Towards an Open Ecosystem for Embedded Tamper Resistant Hardware in Mobile Devices

Michael Hölzl
UAS Upper Austria,
Campus Hagenberg
Softwarepark 11
A-4232, Hagenberg
michael.hoelzl@usmile.at

UbiComp '13 Adjunct, Sept 8-12, 2013, Zurich, Switzerland.
ACM 978-1-4503-2139-6/13/09...$15.00.

Abstract
Insufficient security and privacy level on mobile devices, have made it difficult to utilize sensitive systems like mobile banking, mobile credit cards, mobile ticketing or mobile passports. Solving these challenges in security and privacy, could provide a potential future for such systems towards better mobility and higher confidence level of end-user services. We address this issue by introducing an open ecosystem for mobile applications on tamper resistant hardware. A transparent integration of security features of such hardware will allow security critical applications to securely store and read data, transfer data to servers and securely communicate with other hardware parts (e.g. touch screen, sensors, etc.). In this research we will investigate variants and analyse trade-offs of integrating such an interface.

Author Keywords
Tamper resistant hardware, trusted execution environment, transparent secure channel, usability, secure element

ACM Classification Keywords
D.4.6 [Security and Protection]: Cryptographic controls, Information flow controls.; H.5.m [Information Interfaces and Presentation]: Miscellaneous.

Introduction

Although mobile operating systems have support for authentication (PIN/Pattern entry, face recognition, etc.) or other security features (on-device encryption) major problems still exist: they are either too complicated (e.g. long, complex password) or too easy to circumvent (e.g. 2D single image face authentication). Another problem that has been identified is downloading and using third party applications [1]. Mobile platforms usually use sandboxing together with permissions to prevent "apps" from unauthorized access to personal data or other application environments [5]. Users have to decide if they allow those permissions which is, according to Miller [5], not a good choice. Other research has also shown that sandboxing does not give sufficient security against exploitable vulnerabilities [3, 2]. Those problems lead to our main motivating vision to introduce a new secure user-friendly communication between user and hardware (end-to-end security). We want to achieve a level of security which allows applications like mobile banking, mobile payment, electronic passport to be executed in a trusted environment. The conceptual idea is to use virtualization to restrict even the untrusted operating system from accessing data or hardware components.

In the context of mobile devices we also have to consider that flash memory can not be trusted. Unauthorized access or data manipulation by third party applications is a major security threat for security-critical systems. One attempt to overcome this issue is to use *Embedded Tamper Resistant Hardware (ETRH)* to safely store and process security-critical data and applications. The advantage of this attempt compared to current similar business solutions (e.g. ARM TrustZone) is the higher security level that can be achieved by using a second processor. Examples for such hardware are Smartcards, Secure Elements and Trusted Platform Modules. They all include a secure, tamper-resistant key storage and an auxiliary processor for cryptographic operations. A tight integration of those security methods in our secure layer will allow security critical (e.g. banking, ticketing) as well as non-security critical (e.g. gaming) applications to make use of the ETRH features. After all, we will provide an *open ecosystem* for every application running on the mobile device to make use of these features over our virtual channel. Such an open ecosystem will also give applications the possibility to install, delete and manage own applets (small applications running on the ETRH) on the tamper resistant hardware.

Research Focus

In the topic of this research we will investigate technical approaches and analyse trade-offs of integrating interfaces for an open ecosystem of ETRH into current- and next-generation mobile devices with special focus on mobile phones. However, the result of this research is not only restricted to make use of ETRH in mobile phones. Other possible fields of applications are within sensor modules in a sensors network for secure key storage or further mobile devices with security demands.

Main research questions, which we are focusing on, consider the usability of an open ecosystem as well as necessary requirements to establish a secure channel from the user interface (UI) to the ETRH. We also investigate ways to use the ETRH for secure hardware access (e.g. sensors and touch screen for user input) and to make use of this in the UI. To the best of our knowledge, no research in the matter of this specific secure channel implementation has been published. In the following, we list the main questions with underlying technological sub-questions that are considered in this research topic:

What is the user annoyance limit where security will be turned off?

- How transparent can you make a secure channel between user and hardware? When will the user recognize performance issues and be annoyed by additional interaction effort?
- Does an ETRH have sufficient resources for an open ecosystem that enables a secure channel to third party applications and the user?

Is it feasible to create an open ecosystem on a tamper resistant hardware?

- What are the requirements for such an open ecosystem in terms of memory, processing speed and trust? Which changes are required to current organizational work-flows?
- How to bridge the gap between the requirement for hardware/firmware being as simple as possible from a security point of view and extensibility towards an arbitrary code executed on it (possible approaches are: swap in/out, homomorphic cryptography)?
- How to extend existing interfaces of tamper resistant hardware to allow such functionalities, including the capability for supporting secure boot?

What are the necessary steps to ensure the integrity and authenticity of code executed on a mobile platform?

- How to ensure code integrity and authenticity during boot time (including boot loader, kernel, libraries, and system applications)?
- Which combination of hardware components within mobile phones fulfils the requirements and is realistic for mobile phones (considering size, power efficiency, performance, and cost)?
- How to ensure integrity and authenticity of underlying hardware?

Current Status of Work

Until now, we were able to start with ground work of a software design which interacts with ETRH (see figure 1). As a first step we concentrated on Secure Elements (SE), an integrated circuit which comes with NFC [4], as technology for ETRH. Besides this design of architecture we also progressed in the perspective of these topics:

- First drafts for a authenticated secure channel protocol between third party applets and applications.
- Conceptual work in swapping memory storage and processing power.
- Ground work in the Secure Application Framework for transparent access to Embedded Tamper Resistant Hardware (incl. first performance tests).

Objectives for Doctoral School

At the Doctoral School of UbiComp, the main objectives are to get in touch with people who are doing research in the area of user interface design and cryptography related topics in the context of mobile or wearable devices. As one of our main vision is to bridge the gap between high security functionalities and usability, I would like to get input on improvements and approaches to identify

user-acceptance and sufficient transparency of our concept. I would also like to discuss other possible application areas for the ETRH in mobile other devices (for example in sensor networks for tamper resisting key storage). Additionally, I am interested to get in touch with experts in cryptography on devices with limited resources who can help us with current research problems in the area of homomorphic cryptography (to swap out processing power and memory space from the Secure Element) and authenticated *Elliptic Curve Diffie-Hellman* protocols.

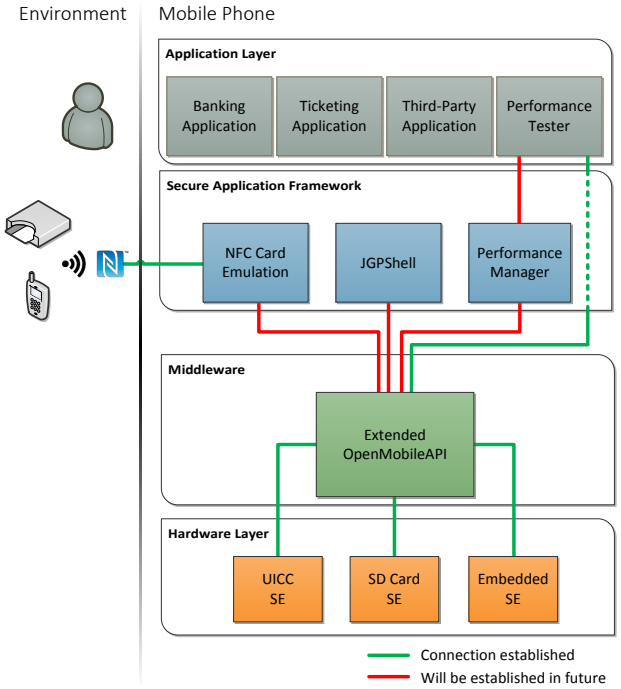

Figure 1: Architecture of the current status for an open ecosystem on Embedded Tamper Resistant Hardware.

Biographical Sketch

Michael Hölzl received his Master of Science in Engineering at the UAS Upper Austria in the Mobile Computing programme in 2012. His main research interests include security on mobile devices, mobile sensors analysis and mobile UI design for senior users. Since March 2013 he is enrolled in the PhD program of JKU Linz with expected date of completion in the end of 2016. Supervisors are **René Mayrhofer**, professor at UAS Upper Astria, and **Josef Scharinger**, professor at JKU Linz.

Acknowledgements

This work has been carried out within the scope of usmile, the Josef Ressel Center for User-Friendly Secure Mobile Environments. We gratefully acknowledge funding and support by the Christian Doppler Gesellschaft, A1 Telekom Austria AG, Drei-Banken-EDV GmbH, LG Nexera Business Solutions AG, and NXP Semiconductors Austria GmbH.

References

[1] Ben-Asher, N., Kirschnick, N., Sieger, H., Meyer, J., Ben-Oved, A., and Mller, S. *On the need for different security methods on mobile phones*. MobileHCI 11. ACM, 2011, 465–473.

[2] Höbarth, R. M. S., and Mayrhofer, R. A framework for on-device privilege escalation exploit execution on android. *Proceedings of IWSSI/SPMU* (2011).

[3] Khan, S., Nauman, M., Othman, A., and Musa, S. *How secure is your smartphone: An analysis of smartphone security mechanisms*. 2012, 76–81.

[4] Madlmayr, G., Langer, J., Kantner, C., and Scharinger, J. *NFC Devices: Security and Privacy*. 2008, 642–647.

[5] Miller, C. Mobile attacks and defense. *IEEE Security Privacy 9*, 4 (2011), 68–70.