A Path Towards Ubiquitous Protection of Media
(Position Paper)

Ronald Toegl, Johannes Winter, and Martin Pirker

Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology, Inffeldgasse 16a, A–8010 Graz, Austria
{rtoegl,jwinter,mpirker}@iaik.tugraz.at

Abstract. Mobile devices have become powerful and user-friendly. At the same time they have become a hosting platform for a wide variety of services. Naturally, the interests of the various stakeholders on a mobile platform are not the same. Thus, there is demand for a strict separation model of services on mobile devices. In this paper, we outline a possible approach to enable a Secure Media Path on mobile devices. Our approach aims to support the needs of the different stakeholders, with respect to openness, content protection and client privacy. The architecture takes into account the resource constraints of mobile devices.
Keywords: Trusted Computing, TrustZone, Multi Screen, Digital Rights Management

1 Introduction

Enabling a Secure Media Path (SMP) on mobile devices is a non-trivial problem, as such small platforms are restricted by multiple constraints. For example, such a device needs to be open enough to enable modifications or replacement of parts of the software stack, while at the same time the mobility aspect of the platform requires efficient adaption to changes in environment and connectivity.

A SMP aims to guarantee the protection of content, while at the same time needs to operate within the contraints of the mobile platform. On a high level, a SMP consists of software and hardware components that allow to protect content (and protection keys) and which enforce associated usage rules. The use of dedicated hardware elements and personalization capabilities make the bypassing of the content protection a significantly hard problem.

The basic platform requirement is isolation. The isolation (or virtualization) features of state-of-the-art CPUs and chipsets enable the hardware assisted separation of small, secure code and data proceesing from common rich OS code and data processing. A strictly reduced Trusted Computing Base (TCB) can be more easily checked for integrity and thus then tasked with the sensitive operations.

Outline The remainder of this paper is organized as follows. In Section 2 we motivate an example scenario for a SMP on modern mobile platforms. Then, Section 3 gives a short summary on the state-of-the-art of security enhanced hardware platforms. Section 4 presents our proposal for a SMP architecture. The paper concludes in Section 5.

2 A Future Scenario of Media Consumption

We imagine the capabilities of media consumption in the near-future and describe it in the following usage scenario: Amadeus has just bought a new smartphone. On the way home, he explores his new device. Among the pre-installed apps, Amadeus finds an app for a large cloud provider's payment service and its movie store. In the App Store, he finds additional applications to access his bank account at Sparkasse, a bitcoin client, a wallet app, and free apps from various broadcasters including free-tv and pay-tv networks.

He launches the Sparkasse App, which uses a specific, secure part of his device's screen to display the message provided by Sparkasse. He enters his account information to access his account. Next he checks the movies available in the pay-tv video store and picks one. Upon touch of the "Play" button, he is presented with a menu of payment options, including a service backed by a federated cloud ID, his account at Sparkasse, and bitcoins. Each account shows the available amount of money. Amadeus taps the Sparkasse account. When the screen changes, he immediately recognizes the visual brand of the Sparkasse app asking him to confirm the payment. The mobile phone indicates that the Sparkasse app is indeed the authentic origin of the payment dialogue. After confirmation, the movie starts to stream and play immediately. The bus arrives. Amadeus sits at the back of the bus and enjoys the first thirty minutes.

At home, he puts his new phone on the coffee table and switches on his smart TV set. As soon as switched on, he sees a menu where he can choose to continue to watch the current movie scene on the TV set as a 3D movie in 4k quality with surround sound, without a need to purchase the movie again. While the movie plays on the smart TV, the phone goes to sleep to conserve battery. His wife joins in to watch the second half of the movie, when Amadeus's phone rings. He picks the phone and goes to the other room to have the call. The smart TV keeps showing the movie so that Amadeus's wife keeps watching it.

This short story illustrates a number of elements not possible today: First, the consumer has full control over the selection of content and payment methods. For privacy-sensitive apps, security is made tangible to the user using a secure portion of the screen. Use credentials are protected from the rich OS installed on the mobile device. It is also important to the consumer that media delivery is seamless across different devices, and the mobile device can act as a media gateway for the home. Different devices that display the same media may offer device-specific enhanced experiences, and cooperate closely and seamlessly. Finally, the content that is consumed is well-protected. There is a strong separation between the protected media content on the device and any apps running on the rich OS. Without it, a pay-tv provider would not agree to stream their media to the device. Yet, content protection is transparent to the user. Equally important to the content providers: they can either provide their content to standard apps and service providers that handle payment in a transparent way, or provide their own apps that link into a secure media interface that is the same across devices. This set of features is not yet possible with current day devices.

3 Security Enhanced Platforms

Modern state-of-the-art platforms provide distinct security support features. They enable enhanced cryptographic primitives, strictly isolated processing and (remote) attestation of the platform state. We now give a short overview on these technologies.

The term *Trusted Computing* has been mostly established by specifications of the Trusted Computing Group (TCG), an industry consortium. The core component, the Trusted Platform Module (TPM) [14], is a low-cost hardware security module that is physically bound to its host device. A tamper-resilient integrated circuit contains implementations for public-key cryptography, key generation, cryptographic hashing, and random-number generation. The TPM provides high-level functionality such as collecting and reporting the current system state, and providing evidence of the integrity and authenticity of this measurement, known as *Remote Attestation*. Consequently, a sucessful TPM-enabled remote attestion of a platform can provide the confidence that the platform is in the correct state to be host for a secure media path environment.

3.1 ARM TrustZone

One of the dominant processor architectures employed in current mobile and embedded devices is the ARM architecture. Current ARM-based processor designs span a wide range of application fields, ranging from tiny embedded devices (e.g. ARM Cortex-M3) to powerful multi-core systems (e.g. ARM Cortex-A9 MPCore). Also, ARM introduced a set of hardware-based security extensions called TrustZone [2] to ARM processor cores and on-chip components.

The key foundation of ARM TrustZone is the introduction of a *secure world* and a *non-secure world* operating mode. This secure world and non-secure world mode split is an orthogonal concept to the privileged/unprivileged modes already found on earlier ARM cores. On a typical ARM TrustZone core, secure world and non-secure world versions of all privileged and unprivileged processor modes co-exist. For the purpose of interfacing between secure and non-secure world a special Secure Monitor Mode together with a Secure Monitor Call instruction exists. The AMBA AXI bus in a TrustZone enabled system carries extra signals to indicate the originating world for any bus cycles. Thus, TrustZone aware System-On-Chip (SoC) peripherals can interpret those extra signals to restrict access to secure world only; a secure world executive can closely monitor any non-secure world attempts to access secure world peripherals. To summarise, an ARM TrustZone CPU core can be seen as two virtual CPU cores with different privileges and a strictly controlled communication interface.

3.2 Trusted Execution Environments

Previously, ARM had published its own TrustZone software API specification [1]. Together with Trusted Logic, ARM has developed a closed-source TrustZone software stack, complementing the TrustZone hardware extensions. ARM has

since donated its TrustZone API to the GlobalPlatform industry association and this has developed into the Trusted Execution Environment (TEE) Client API [5]. It allows an application in the "non-secure world", which typically runs a rich-OS such as Google Android or Microsoft Windows Mobile 8, to communicate with the "secure world". ARM has also been working with other companies to develop the TEE Internal API [6] that interfaces between a Trusted OS, running in the secure world and a Trusted Application.

Today, all modern ARM-based Smartphones (Cortex-A CPU based) include a TEE based on SoCs by manufacturers like Qualcomm, Samsung, Nvidia, and Texas Instruments. Accordingly, TEEs are already deployed on the field since for several years, featuring Trusted OSes currently made by Trusted-Logic/Gemalto (Trusted Foundation) or Giesecke & Devrient (Mobicore). Moreover, ARM, Gemalto and Giesecke & Devrient and others have recently created the "Trustonic" Joint Venture on TEE Trusted OS and its ecosystem of services.

3.3 Research on TEEs and TEE Applications

Several scientific publications deal with proposals for secure mobile and embedded system designs based on the ARM TrustZone security extensions. Use of ARM TrustZone hardware to securely manage and execute small programs ("credentials") were described in [9] and [3]. A similar runtime infrastructure was used by the authors of [4] to implement a mobile trusted platform module. Similarly [12] proposes a trusted runtime environment utilizing Microsoft's .NET Framework inside the TrustZone secure world. With the use of a managed runtime environment the authors try to benefit from the advantages of a high-level language combined with hardware security and isolation mechanisms provided by the underlaying platform.

A large number of publications deal with possible applications of ARM TrustZone to implement, for example, digital rights management [8], cryptographic protocols [15], mobile ticketing [7] and [10], wireless sensor networks [17], or anonymous payment for remote cloud service resource consumption [11].

An approach of using a modified Linux kernel acting as secure world operating system for a mobile virtualization scenario has been discussed in [16]. This work showcases an experimental open-source software environment for experiments with ARM TrustZone in combination with Trusted Computing primitives. The software framework offers a prototype kernel running within a trusted environment and features a software based Trusted Platform Module hosted in a TrustZone protected runtime environment and an Android operating system accessing it through a high-level API.

4 Proposed Architecture

Media processing is generally a resource intensive task with high demands of processing power memory and bandwidth, especially with high definition material. Traditional, stationary set-top boxes employ various types of smart cards in

combination with specialized system-on-chip and board-level designs to provide adequate performance as well as protection of content data, which is delivered and processed on the device. Commonly, these traditional set-top boxes are closed special-purpose embedded systems with well-defined restrictions on the software and configuration changes an end-user of the device is able to perform. However, on smart phones and tablet computers, users expect to be able to customize their devices to a great degree, for example by installing all kinds of third-party applications.

Typical transformations on the stream include signal processing tasks like decompression, color-space conversions, equalization of audio signals, and scaling or rotation of video signals. Current mobile computing platforms often implement at least parts of these computationally intensive tasks directly in hardware to reduce the computational requirements and power-consumption of the platform. To support secure media paths, it is necessary to securely integrate additional transformation steps in the basic architecture. Such steps include content decryption and surrounding frameworks like policy engines and key management. To avoid unintended and unwanted interference between arbitrary applications running on the platform and the SMP core services, it is necessary to introduce two separate security domains on the platform. Due to the bandwidth requirements of high-quality video content, encryption algorithms may be moved into dedicated hardware blocks.

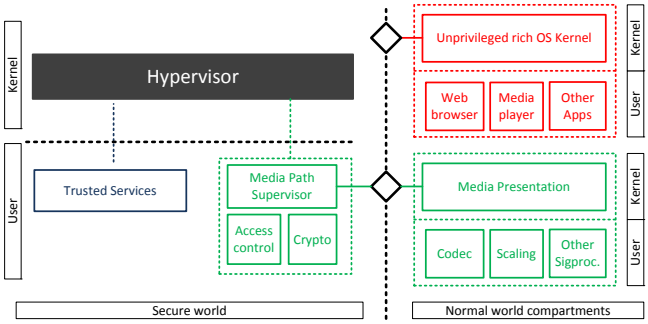

Fig. 1. Layout of the proposed SMP architecture.

We propose to leverage the TrustZone hardware-extensions to establish a software-based SMP. In our proposed architecture, the components constituting the core implementation of the SMP itself are protected against direct interference from malicious applications through software and hardware isolation, and virtualization techniques. By executing the content processing in domains isolated from the rest of the system, the use of media-processing software components provided by the content provider along with the content become possible. Because media processing is isolated from the rich OS, the interests of the content provider to protect their content from piracy are preserved. At the same

time, such software is not able to subvert the security and privacy of the user because it can access the relevant parts of the media pipeline only. Secure handover between devices can be supported by remote attestation, which can also be easily done over Bluetooth or Near-Field-Communications [13].

ARM TrustZone divides the platform into multiple worlds. The so-called secure world is controlled by the highly secure and low-complexity trusted OS. Besides the trusted OS, the platform executes one or multiple instances of a rich OS such as Android in the so-called normal world. There, playback is controlled and presented to the user either by specialized apps, or simply in the HTML5 compliant web browser. Thereby, our approach retains compatibility with current mobile operating systems. Because the secure world is hidden from any software executed in the normal world, information that is critical for security and privacy can be protected by processing it in the secure world only. Furthermore, hardware components that are critical for the SMP can be explicitly assigned to the secure world, eliminating attack vectors for sniffing high-value content from the normal world.

Because the rich operating system cannot be assumed to be free of security-critical bugs, it is necessary to address the challenge of a secure channel to protect the integrity of user input passed to trusted apps.

5 Conclusions

We presented our vision and proposal for protecting the presentation of media in highly mobile and interactive systems. Our approach is motivated through a future usage scenario which illustrates the interaction of users with several platforms that seamlessly distribute high-fidelity media. We have reviewed the state-of-the art of TrustZone-enabled systems and proposed to leverage it to establish secure media paths.

For the future we would like to encourage the community to work together to reach the manifestation of this vision.

Acknowledgments. This paper presents an idea and approach that was contemplated together with Roderick Bloem and Christian Schwarz. This work was supported by the EC, through project FP7-ICT-STANCE, grant agreement number 317753, and project DALIA of the AAL joint programme.

References

1. ARM Limited: TrustZone API Specification v2.0 (June 2006), pRD29-USGC-000089
2. ARM Limited: ARM Security Technology Building a Secure System using TrustZone Technology. `http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf` (2009), pRD29-GENC-009492C

3. Ekberg, J.E., Asokan, N., Kostiainen, K., Rantala, A.: *Scheduling execution of credentials in constrained secure environments.* In: Proceedings of the 3rd ACM workshop on Scalable trusted computing. pp. 61–70. STC '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1456455.1456465
4. Ekberg, J.E., Bugiel, S.: *Trust in a small package: minimized MRTM software implementation for mobile secure environments.* In: Proceedings of the 2009 ACM workshop on Scalable trusted computing. pp. 9–18. STC '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1655108.1655111
5. GlobalPlatform: TEE Client API Specification v1.0. http://www.globalplatform.org/specificationsdevice.asp (July 2011)
6. GlobalPlatform: TEE Internal API Specification v1.0. http://www.globalplatform.org/specificationsdevice.asp (December 2011)
7. Hussin, W.H.W., Coulton, P., Edwards, R.: *Mobile Ticketing System Employing TrustZone Technology.* In: Proceedings of the International Conference on Mobile Business. pp. 651–654. IEEE Computer Society, Washington, DC, USA (2005), http://dl.acm.org/citation.cfm?id=1084013.1084282
8. Hussin, W.H.W., Edwards, R., Coulton, P.: *E-Pass Using DRM in Symbian v8 OS and TrustZone: Securing Vital Data on Mobile Devices.* Mobile Business, International Conference on 0, 14 (2006)
9. Kostiainen, K., Ekberg, J.E., Asokan, N., Rantala, A.: *On-board credentials with open provisioning.* In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. pp. 104–115. ASIACCS '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1533057.1533074
10. Pirker, M., Slamanig, D.: A Framework for Privacy-Preserving Mobile Payment on Security Enhanced ARM TrustZone Platforms. In: Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. pp. 1155–1160. TRUSTCOM '12, IEEE Computer Society, Washington, DC, USA (2012)
11. Pirker, M., Slamanig, D., Winter, J.: Practical Privacy Preserving Cloud Resource-Payment for Constrained Clients. In: PETS 2012. LNCS, vol. 7384, pp. 201–220. Springer Verlag (2012)
12. Santos, N., Raj, H., Saroiu, S., Wolman, A.: *Trusted Language Runtime (TLR): Enabling Trusted Applications on Smartphones* (2011)
13. Toegl, R., Hutter, M.: An approach to introducing locality in remote attestation using near field communications. The Journal of Supercomputing 55(2), 207–227 (2011), http://dx.doi.org/10.1007/s11227-010-0407-1
14. Trusted Computing Group: *TCG TPM Specification Version 1.2 rev 113* (2011), https://www.trustedcomputinggroup.org/developers/
15. Wachsmann, C., Chen, L., Dietrich, K., Löhr, H., Sadeghi, A.R., Winter, J.: *Lightweight Anonymous Authentication with TLS and DAA for Embedded Mobile Devices.* In: Burmester, M., Tsudik, G., Magliveras, S., Ilic, I. (eds.) Information Security, Lecture Notes in Computer Science, vol. 6531, pp. 84–98. Springer Berlin / Heidelberg (2011), 10.1007/978-3-642-18178-8_8
16. Winter, J.: Trusted computing building blocks for embedded linux-based arm trustzone platforms. In: Proceedings of the 3rd ACM workshop on Scalable trusted computing. pp. 21–30. ACM, Alexandria, Virginia, USA (2008)
17. Yussoff, Y.M., Hashim, H.: *Trusted Wireless Sensor Node Platform.* In: Ao, S.I., Gelman, L., Hukins, D.W., Hunter, A., Korsunsky, A.M. (eds.) Proceedings of the World Congress on Engineering 2010 Vol I, WCE '10, June 30 - July 2, 2010, London, U.K. pp. 774–779. Lecture Notes in Engineering and Computer Science, International Association of Engineers, Newswood Limited (2010)