How to Not Lose Your Keys

Approaches to Managing Multiple Identities and Devices for Single Users

Mark MacIntyre Wheatley
University of Waterloo
200 University Ave. W
Waterloo, ON N2L 3G1
ms3macin@uwaterloo.ca

ABSTRACT

In today's day and age, the average computer user finds at their disposal a plethora of computing devices of various sizes, form factors and computing power, targeted at different aspects of the user's lifestyle. Indeed, it is not uncommon to find the average user in possession of at least two devices, namely, a computer and a smartphone. Additionally, we find new devices being released everyday: smart accessories, smart appliances and the ilk. The average user expects all their applications to work seamlessly across their devices. A challenge in security arises when the user needs to manage their multiple interconnected devices and their multiple identities for their applications.

This paper aims to provide an overview of developments in approaches to managing multiple devices and identites for single users. **Comment:** [list of names and some findings]

1 INTRODUCTION

Comment: Moved some of the content from the abstract here

The new milleneum saw an explosion in the developement of such devices and a moving away of users from single computers towards personal computing [4]. Now, in 2017, it is commonplace for an average user to own and actively use several computing devices. With new trends and technology comes new challenges and a challenge in security arises when the user needs to manage their multiple interconnected devices and their multiple identities for their applications. The device owner usually relies on some "Keychain" app to share the user's cryptographic private key and this key sharing can often be insecure and expose the user to grave security concerns of theft and malware[1]. Devices such as phones and watches are frequently removed or replaced. Also, the loss of a single device can compromise the security of the user and their remaining devices.

In the next section, we look at the definition of this problem and look at the various possible solutions that try to solve it.

2 BACKGROUND INFORMATION

In section 2.1 we talk about the multi-device problem and describe it. Section 2.2 enumerates the various possible solutions and briefly describes each one.

2.1 The Multi-device Problem

The multi-device problem as introduced above is a situation that arises when a user possesses several computing devices at a time. Cryptographic keys that were once simply stored on a single computer are now required by several devices and must somehow be securely distributed across these devices. Users need these to consistently authenticate themselves as single identity across multiple devices. This means that secret keys which are intended to be stored securely are now going to be moved around exposing them to multiple threats of theft such as phishing, pulling them from memory or password cracking attempts (if they are password protected) [1]. The simplest solutions to this problem involve either attempting to securely synchronize a single key across multiple devices or to have a per-device key system. While these simple solutions solve the *multi*-device problem, they create a set of new problems that must be addressed[1].

2.2 Overview of Various Solutions

We will now present and overview of the several solutions that are capable of resolving the multi-device problem as well as accompanying issues. They are as follows[1]

- **Per-device keys** As described above, the user has an independant key on each device. The idea of per-device keys does not really solve the multi-device problem in that the user must generate and maintain several individual keys which is cumbersome. Also, all third parties must be made aware of every idividual private key owned by the user to be able to consistently authenticate that user with the same identity. Also, unique keys for each device will keep the verifying third party informed of which device the user is currently using. This creates a privacy issue where third parties can monitor the usage of a user's devices just from the keys.

- **Key sync** This involed the user making several copies of the same secret key and copying them to all devices. This procedure has several issues. The act of copying the secret key itself is susceptible to various attacks leading to the key being compromised. If the user loses one device than they must update the key for all other devices. This is because the lost device may be used to obtain the secret key and compromise all the other devices owned by the user.

- **Manual thresholding** The user has per-device keys but embeds and additional police in each signature that tells the verifying third party to look for other signatures from other devices belonging to the user. The issue with manual thresholding is the similar to that of the simple per-device keys method. A verifying third party can monitor the usage of the user's devices.

- **Personal PKI** Is one of the approaches that has seen significant interest and development. It built on the idea of the key sync but the user has an additional "master" key on one device which is use to validate the other keys. It is based on the idea of a public key infrastructure, but in this case it is entirely personal and provides the user with the ability to manage their own keys. **Comment:** [Revise/Add more Information] The obvious disadvantage is that the user has to manage an additional "master" key.

- **Group Signatures** Group Signatures work on the idea that a single member in the group can sign a message for the other members. They present a single key to the third party but internally have separate keys[2]. The idea of group signatures can be extended to devices where the user uses one device to sign for all the other devices. This idea has been explored for the case of multiple password management in the form of Pico[5] where the user can use a single device to replace all their PINs and passwords. This has not seen significant development for the multiple devices.

- **Threshold Cryptography** Is another approach that has seen significant interest and development. It is by far the most promising approach as it deals with the multi-device problem and also provides security against lost or theft of a device[3]. In this method either the key itself or some cryptographic operations are distributed among the user's n devices[**?**]. Whenever a devices requires to perform a cryptographic operation, either a user defined *threshold* number of devices t where $t \leqslant n$ work together to recover the original key or the cryptographic operation itself is distributed over t devices. Any less then t devices will not be able to divulge any valuable information.

Comment: Add Figure from Shatter showing comparison
Comment: Add definitions for properties of the schemes

3 THE *PERSONAL* KEY INFRASTRUCTURE

3.1 PorKI

3.2 Personal PKI for Smart Devices

3.3 CONIKS

4 THRESHOLD CRYPTOGRAPHY

4.1 Shatter

4.2 Blockchains and CoSi

5 CONCLUSIONS

REFERENCES

[1] Erinn Atwater and Urs Hengartner. 2016. Shatter: Using Threshold Cryptography to Protect Single Users with Multiple Devices. *WiSec '16: Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks* (2016), 91–102. https://doi.org/10.1145/2939918.2939932

[2] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. 2003. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. (2003), 614–629. https://doi.org/10.1007/3-540-39200-9_38

[3] Yvo Desmedt, Mike Burmester, Rei Safavi-Naini, and Huaxiong Wang. 2001. Threshold Things That Think (T4): Security Requirements to Cope with Theft of Handheld/Handless Internet Devices. *Symposium on Requirements Engineering for Information Security, West Lafayette, Indiana, USA* (2001).

[4] John Lyle, Andrew Paverd, Justin King-Lacroix, Andrea Atzeni, Habib Virji, Ivan Flechais, and Shamal Faily. 2013. Personal PKI for the smart device era. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7868 LNCS (2013), 69–84. https://doi.org/10.1007/978-3-642-40012-4_5

[5] Frank Stajano. 2011. Pico: No more passwords! *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7114 LNCS (2011), 49–81. https://doi.org/10.1007/978-3-642-25867-1_6