IJETIE International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 4, April 2015
ISSN: 2394 - 6598
www.ijetie.org

Enhancing Content Security using Hidden Volume Concept Deniable Encryption

A.Prema,
M.E., CSE.,
VPMM Engineering College for Women,
Srivilliputhur,
Mail:apremacse@gmail.com

Mr.John De Britto,
Assistant Professor,
VPMM Engineering College for Women,
Srivilliputhur,
Mail:dany_melbia_pieo@yahoo.co.in

Abstract: **Smartphone or mobile devices have become a very crucial part of our daily life in which we store more confidential information. Encryption is the general way of providing confidentiality to our content and confidentiality remains safe till our decoy keys remain safe. Plausibly deniable encryption (PDE) is encryption which helps us to secure our data in a situation when user cipher text is intercepted and he/she is forced into revealing the key. They may instead provide a decoy key to reveal a plausible and begin decoy message. Here our work is to performance this kind of deniable encryption in our data storage system to secure our content and extending it feature using hidden volume. It provides storage encryption without deniability. We use the terms "decoy" and "outer" interchangeably when referring to passwords, keys, and volumes in the standard mode. In that creates fixed memory space for hidden volume which is a drawback for user storing more information. so we implement an extended mobiflage which actually have a dynamic memory hidden volume for plausible deniable encryption.**

INDEX: File System Security, mobile platform security, storage encryption, deniable encryption.

I. Introduction

Smartphone's and other mobile computing devices are being widely adopted globally. With this increased use, the amount of personal/corporate data stored in mobile devices has also increased. Due to the sensitive of this data, all major mobile OS now include some level of storage encryption. Mobile phones have been extensively used to capture and publish many images and videos of recent popular revolutions and civil disobedience. Mobile OS lacks of deniable encryption—a critical feature helps user in a forced situation to give their decryption keys. But they provide a decoy (fake) key which produce a different reasonable plaintexts output from a given cipher text, keeping main plaintext safe. We postulate that PDE would be an attractive or even a necessary feature for mobile devices. Note, however, that PDE is only a technical measure to prevent a user from being punished if caught with contentious material; an adversary can always wipe/confiscate the device itself if such material is suspected to exist. Mobiflage's threat model and operational assumptions, and few legal aspects of using PDE in general. The major concern with maintaining plausible deniability is whether the system will provide some indication of the existence of any hidden data. Mobiflage's threat model is mostly based on past work on desktop PDE solutions; we also include threats more specific to mobile devices. It provides storage encryption without deniability. The user will supply their decoy password at boot time to enter the standard mode. In this mode, the storage media is mounted in the default way (i.e., the same configuration as a device without Mobiflage). We use the terms "decoy" and "outer" interchangeably when referring to passwords, keys, and volumes in the standard mode.

IJETIE
International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 4, April 2015
ISSN: 2394 - 6598
www.ijetie.org

II. Background :

1. We explore sources of leakage inherent to mobile devices that may compromise deniable storage encryption. Several of these leakage vectors have not been analyzed for existing desktop PDE solutions.

2. We present the Mobiflage PDE scheme based on hidden encrypted volumes—the first such scheme for mobile systems to the best of our knowledge.

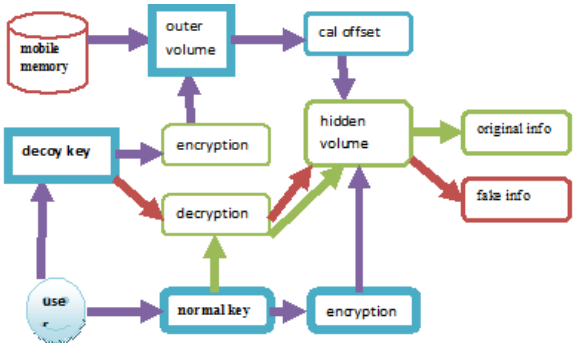
3. We provide a proof-of-concept implementation of Mobiflage for Android 4.x (Ice Cream Sandwich and Jelly Bean). We incorporated our changes into 4.x and maintained the default full disk encryption system. During the normal operation of Mobiflage (i.e., when the user is not using hidden volumes), there are no noticeable differences to compromise the existence of hidden volumes.

4. We address several challenges specific to Android. For example, to avoid PDE- unfriendly features of the Ext4 file system (as used for the Android userdata partition), we implement our hidden volumes (userdata and external) within the FAT32-based external partition.

5. We analyze the performance impact of our implementation during initialization and for data-intensive applications. In a Nexus S device, our implementation appears to perform almost as efficiently as the default Android 4.x encryption for the applications we tested. However, the Mobiflage setup phase takes more time than Android FDE, due to a two-pass wipe of the external storage (our Nexus S required almost twice as long; exact timing will depend on the size and type of external storage).

III. System Design:

We introduce Mobiflage, a plausibly deniable encryption (PDE)-enabled storage encryption system for the mobile devices. It includes countermeasures for known problems against desktop PDE implementations. It uses a hidden volume to store the PDE data so that the adversary has the encrypted device and full knowledge of Mobiflage's design, but lacks the PDE key and password. The existence and location of the hidden volume is therefore also unknown. In that creates fixed memory space for hidden volume which is a drawback for user storing more information. so we implement an extended mobiflage which actually have a dynamic memory hidden volume for plausible deniable encryption.



MOBILE MEMORY:

The mobile memory is the storage of the mobile. The entire disk is encrypted with a decoy key and formatted for regular use, we call this the outer volume.

OFFSET CALCULATION:

After creating the outer volume, and before creating the hidden volume, we calculate a list of the outer volume blocks that contain meta-data and generate a offset which determine the size of the hidden volume. The generated offset is greater than one half and less

IJETIE
International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 4, April 2015
ISSN: 2394 - 6598
www.ijetie.org

than three quarters of the disk; i.e., the hidden volume's size is between 25-50 percent of the total disk. We choose this offset as a balance between the hidden and outer sizes: the outer volume will be used more often, the hidden volume is used only when necessary.

DECOY KEY:

Decoy key is a key. It use to encryption and decryption of mobile memory. It encrypting the information and decrypting the message but it does not gives the original information. It gives information related to document but its not original.

NORMAL KEY:

Normal key is a key. It use to encryption and decryption of mobile memory. It encrypting the information and decrypting the message it gives the original information.

HIDDEN VOLUME:

It is one part of the mobile memory. It does not shown to the adversary, the only user can known that memory. It used for storing the personal information and secret information of the user.

ENCRYPTION:

In cryptography, **encryption** is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption does not of itself prevent interception, but denies the message content to the interceptor.

RESULTS AND DISCUSSION:

In that discussion about the modules and it performance of the module. To explain that different module using the data flow diagrams. To implement the project to using the different module. It having the four modules. It using different concept for implementing the project.

Steganographic File-systems vs. Hidden Volumes:

There are currently two main types of PDE systems for use with FDE: stegano-graphic file systems and hidden volumes. Steganographic file systems' known drawbacks include: in efficient use of disk space, possible data loss, and increased IO operations. These limitations are unacceptable in a mobile environment, for reasons such as performance sensibility, and relatively limited storage space. Consequently, we choose to use hidden volumes for Mobiflage. This implies: no altered file system drivers are required; IO is as efficient as a standard encrypted volume; and the chance of data loss is mitigated, although not completely eliminated. Most deniable file systems are lossy by nature. Hidden volumes mitigate this risk by placing all deniable files toward the end of the storage device. Assuming the user knows how much space is available for the deniable volume, they can refrain from filling the outer volume past the point at which the hidden volumes begin.

MODULES:

STORAGE LAYOUT:

The entire disk is encrypted with a decoy key and formatted for regular use, we call this the outer volume. Then an additional file system is created at an offset within the disk and encrypted with a different key; this is referred to as the hidden volume. When the outer volume is decrypted and mounted, it

does not reveal the existence or location of the hidden volume. To prevent leakage, Mobiflage must never mount hidden volumes alongside outer volumes. Thus, we create corresponding hidden volumes, or RAM disks, for each mutable system mount point. Some hidden volumes may be decoys, but at least one hidden volume will contain the actual sensitive data and be encrypted with the true key. Since the outer volume is filled with random noise before formatting, there are no distinguishing characteristics between empty outer-volume blocks and hidden volume blocks. When the outer volume (or a hidden decoy volume) is mounted, it does not reveal the presence or location of any other hidden volumes. . The disk can be thought of as the concatenation of encrypted volumes, each with a different key, When the disk is decrypted with a given key, the other volumes will appear to be uniformly random data. Each decrypted volume will appear to consume all remaining disk space on the device. For this reason it is possible to destroy the data in the hidden volumes by writing to the currently mounted volume past the volume boundary. This is unavoidable since a visible limit on the mounted volume would indicate the presence of hidden volumes.

OFFSET CALCULATION

After creating the outer volume, and before creating the hidden volume, we calculate a list of the outer volume blocks that contain meta-data and generate a offset which determine the size of the hidden volume. The generated offset is greater than one half and less than three quarters of the disk; i.e., the hidden volume's size is between 25-50 percent of the total disk. We choose this offset as a balance between the hidden and outer sizes: the outer volume will be used more often, the hidden volume is used only when necessary. It complicates a dictionary attack, by mandating the adversary capture a larger portion of the disk. If the offset was at a known location, then an adversary could perform a dictionary attack on a couple of kilobytes of data captured from that region (only the key and file system magic-number are necessary to prove the existence of a hidden volume). With our approach, the adversary must capture at least 25% of the storage to mount an attack. Note that the efficiency of a dictionary attack is not affected by the offset location.

ENABLE PDE

During the enabling process the data within the partition gets formatted so the user needs to back up the data. User then enters the decoy and true passwords, for the outer and hidden volumes respectively. Mobiflage then formats, and encrypts the outer and hidden volume.

PDE INTERFACE

The user enters the decoy password during pre-boot authentication to activate the standard mode. All data saved to the device in this mode will be encrypted but not hidden. When the user requires the added protection of deniable storage, they will reboot their device and provide their deniable password when prompted; In the PDE mode hidden volume is not shown to the users, while only outer volume are shown.

IV.Conclusion:

Hidden Volume concept takes advantage of the already existing tendency towards choosing familiar words as passwords. Users are generally frowned

upon by security advocates for making such choices, as these words can easily be subjected to dictionary attacks. To keep such records hidden from authorities, deniable storage encryption may offer a viable technical solution. Such PDE-enabled storage systems exist for mainstream desktop/laptop operating systems. With Hidden volume explore design and implementation challenges of PDE for mobile devices, which may be more useful to regular users and human rights activists. It design is partly based on the lessons learned from known attacks and weaknesses of desktop PDE solutions. To consider unique challenges in the mobile environment. To address some of these challenges, we need the user to comply with certain requirements.

V. Reference:

[1]D. Dolev and A.C. Yao, "On the Security of Public Key Protocols," IEEE Trans. Information Theory, vol. IT-29, no. 2, pp. 198-208, Mar.1983.

[2]J. Assange, R.-P. Weinmann, and S. Dreyfus, "Rubberhose: Cryp-tographically Deniable Transparent Disk Encryption System,"1997.

[3]comScore, "comScore Reports September 2012 U.S. Mobile subscriber Market Share," 2012.

[4]R.-P. Weinmann, "Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks," Proc. USENIX Workshop Offensive Technologies (WOOT '12), 2012.

[5]A. Skillen and M. Mannan, "On Implementing Deniable Storage Encryption for Mobile Devices," Proc. Network and Distributed System Security Symp. (NDSS '13), Feb. 2013.