

Multifactor, multiple people. Authentication approach for unlocking encrypted files.

Yuri Gorokhov *, Lars Noergaard Nielsen *

* University of California, San Diego

Submitted as part of graduate security course CSE227.

Proposal of a system to authenticate access to encrypted files using both Multifactor and multiple people, across different locations. Making sure that files are only accessible with the consent of all involved participants.

Multifactor | Encryption

Introduction

Controlling who has access to files is often a requirement in industry and various other contexts. Systems for dealing with information that only is accessible with multiple people's consent is therefore interesting to investigate. Software for file access control purposes include Dell Identity Manager[2], User Lock Access Manager [1] and native OS support such as an Access Control List. These systems is not addressing security as such, as not providing encryption capabilities. Common for these solutions is that file access is administered centrally by a administrator. We propose an approach were users actively set file permissions by agreeing to encrypt files by their common consent, only allowing access to these files when all parties have responded to the access request. The latter step is additionally secured by MultiFactor Authentication.

Yubikey

TODO

Previous work

We based our model on a proposed hard drive encryption mechanism published on the Yubikey website[3]. In their proposed configuration the Yubikey is programmed with a secret key after which it is able to perform HMAC-SHA1 encryption. The device is said to be operating in Challenge-Response mode

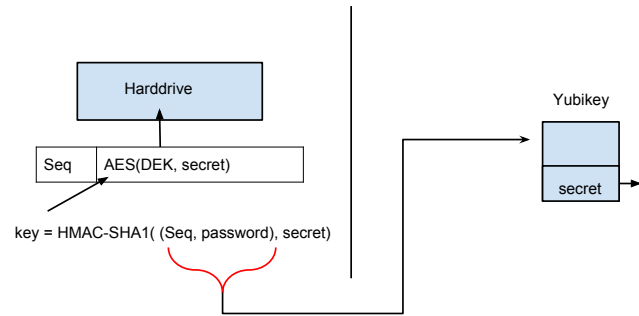


Fig. 2. Local hard drive encryption configuration.

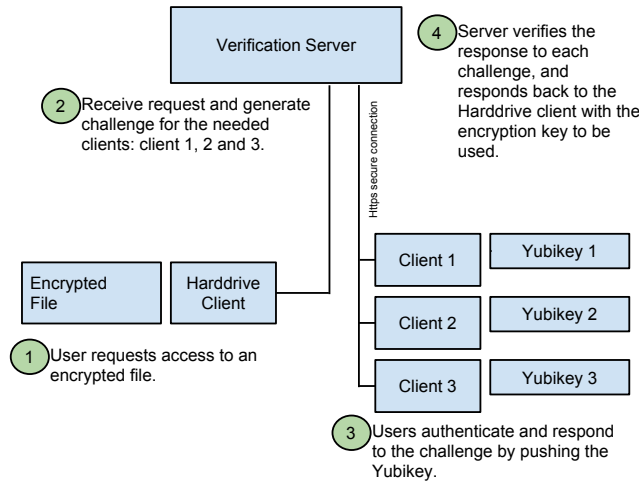


Fig. 3. System components and interaction. 4 major steps to grant access.

since you can send it a challenge and it will respond with the HMAC-SHA1 encryption of the challenge with the secret key. This is depicted in figure 1.

Figure 2 shows how this mode is used for encrypting a local hard drive. The hard drive is encrypted with a Drive Encryption Key (DEK) which is stored in a table along with the

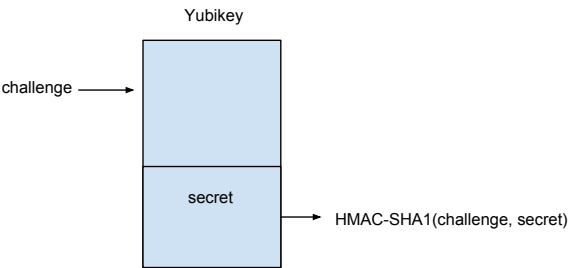


Fig. 1. Challenge-Response mode of Yubikey.

Reserved for Publication Footnotes

secret. Both are encrypted using AES. The key used for the encryption is shown in Figure 2. Once the user enters his password a challenge is generated. The challenge consists of the password itself and a sequence number (Seq) that is also stored in the table. This challenge is sent to the Yubikey, whose response allows us to decrypt the DEK and the secret. The hard drive can now be decrypted. After decryption the DEK is re-encrypted with a new sequence number and the secret.

System Description

The proposed system is composed of a user client, harddrive client and a verification server.

A benefit from this central point of control is that the server can deny access to a file, even though participants grant access, which might be useful in some access schemes.

Multifactor authorization: Yubikey

In this work we choose Yubikey for Multifactor authentication for several reasons: It provides a simple procedure for the user - only a physical touch on the device is necessary to allow the device to respond to the presented challenge.

Encryption scheme

This section presents the encryption scheme used, to ensure that decryption of the file is only possible when responses from all participants and their Yubikey challenges are retrieved.

Discussion

Discussion on strengths and weaknesses of the solution

ACKNOWLEDGMENTS. This work was supported by..

1. <http://www.isdecisions.com/lp/userlock/userlock-windows-network-security.htm?gclid=CMfPI-rqisQCFciBfgodhxwAmQ>
2. <http://software.dell.com/products/identity-manager-data-governance/>
3. <https://www.yubico.com/applications/disk-encryption/full-disk-encryption/>