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 net-

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

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

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.

Transitive Trust

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.

- **Secure Boot:** Secure Boot begins with the execution of immutable code from a fixed locm.fm

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.

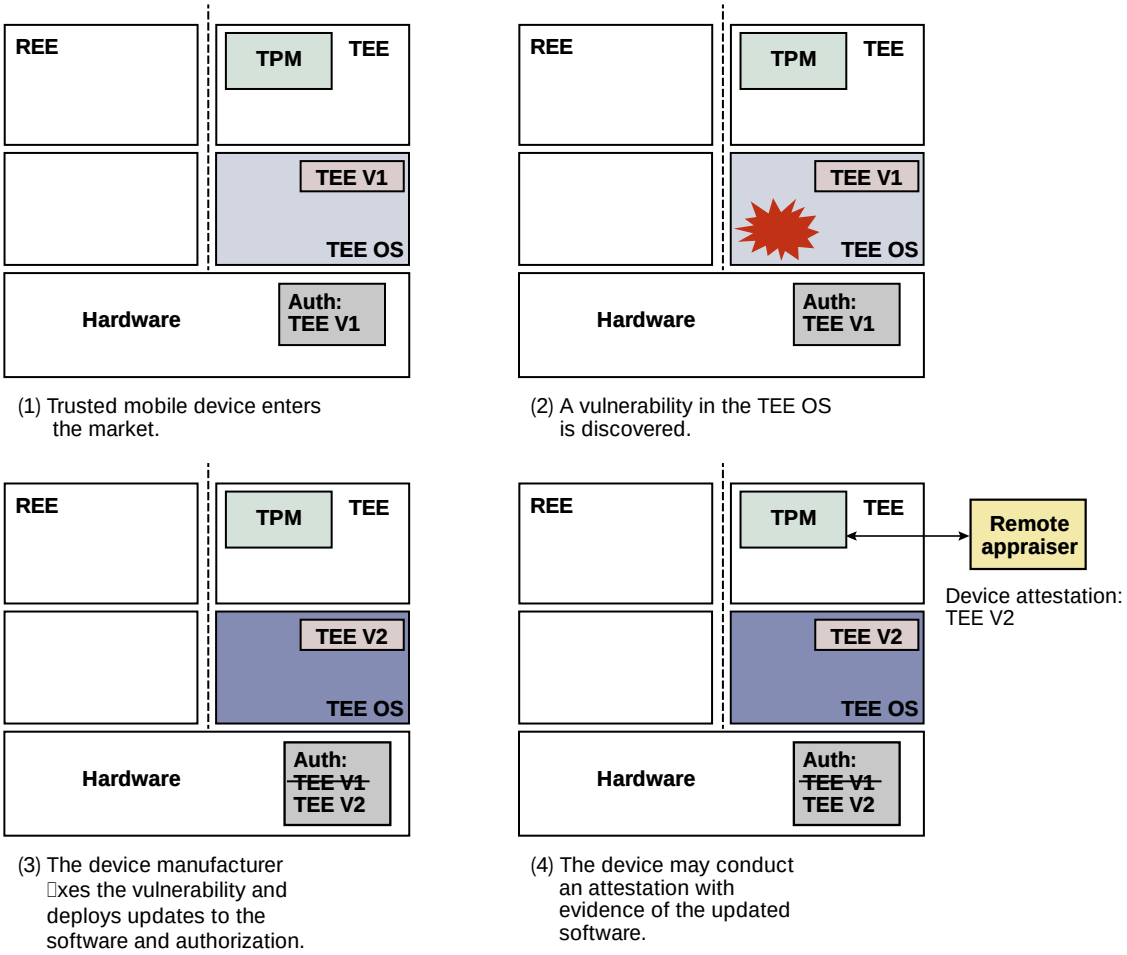
A TPM MOBILE EXAMPLE IMPLEMENTATION

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.

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



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