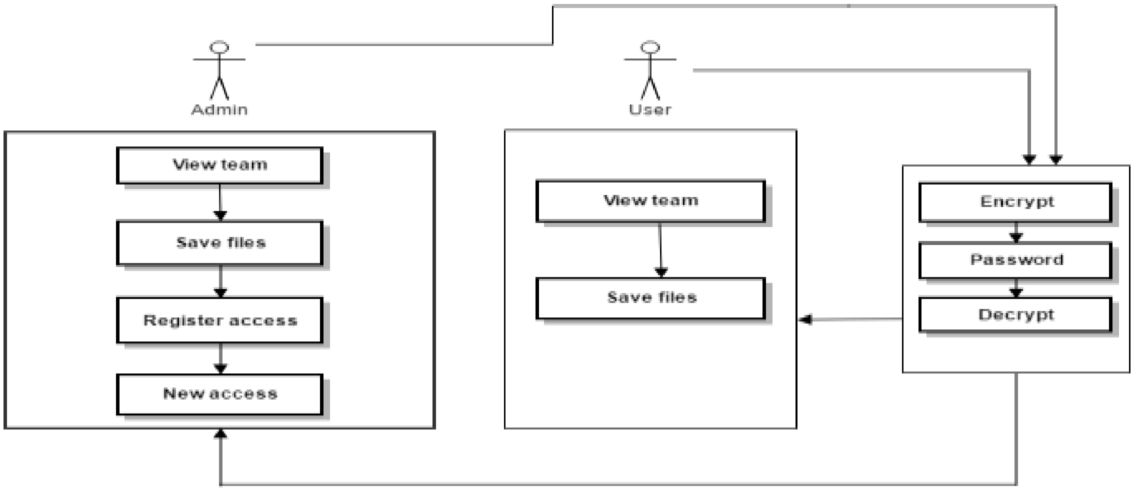

Fig.2 Architecture

This can be achieved by 7Sec a related tboot discussion thread at (Aug. 2012) [6]: to required file formats (e.g., ELF) and header structures (e.g., multiboot version number). 2) It must load the TrueCrypt loader for usual operations, e.g., decrypt the correct volume and load Windows. This is mainly about parameter passing (e.g., TrueCrypt assumes register DL to contain the drive number). 3) It must access the TPM chip and perform several TPM operations including sealing/unsealing, quote generation, and NVRAM read/write. Note that at this point, there is no OS or trusted computing software stack (such as TrueSerS [2]) to facilitate TPM operations, 4) It must provide an expected machine state for the component that will be loaded after the wiper (e.g., Windows). Both TrueCrypt and Windows assume a clean boot from BIOS; however, Windows supports only strict chain loading, failure of which causes several troubles including system crash (see Section V). Execution steps. (1) The system BIOS loads GRUB, which then loads tboot binary as the kernel, together with other modules including the wiper, ACM SINIT module and the policy list. (2) Tboot checks for required support on the platform; if succeeded, tboot starts the MLE by calling GETSEC [SENTER]. (3) All measurements are calculated and matched with the values stored in TPM.

If the matching is successful, the wiper is loaded in the same context as tboot; otherwise, execution is halted. (4) The wiper prompts the user for password, and uses the entered password to decrypt location where we store KH/KN one by one. If none is decrypted, it halts the system; otherwise, the wiper copies the decrypted key (i.e., TrueCrypted password) to a memory location to be retrieved later by TrueCrypt. (5) If one of the P Ds is entered (indicate by the decrypted data), the wipers immediately erase KH from TPM, and perform a quote to display the attention string on the screen. It either halts the system or continues loading the decoy system according to user choice. (6) The wiper switches the system back to real-mode, reinitializes it by mimicking what is done by BIOS at boot time, and replaces the handler of INT 13h. (7) TrueCrypt MBR is executed, which decompresses the subsequent sectors from the hard drive into system memory. TrueCrypt also inserts its filter to the handlers of INT 13h and 15h. The corresponding volume is decrypted on-the-fly, if the TrueCrypt password (as received from the wiper) is correct. Then the boot record on the decrypted partition is chain loaded, and Windows is booted. Storing Gracewipe components. For booting the target system, Gracewipe's software components (GRUB, tboot, wiper, TrueCrypt MBR) can reside on any media, including secondary storage. In our proof-of-concept system, we keep these components on a secondary USB storage. The target hard drive only contains TrueCrypt modules (except its MBR) and all encrypted partitions. All Gracewipe components can also be placed on the target hard drive alone, with additional effort, including: (a) an extra partition with file system is needed to store tboot and its modules: (b) GRUB MBR will overwrite part of TrueCrypt, and thus either of them must be relocated; and (c) TrueCrypt MBR must be modified not to read the TrueCrypt, and thus either of them must be from their predefined locations (i.e., sector 2of the disk).

4 FEATURES OF GRACEWIPE

TrueCrypt was a popular open source, on-the-fly encryption application that allowed you to work with encrypted files as you would work on files located on a regular drive.

Without on-the-fly encryption, actively working with encrypted files is an enormous pain and the outcome is usually either that people simply do not encrypt their files or they engage in poor security practices with their encrypted files because of the hassle of decrypting and encrypting them ETL processes for cleaning data, and also for building, processing, training, and updating models. An attacker cannot share an inherent limitation.

5 LITERATURE REVIEW

Lianying Zhao and **Mohammad Mannan** [1] proposed by making data permanently inaccessible when the file under coercion. If the key deletion occurs by a user supplied deletion password, the user may face serious consequences. The deletion should be used only for very high-value data, which must not be exposed to third parties.

Rescorla [2] blog post, discusses technical and legal problems of data protection under coercion. Limitations of existing approaches including deniable encrypting, verifiable destruction (Vanish [4]) have been discussed. He also proposes possible solutions, one of which is based on leveraging a hardware security module (HSM) with a limited try scheme. The HSM will delete the encryption key if wrong keys are entered a limited number of times.

Reardon et el [3] provide a comprehensive survey of existing solutions for secure deletion of user data on physical media, including flash, and magnetic disks. Solutions are categorized and compared based on how they are interfaced with the physical media. And the features they offer. The adversary in Gracewipe can be classified as bounded coercive as he can detain the victim and keep the device for as long as he can needs with all hardware tools available but Gracewipe protected data without the proper key.

6 CONCLUTION

We using by this Gracewipe technique files can also be shared among the users and admin for better interaction, when the user specifically shares the documents. The real data cannot be accessed under any attack which makes the Gracewipe more secure. In future, tutor techniques can be used encrypt and decrypt both audio and video recognition. And the bugs in the Gracewipe technique are fixed for better interface. So, the project can become more secure and reliable for better feasibility.

7 REFERENCES

[1] Lianying Zhao, Mohammad Mannan ,"Deceptive Deletion Triggers Under Coercion", vol.11,no.12,Dec 2016.

[2] E.Rescorla, "Protecting Your Encrypted Data in the Face of Coercion" (Feb 11, 2012). [Online]. Available: http://www. educatedguesswork.org/2012/02/protecting_your_encrypted_data.html.

[3] J. Reardon, D. Basin, and S. Capkun, "SoK: Secure data deletion," in Proc. IEEE Symp. Secur. Privacy, San Francisco, CA, USA, May 2013.

[4] R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," in Proc. USENIX Secur. Symp., Montreal, QC, Canada, Aug. 2009

[5] J. A. Halderman et al., "Lest we remember: Cold-boot attacks on encryption keys," in Proc. USENIX Secur. Symp., San Jose, CA, USA, 2008.

[6] H. Bojinov, D. Sanchez, P. Reber, D. Boneh, and P. Lincoln, "Neuroscience meets cryptography: Designing crypto primitives secure against rubber hose attacks," in Proc. USENIX Secur. Symp., Bellevue, WA, USA, Aug. 2012.

[7] J. Reardon, S. Capkun, and D. Basin, "Data node encrypted file system: Efficient secure deletion for flash memory," in Proc. USENIX Secur. Symp., Bellevue, WA, USA, Aug. 2012.