Trusted Mobile Devices: Requirements for a Mobile Trusted Platform Module

Kathleen N. McGill

In recent years, mobile devices have replaced desktop PCs as the computing platform of choice for many users. Unfortunately, most mobile devices lack the security measures, such as hardware roots of trust, available in more traditional business-class computing platforms. Researchers at APL are working with the Trusted Computing Group (TCG) to develop specifications for trusted computing technologies in mobile devices. These technologies will enable desirable security properties, such as device integrity and protected storage, in mobile devices. In this article, we provide an analysis of the difficulties that must be overcome by those specifications. We describe key features of trusted mobile devices: roots of trust, the Trusted Platform Module (TPM) Mobile host environment, and the Secure Boot mechanism. We present and analyze an example implementation of a near-term trusted mobile phone. Finally, we outline the TPM Mobile roadmap to bring trusted mobile devices to market.

INTRODUCTION

In recent years, mobile devices have replaced desktop PCs as the primary computing platform for many users. This trend is encouraged by convenient access to bank accounts, personal networks, and a wide range of networked resources through our tablets and mobile phones (see Fig. 1). Many organizations would like to use mobile devices in the work environment as a cost-savings and efficiency measure. Organizations may also wish to allow employees to use their personal mobile devices to access enterprise resources, an initiative commonly called "Bring Your Own Device."

Because most mobile devices lack the security measures available in more traditional computing platforms, enterprises are concerned about associated risks of integrating mobile devices into their networks. For example, most mobile devices do not include the hardware roots of trust that are built into traditional business-class platforms. These hardware roots of trust are the foundation of trust in any platform and enable security properties for protection-conscious enterprises that wish to use them.

The National Institute of Standards and Technology (NIST) recommends a set of desired capabilities for

Figure 1. Mobile phone applications for enterprise and banking services. (Image on left reprinted from Ref. 1.)

evaluates this evidence to determine whether the device is in an expected state. If the appraiser is satisfied, the attestation passes, and the appraiser grants access to the resource. This scenario and its variants are frequently used within enterprises and the U.S. government to control access to their networks.

Many traditional business-class computing platforms provide desired security capabilities through a hardware root of trust called the Trusted Platform Module (TPM). A TPM is a discrete hardware component of a platform that provides secure generation and management of cryptographic identities, isolated processing resources, and protected storage for sensitive data.[3] The TPM specification was developed by the Trusted Computing Group (TCG), an international board that aims to define and promote open industry standards for interoperable trusted computing platforms.[4]

A TPM bolsters an attestation by vouching for the authenticity of the measurement data and binding the attestation to a particular device and its software configuration. The TPM storage protects measurement data collected on the device. Typically, these data include hashes of the software that booted on the device. The TPM can provide a report of the measurement data for an attestation and sign the data with a private attestation key protected by the TPM. The attester provides the signed data and a credential for the TPM attestation key to the appraiser. This credential allows the appraiser to verify the authenticity of the measurement data and to verify that the private key used to sign the data is protected by a genuine TPM.

security-enabled mobile devices[2]: device integrity, isolation, and protected storage. A device has integrity if its software, firmware, and hardware configurations are in an expected state. Device isolation provides separate domains for computations and data owned by different entities, or stakeholders, on the same device. Protected storage enables mobile devices to protect the confidentiality and integrity of data on the device while in use and at rest.[2]

A mobile device should be able to provide evidence of device integrity to a third party by some form of device attestation.[2] An attestation may be described as a claim made by a device, the attester, about its properties to a third party, an appraiser, and provides evidence to support that claim. The appraiser makes a decision about the attester based on the provided evidence. Figure 2 displays a common device attestation scenario. The attester is a mobile device that seeks access to some protected resource. The appraiser is a remote server that serves as a gatekeeper for the protected resource. First, the device requests access to the resource. The appraiser requires a device attestation. In this attestation, the mobile device will provide some measurement data, which represent the state of the device as evidence of its integrity. The appraiser

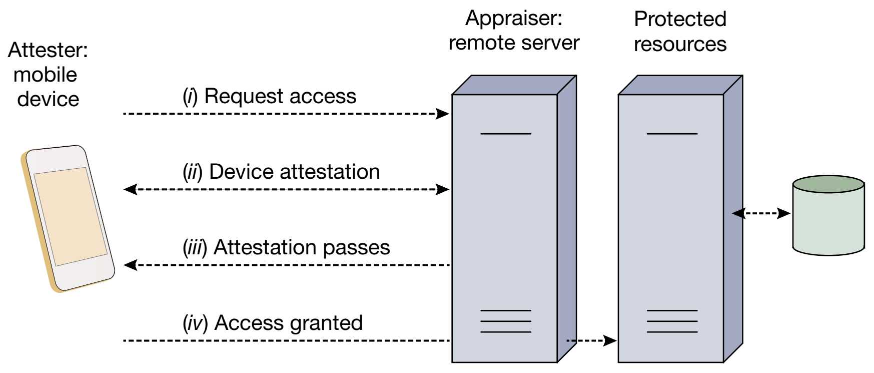

Figure 2. Device attestation.

Many mobile devices have space, power, and cost constraints, which have hindered the adoption of a discrete TPM chip. However, recently, new mobile technologies have emerged, enabling alternative implementations of the traditional TPM chip in the mobile arena. For example, the ARM TrustZone technology uses hardware mechanisms to partition the device into secure and nonsecure resources.[5] Similar to a TPM, these mechanisms may be capable of protecting trusted components within the secure resources.

Researchers at APL are working to enhance existing specifications for trusted computing technologies in mobile devices. APL is working within the Mobile Platform Working Group to specify a mobile adaptation of the TPM—called the TPM Mobile—and a Reference Architecture. The objective of this work is to encourage trusted mobile computing technologies in the market. This goal targets the growing need for trusted mobile devices at APL, within the U.S. government, and throughout the nation.

This article provides an introduction to the TPM Mobile requirements and an analysis of the problems such a specification must overcome, with an emphasis on APL's contributions. The article begins with an introduction to trust concepts in mobile devices. The article presents definitions for mobile roots of trust that were developed in collaboration between APL and members of the U.S. government. Then key features of trusted mobile device architectures are described: the TPM Mobile, its host environment, and secure boot of the mobile device. APL drafted the host environment section in the Reference Architecture specification and worked with members of the Mobile Platform Working Group to resolve issues that would impact market adoption. Finally, the article presents an example and analysis of a TPM Mobile that is implemented using a combination of hardware, firmware, and software. APL developed this model for use in the final specification to serve as an example for implementers. The model is also used to demonstrate issues the specification must resolve.

TRUST CONCEPTS

Roots of Trust

The security services of a mobile device are enabled by roots of trust. A root of trust is a hardware, firmware, and/or software component that is inherently trusted to perform security services on a device. Roots of trust must behave as expected because their misbehavior cannot be detected.[6] For this reason, roots of trust must be designed securely. There must be mechanisms in place to protect the roots of trust from unintentional or malicious action. If a root of trust can be modified, those modifications must be made using a secure update process that includes cryptographic owner authentication of the root of trust or a secure local mechanism. Finally, a root of trust should include as minimal functionality as is possible to reduce the attack surface of trusted code.[2]

The following roots of trust provide the security capabilities of a trusted mobile device.

- **Root of trust for confidentiality (RTC):** The RTC provides locations for protecting confidential data such as private keys and random number generator states. It also must include a protected interface that restricts access to and modification of these data.[2]

- **Root of trust for integrity (RTI):** The RTI provides locations for protecting integrity-sensitive data. It also includes an interface that restricts access to integrity parameters. The most significant difference between the RTI and the RTC is that the RTC protects secrets while the RTI protects values that are meant to be shared. They require different authorized interfaces. In traditional TCG terminology, the RTC and the RTI combine to form the root of trust for storage.[2]

- **Root of trust for reporting (RTR):** The RTR provides authenticity and nonrepudiation services for use in attestation. This is typically represented by a signing key that is unique to the device. A typical use of the RTR is to sign integrity data during an attestation of the device.

- **Root of trust for measurement (RTM):** The RTM performs integrity measurements on the device. Usually, an RTM measurement is a cryptographic hash of a software image.

- **Root of trust for verification (RTV):** The RTV verifies the integrity of software and data. This can be done either with a digital signature using a public key stored securely on the device, or by comparison against reference values stored securely on the device.

- **Root of trust for update (RTU):** Because it is impractical to ship commercial devices without a means of updating them in the field, a means must be available to secure the update mechanism used on roots of trust. The RTU must verify that an update is authentic and not allow an update to proceed unless an update is verified. Additionally, it must provide protection against unauthorized rollback of updates, once installed.

Figure 3 displays an example of the interactions between the roots of trust that provide services to support an attestation. First, the RTM performs an integrity measurement of some piece of software on the device.

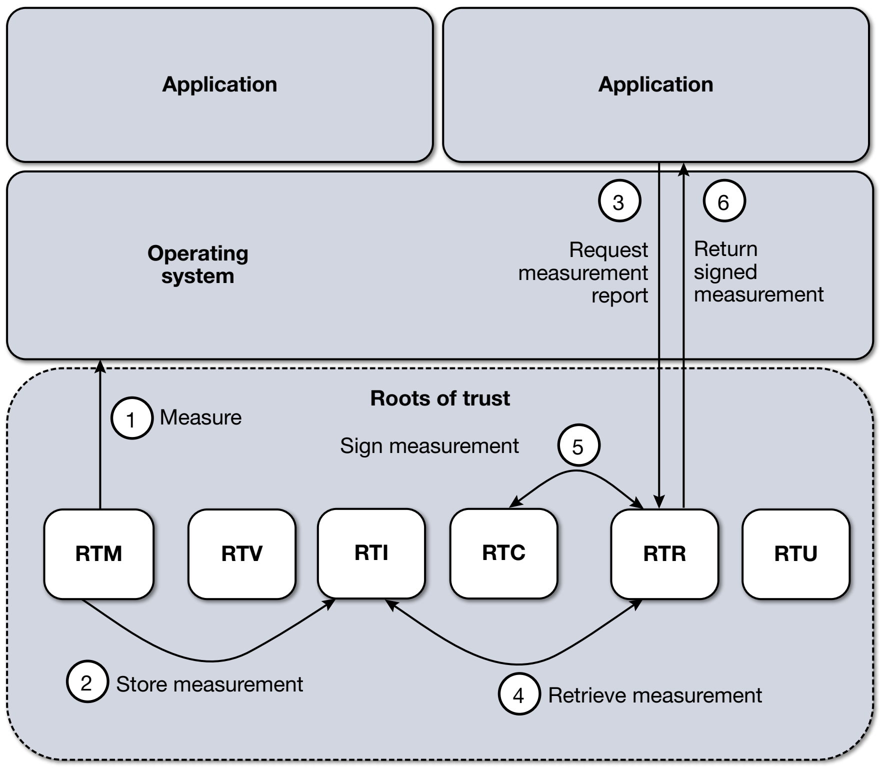

Figure 3. Root of trust interactions for an attestation.

This measurement may occur during the boot cycle or upon request after the device has booted. The RTM stores the measurement in the RTI. During an attestation, an application on the device requests a signed report of the integrity measurement from the RTR. The RTR retrieves the integrity measurement from the RTI and signs the measurement with the identity in the RTC. Finally, the RTR returns the signed measurement to the requesting application for use in the attestation.

Ideally, roots of trust are implemented in dedicated hardware, or at least protected by hardware mechanisms. Dedicated hardware, such as TPM 1.2 implementations, are hardware that are only used by designated entities and are isolated from other entities of the device.

Initial TPM 2.0 implementations today are run from firmware—sometimes as an application running TrustZone. Because TrustZone runs on the same CPU as the main application space, this requires a shift in emphasis for TPM Mobile from requirements of dedicated hardware to requirements of the properties of TPM Mobile's host environment and mobile roots of trust. This shift will enable alternative implementations of trusted mobile devices with minimal dedicated hardware.

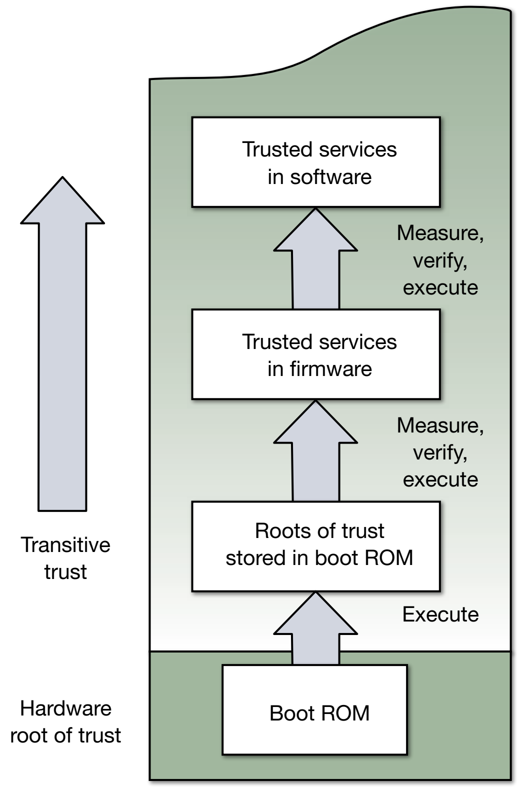

Figure 4. Transitive trust in mobile devices. ROM, read-only memory.

Transitive Trust

Transitive trust is crucial to enabling alternative implementations of trusted mobile devices. A root of trust can delegate its services to another entity on the device. In this case, a root of trust provides a transitive chain of trust to the entity to establish trustworthiness of that entity. In order to extend a transitive chain of trust to an entity, the entity must first be measured and verified by a trustworthy entity. These delegates of the roots of trust are referred to as trusted services. Figure 4 displays a notional transitive chain of trust. The transitive trust begins with a hardware root of trust, such as code in ROM of the device. The code in ROM measures and verifies the next piece of code in the boot cycle, such as other firmware. After the firmware is verified, it is allowed to execute. This measure–verify–execute sequence continues as each code module measures and verifies the next code module before passing execution.

The transitive trust extends as far as this sequence is applied. Transitive trust enables flexibility in implementing trusted services. The roots of trust can delegate services to other entities on the device while preserving the trustworthiness derived from the original hardware root of trust.

TPM MOBILE ARCHITECTURE

TPM Mobile

The challenge of the TPM Mobile specification is to adapt the TPM 2.0 specification for mobile devices. The goal of the TPM Mobile specification is to enable implementation on a wide range of mobile devices,[6] and much of this goal will be achieved by clearly defining requirements of the architecture in which the TPM Mobile is deployed. Whether a TPM implementation is implemented in hardware or in firmware, isolation of the TPM implementation must be maintained—it must have its own execution and memory resources to prevent exposure of secrets that it is tasked to maintain.

A defining feature of mobile devices is that there are usually multiple stakeholders on a single device—each demanding private data and processing resources. This can be achieved in a TPM 2.0 environment in two ways—either by using separate key chains rooted in separate storage root keys that coexist on the TPM or by having several TPM implementations on the same device. To maintain compatibility for software created to be used on personal computers or mobile devices, the TPMs instantiated on those devices must also be compatible. Therefore, if multiple TPM implementations are to be used, it is important that one of those TPMs be set apart to be used for attesting to the device integrity and that this TPM be compatible with existing server and desktop systems that use TPMs.

Host Environment

The host environment of a TPM is the isolated execution environment in which it runs. The isolated execution property of the host environment distinguishes it from the device environment as a whole. In the case of the discrete TPM chip, the chip is the host environment. In cases where isolation is implemented in hardware and/or software, the architecture of the device determines one or more host environments within the device. The roots of trust of the device reinforce the host environment by supporting the required properties. NIST has published those requirements that must be present to provide hardware rooted security in a TPM Mobile host environment.[2] The implementation of the host environment may take any form provided that it satisfies these requirements.

Minimum Requirements of a TPM Mobile Host Environment

The following are requirements for a TPM in a mobile host environment. These requirements can be met with either a hardware or software TPM.

1. The TPM's execution environment must interface only with other execution environments through standard protected interface APIs, as defined in the TPM 2.0 specification. This protects TPM resources during execution.
2. The TPM must have a unique cryptographic identity. This is necessary to prevent spoofing during attestation.
3. The TPM must have a means of protecting its resources when it is not executing. No untrusted or unverified code should have access to TPM secrets.
4. The TPM must have a means to prevent rollback of its resources when it is not executing.
5. The TPM must be uniquely attached to the device. This is necessary to allow the TPM to perform device attestation.
6. The TPM must have its code integrity protected while it is executing. This requires a strong host environment isolation mechanism.
7. The TPM must have its code integrity protected while it is not executing. This is typically done by verifying the integrity during initial program load of both the TPM code and any other code that must be trusted with TPM secrets.
8. The TPM must provide a mechanism that allows attestation of the version of both TPM code and any other code that must be trusted to maintain TPM integrity.
9. The execution environment must provide resources necessary for the TPM to meet the requirements of the TPM 2.0 specification.

Implementation Options

Several implementation options are available for a TPM Mobile and its host environment. We typically refer to different implementations as "hardware," "software," or "firmware" implementations. Although all implementations include software/firmware and the hardware on which it executes, these terms aim to differentiate the amount of dedicated hardware in each implementation. In general terms, we differentiate firmware from software by where it is stored on the device. Firmware is typically stored in nonvolatile ROM.

A hardware of firmware implementation of a TPM Mobile provides the best security. A software TPM Mobile shares the same processing resources as the rest of the mobile device. Any isolation properties are achieved through software virtualization mechanisms in the operating system (OS) or hypervisor and are based on trust that the operating system or hypervisor itself provides. There is skepticism within the trusted computing communities that this approach can provide the necessary security capabilities. Trust must be placed in a complex software solution that supports the general purpose of the device as well as the trusted services of a TPM Mobile. History has shown that general-purpose software is likely to include bugs, which lead to unexpected software states. It is usually infeasible to verify that software of this complexity will behave as expected.

Hardware TPM Mobile

The hardware TPM Mobile is a traditional TPM chip. The TPM Mobile includes a dedicated processor and storage resources that are physically distinct from the other resources of the device. This physical distinction provides protection from other resources on the device and allows for certification of the TPM independent of the rest of the device. The main disadvantage of the hardware TPM Mobile is the requirement of dedicated hardware in a resource-constrained device.

Firmware TPM Mobile

A firmware TPM Mobile is an implementation in which software isolation is enforced by the hardware architecture of the system-on-a-chip. The advantage of this approach is that the TPM Mobile does not require as many dedicated resources as a traditional TPM. The TPM Mobile uses the main processor on the device and some of the same hardware roots of trust on the system-on-a-chip.

The firmware implementation has been enabled by emerging hardware and software technologies. For example, the ARM TrustZone technology includes processor extensions and hardware modules to allow manufacturers to partition the system-on-a-chip into secure and nonsecure resources.[5] Software technologies have been developed to use the TrustZone extensions to provide an isolated execution environment within these devices. Figure 5 displays the notional software architecture of a TrustZone device.[5] With the TrustZone extensions, the processor has two contexts: normal and secure. The hardware restricts access to secure world resources—such as ROM, random-access memory (RAM), and nonvolatile storage—based on the processor context. Monitor software controls context switching between the normal and secure worlds. The end result is an isolation boundary between the normal and secure worlds that resembles a dedicated hardware solution.

GlobalPlatform, another international standards board, has developed a specification for implementing an isolated execution environment, called a Trusted Execution Environment (TEE).[7] A TEE is an execution environment that runs on the same hardware as the Rich Execution Environment (REE). The REE is a more versatile environment, such as the Android environment, with support for a wide range of application features. The GlobalPlatform TEE is specified with a set of security capabilities, such as rigid access control of internal data and functions and resistance to a defined set of threats.[7]

Because many of the properties of the TEE are also required of a TPM Mobile host environment, the TCG and GlobalPlatform have collaborated to develop standards that are consistent with each other. Many developers believe a firmware implementation of this kind is the most likely implementation of the TPM Mobile in the near future, and AMD has announced that it plans on having a TPM running in TrustZone in its new processor.

Secure Boot

Secure Boot is the fundamental mechanism that ensures mobile device integrity. During the boot cycle, each software image is verified before execution so that the device never boots into an unexpected state. The device manufacturer is responsible for implementing a Secure Boot mechanism, which prevents malicious or unintentional compromise of devices in the market.

There are several terms associated with Secure Boot, and they have different meanings in different communities. The following terms are used to describe three types of Secure Boot used in trusted mobile devices:

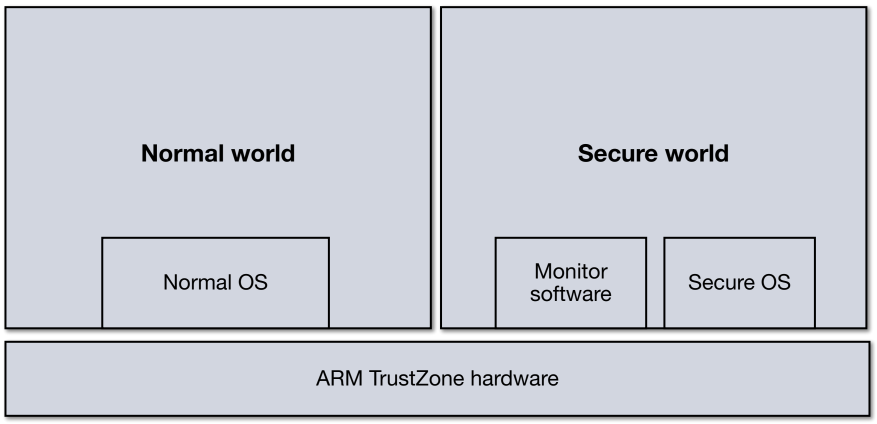

Figure 5. The notional software architecture of an ARM TrustZone device.[5]

- **Secure Boot:** Secure Boot begins with the execution of immutable code from a fixed location, typically ROM. Each module of code that is executed verifies the integrity of the next module of code before loading it for execution. If any integrity check fails, the boot will fail. Secure Boot does not store evidence of the integrity checks on the device.

- **Certified Boot:** Certified Boot combines Secure Boot with an attestable record of the modules executed during the boot sequence. Because code needs to be run in order to do this measurement, the integrity of at least the initial measurement code is inferred from integrity guaranteed by the Secure Boot. Additionally, during the boot sequence these measurements are compared to Reference Integrity Measurements (RIMs) to verify the integrity of the code before it is launched. Preferably, the measurement is a hash, in which case the reference data are usually generated using a digital signature of the module and a trusted certificate. If an integrity check fails, the boot will fail. The integrity measurements of a Certified Boot are available for future integrity reports of the device.

- **Measured Boot:** Measured Boot is also conducted with a TPM Mobile. Typically, Measured Boot begins after Certified Boot; therefore, the initial module is measured by a module measured during Certified Boot. Each module of code is measured before execution, and the measurements are stored in integrity-protected locations within the TPM Mobile. This code is executed without verifying the integrity of the measurements. Although the integrity measurements are still available for future integrity reports, they are never used to prevent the device from booting.

Secure Boot begins when the device is powered on with the execution of known code at a fixed location. When a firmware TPM is implemented, it cannot be used to store integrity measurements until it is instantiated. Therefore, a Secure Boot is used to verify all of the code that is initialized before the TPM Mobile functionality is established. Once the TPM Mobile is functioning, the boot proceeds with Certified Boot using the TPM to record integrity measurements. Typically, Certified Boot would be used to verify, at minimum, the rich OS image. The extent to which Secure Boot mechanisms are applied to the rich software stack may vary from device to device. Device manufacturers may use Certified Boot or Measured Boot to extend integrity measurements to driver, interpreter, and application images.

The Secure Boot mechanism must be derived from a hardware root of trust. The first piece of code executed during Secure Boot is typically in on-chip ROM. It is either immutable or can be modified only through a secure update process. This code must be trustworthy because it enforces Secure Boot and serves as the root of the chain of trust for the device. This code in on-chip ROM serves as a hardware root of trust. Its integrity is ensured by its placement in on-chip ROM, where attacks are particularly costly and beyond the scope of many malicious actors.

A TPM MOBILE EXAMPLE IMPLEMENTATION

Figure 6 shows a notional example of a TPM Mobile with a TEE-style host environment that identifies the hardware and software components of the roots of trust and the associated trusted services.

- Upon power being applied to the device, the system begins its boot from ROM memory. At this point, software has protected access to a public key, replay protected memory blocks (RPMB), and a symmetric key burned into eFuses on the device. It uses the public key to verify code stored outside the chip before execution, thus providing a root of trust for verification for the Secure Boot mechanism.

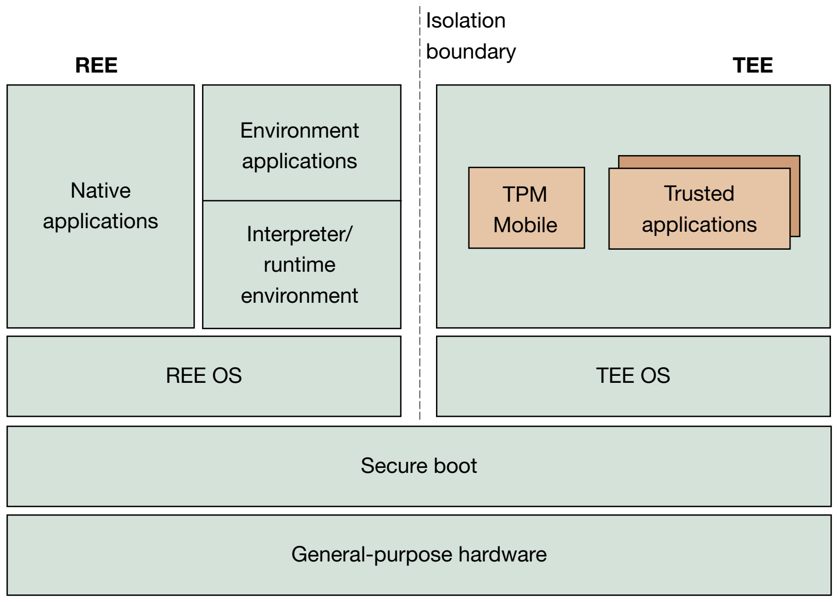

Figure 6. Mobile device architecture of a firmware TPM Mobile implementation using a TEE-style host environment.

- After verifying this code, this code initializes the TEE environment.
- The TEE environment in turn verifies the Mobile TPM code before allowing it to run in that TEE environment. The symmetric key is used to provide confidentiality mechanisms to the TPM for data at rest, and RPMB is used to provide replay-based protection to TPM storage.
- At this point, the TPM services can be used for a RTI, a RTC, and a RTR. Therefore a Certified Boot can then proceed. At this point, the TEE can provide measurement services, allowing the REE OS to be both verified and measured and then launched.
- Once the REE establishes access to the TPM, measurement services can be provided by the REE OS using the Mobile TPM-provided RTI services.

This example demonstrates a trusted mobile device implementation that can be achieved with near-term technologies. The TCG specifications must clearly outline required properties of the host environment, Secure Boot, and other features to accomplish this. Many requirements, such as the availability of a unique device identity, evidence of device firmware and TEE environment, and restrictions on the debug facilities, are either absent or inadequately specified in existing mobile device literature. Overall, this implementation of the TPM Mobile specifications would provide a more trustworthy device than that which is available in the market today.

Analysis of Potential Issues

Because the protections afforded by firmware TPMs are different than those provided by hardware TPMs, more analysis is required when considering TPM usage in mobile devices. Issues arise because, although hardware TPMs can incorporate firmware and the entire hardware TPM chip can be certified and thus have proven integrity, TPM firmware that is not stored and executed within a TPM chip cannot be certified in the same way. The discussion below focuses on ensuring that adequate consideration is given for how to protect the TPM functionality stored in firmware but running on general-purpose hardware. For clarity, an example is provided using a firmware TPM running in TrustZone.

If a firmware TPM is run in a TrustZone environment, the amount of code that must have its integrity protected by the Secure Boot mechanism presents a potential issue. In a traditional TPM, a Measured Boot is used, and integrity measurements taken during the boot cycle are stored in the TPM. The RTM is a small section of code in the BIOS, and the TPM itself can be certified via Common Criteria (EAL 4+) or FIPS. With a firmware TPM, the code that forms the RTM includes the code launching the TEE environment, the TEE environment, and also the TPM firmware itself. None of this code can be attested to, but its integrity must be trusted.

Certificates for hardware TPM's endorsement keys are produced that include unique certificates that attest to the Common Criteria or FIPS qualifications of the device. Platform certificates are used to provide evidence that the TPM is attached to a particular device with a RTM correctly implemented. In a mobile device, similar certifications could be used as an alternative to attestation for code that runs before the instantiation of the TPM. Such certificates would have to have similar properties—they must be unique to the device to prevent Break Once Run Everywhere (BORE) attacks. They must show that the protections on that code were correctly implemented and certify that the code has passed some sort of evaluation.

Absent such certificates, there is no trustworthy evidence of the code verified during Secure Boot that can be reported in an attestation. As a result, all of the code

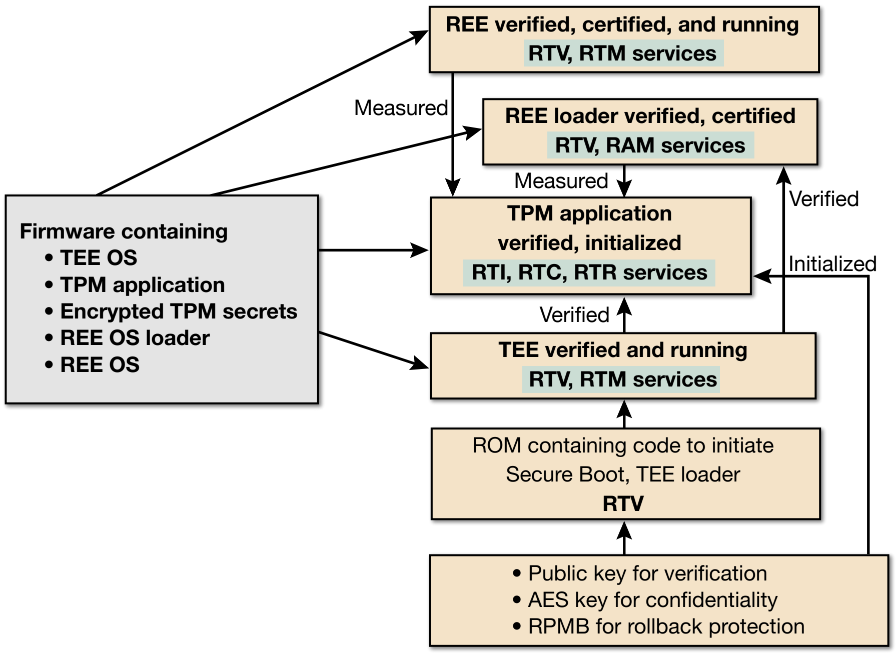

Figure 7. Trusted mobile device with a TEE-style host environment.

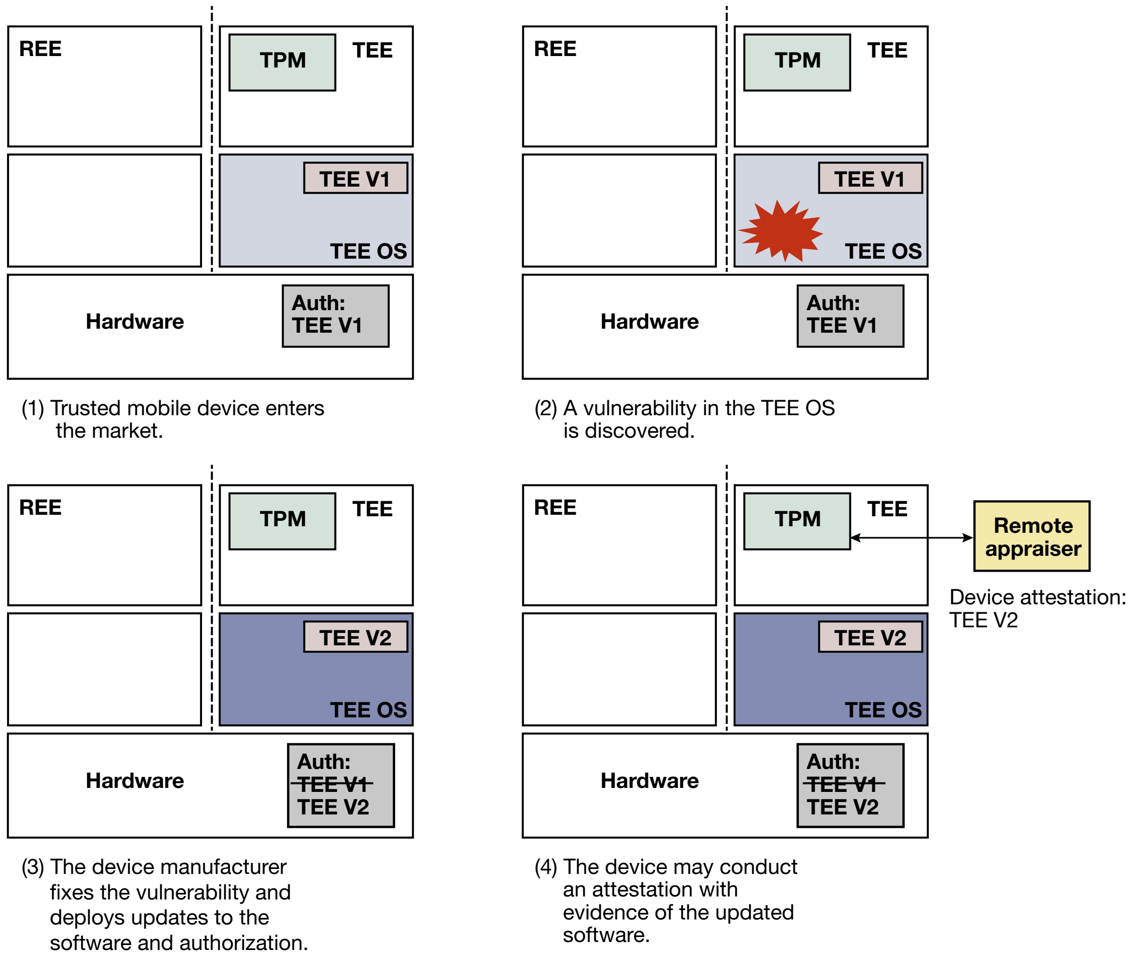
REE
TPM
TEE
TEE V1
TEE OS
Hardware
Auth: TEE V1
(1) Trusted mobile device enters the market.
REE
TPM
TEE
TEE V1
TEE OS
Hardware
Auth: TEE V1
(2) A vulnerability in the TEE OS is discovered.
REE
TPM
TEE
TEE V2
TEE OS
Hardware
Auth: ~~TEE V1~~ TEE V2
(3) The device manufacturer fixes the vulnerability and deploys updates to the software and authorization.
REE
TPM
TEE
Remote appraiser
Device attestation: TEE V2
TEE V2
TEE OS
Hardware
Auth: ~~TEE V1~~ TEE V2
(4) The device may conduct an attestation with evidence of the updated software.

Figure 8. The state of a deployed mobile device in which a vulnerability is discovered and remedied using a secure update process.

associated with Secure Boot must be trusted to behave as expected without evidence that it is trustworthy. Unfortunately, this code includes rich functionality, significant control of the device, and a larger memory footprint than the roots of trust in traditional computing, as indicated by the shaded regions in Fig. 7.

The following example demonstrates the risk associated with the Secure Boot requirements absent of a unique certificate. Figure 8 depicts the state of a deployed mobile device in which a vulnerability is discovered and remedied using a secure update process. (1) A mobile device enters the market with a TEE-style TPM Mobile implementation. The device manufacturer has implemented Secure and Certified Boot with the appropriate certificates to authorize the TEE. The manufacturer has included a certificate in the TEE OS image to provide evidence that the device is running version 1 of the TEE. (2) While the device is in the market, a vulnerability is discovered in the TEE OS that compromises the security properties of the TEE. This vulnerability undermines the integrity of any device running version 1 of the TEE. (3) The device manufacturer fixes the vulnerability in the TEE OS and deploys the new image to affected devices using a secure update process. The update process appropriately revokes the authorization of version 1 of the TEE so that the old OS will not reboot. The new OS image includes a certificate in the image identifying the software as version 2 of the TEE. (4) This device may participate in an attestation, and the certificate in the TEE OS will provide accurate evidence of the software in the TEE. An appraiser might be satisfied with this evidence and grant the device access to a protected resource.

It is possible that some devices deployed with version 1 of the TEE are not updated. This scenario can occur even if the device manufacturer is well intentioned and employs an update process for securing devices in the market. These devices may be disconnected from the network or experience a communication or hardware malfunction. This scenario could also be achieved by corrupting the RTU, although it is unlikely because the RTU is stored in on-chip ROM.

Figure 9 depicts the state of a deployed mobile device in which a vulnerability is discovered and the device is

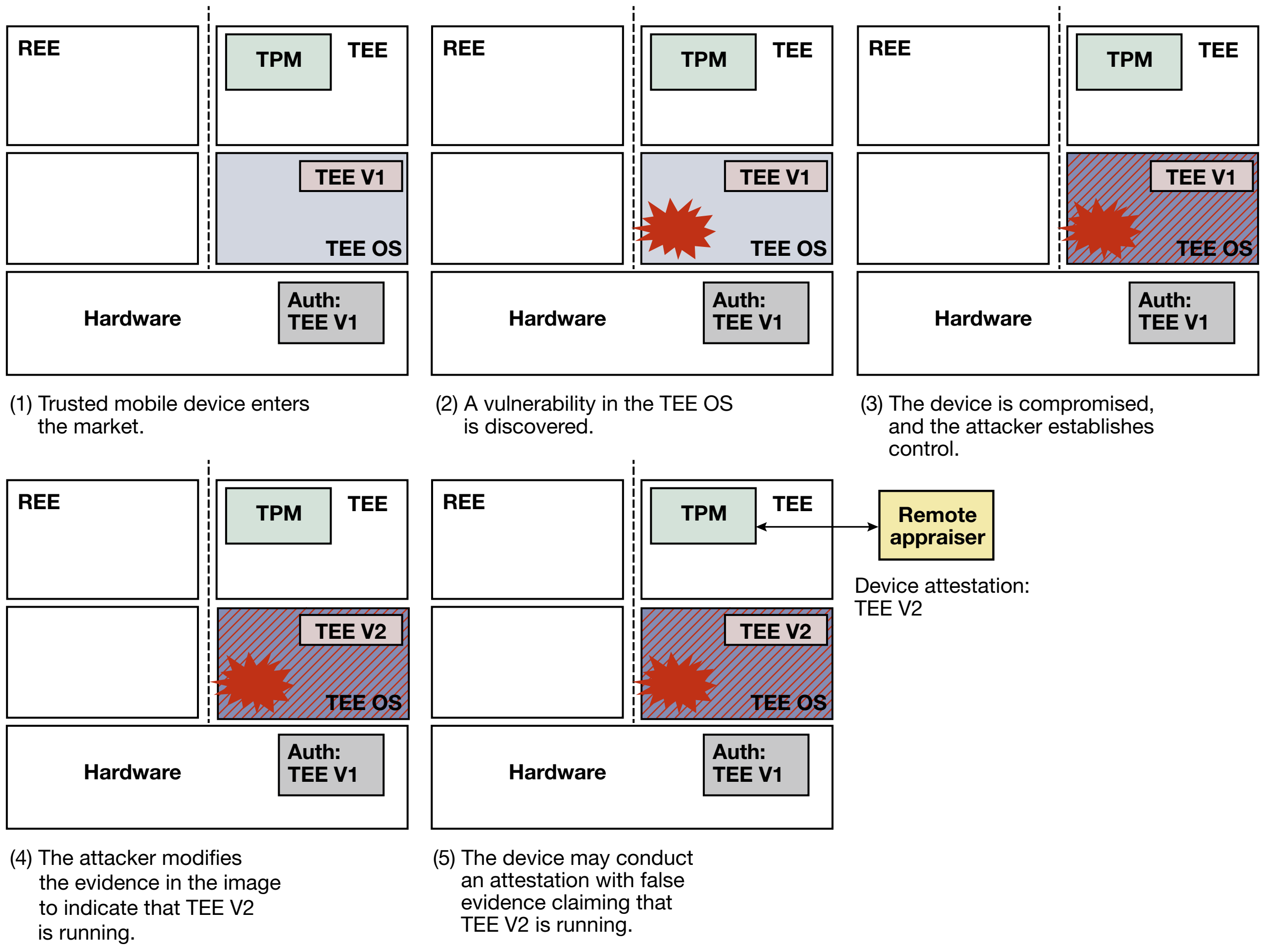
REE
TPM
TEE
TEE V1
TEE OS
Hardware
Auth: TEE V1
(1) Trusted mobile device enters the market.
(2) A vulnerability in the TEE OS is discovered.
(3) The device is compromised, and the attacker establishes control.
TEE V2
(4) The attacker modifies the evidence in the image to indicate that TEE V2 is running.
Remote appraiser
Device attestation: TEE V2
(5) The device may conduct an attestation with false evidence claiming that TEE V2 is running.

Figure 9. The state of a deployed mobile device in which a vulnerability is discovered and the device is not updated.

not updated. In this scenario, events (1) and (2) are the same as those shown in the previous scenario. However, this device is not updated. The vulnerable TEE OS will continue to pass the Secure Boot mechanism and execute although the update has not occurred. (3) At some point, the device may be compromised through the TEE OS vulnerability. Having compromised the TEE software, the attacker has significant control of the device. (4) To conceal its presence on the device, the attacker may modify the certificate in the image to indicate that version 2 of the TEE is running. (5) This device may participate in an attestation, spoofing version 2 of the TEE with false evidence. An appraiser might be satisfied with this evidence and grant the attacker access to a protected resource. This attack undermines the integrity of the device and is not detectable by a remote entity.

TPM MOBILE ROADMAP

APL has worked with the TCG community to establish requirements for the development of trusted mobile devices with strong security capabilities. A collaboration between the TCG and Global Platform Alliance aims to accelerate adoption of the TPM Mobile by device manufacturers. Although this article identifies some potential challenges, the work of the TCG on TPM Mobile can lead to great strides in enhancing trust in mobile devices. The open issues identified—as well as other advanced features such as run-time integrity checking—are potential topics that may be explored in the future.

Beyond identification of architectures that can provide these strong security capabilities, the TCG has in the past also provided certification programs for implementers of those specifications. APL plans on working on such certification programs as well. The challenge is finding a certification strategy that is compatible with the life cycles of mobile devices. Such a program must be designed to be flexible so that manufacturers may select the appropriate security evaluation methodology such as Common Criteria or FIPS 140-2. Completion of TPM Mobile specifications and certification program definitions would represent a significant step toward a new generation of trusted mobile devices.

ACKNOWLEDGMENTS: I thank members of the TCG Mobile Platform Working Group for their significant contributions to the technical content of the specifications article as well as their comments and feedback on this writing. I also thank members of the U.S. government for their contributions to the definitions of the mobile roots of trust.

REFERENCES

[1]Azurati Limited, SharePoint2Go, "Secure Cross-Platform Mobile SharePoint for Enterprise Users," http://www.azurati.com/products/sharepoint2go#screenshots (accessed 14 Mar 2013).

[2]Chen, L., Franklin, J., and Regenscheid, A., *Guidelines on Hardware-Rooted Security in Mobile Devices* (DRAFT), Special Publication 800-164, NIST, Gaithersburg, MD (Oct 2012).

[3]Trusted Computing Group (TCG), "Trusted Platform Module Library Specification, Family "2.0" Level 00 Revision 00.96," http://www.trustedcomputinggroup.org/resources/tpm_library_specification (accessed 14 Jun 2013).

[4]Trusted Computing Group (TCG), Homepage, http://www.trustedcomputinggroup.org (accessed 1 Feb 2013).

[5]ARM, Inc., TrustZone, http://www.arm.com/products/processors/technologies/trustzone.php (accessed 1 Feb 2013).

[6]Trusted Computing Group (TCG), "TPM 2.0 Mobile Trusted Module Use Cases," http://www.trustedcomputinggroup.org/resources/mobile_trusted_module_20_use_cases (accessed 1 Feb 2013).

[7]GlobalPlatform, "GlobalPlatform Device Technology System Architecture Version 1.0," http://www.globalplatform.org/specificationsdevice.asp (accessed 1 Feb 2013).

The Author

Kathleen N. McGill is a member of the Senior Professional Staff in APL's Asymmetric Operations Department. She represents APL in the TCG's Mobile Platform Working Group and acts as a liaison between the greater trusted mobile computing community and researchers at APL. Her work at APL focuses on dynamic integrity measurement of desktop, server, and mobile devices to ensure the integrity of target software. Her e-mail address is kathleen.mcgill@jhuapl.edu.