National Institute of
Standards and Technology
U.S. Department of Commerce

Special Publication 800-164
(Draft)

Guidelines on Hardware-Rooted Security in Mobile Devices (Draft)

Recommendations of the National Institute of Standards and Technology

Lily Chen
Joshua Franklin
Andrew Regenscheid

NIST Special Publication 800-164 (Draft)

Guidelines on Hardware-Rooted Security in Mobile Devices (Draft)

Recommendations of the National Institute of Standards and Technology

Lily Chen
Joshua Franklin
Andrew Regenscheid
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD

C O M P U T E R S E C U R I T Y

October 2012

U.S. Department of Commerce

Rebecca M. Blank, Acting Secretary

National Institute of Standards and Technology

Patrick D. Gallagher, Under Secretary of Commerce for Standards and Technology and Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Authority

This publication has been developed by NIST to further its statutory responsibilities under the Federal Information Security Management Act (FISMA), Public Law (P.L.) 107-347. NIST is responsible for developing information security standards and guidelines, including minimum requirements for Federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate Federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), *Securing Agency Information Systems*, as analyzed in Circular A-130, Appendix IV: *Analysis of Key Sections*. Supplemental information is provided in Circular A-130, Appendix III, *Security of Federal Automated Information Resources*.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-164 (Draft)
Natl. Inst. Stand. Technol. Spec. Publ. 800-164, 33 pages (October 2012)
CODEN : NSPUE2

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by Federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, Federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST publications, other than the ones noted above, are available at http://csrc.nist.gov/publications.

Public comment period: October 26 through December 14, 2012

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930), Gaithersburg, MD 20899-8930
Email: 800-164comments@nist.gov

Acknowledgments

The authors wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content.

Abstract

Many mobile devices are not capable of providing strong security assurances to end users and organizations. Current mobile devices lack the hardware-based roots of trust that are increasingly built into laptops and other types of hosts. This document focuses on defining the fundamental security primitives and capabilities needed to enable more secure mobile device use. This document is intended to accelerate industry efforts to implement these primitives and capabilities. The guidelines in this document are intended to provide a baseline of security technologies that can be implemented across a wide range of mobile devices to help secure organization-issued mobile devices as well as devices brought into an organization, such as personally-owned devices used in enterprise environments (e.g., Bring Your Own Device, BYOD).

Keywords

information security; mobile device security; root of trust; smartphone; tablet

Trademark Information

Table of Contents

List of Appendices

List of Figures

Executive Summary

As mobile technology matures, employees increasingly want to use both organization-issued and personally-owned mobile devices to access corporate enterprise services, data and resources to perform work-related activities. Enterprises are under pressure to accept the associated security risks inherent in today's mobile devices because of several factors including anticipated cost savings and employee desire for greater convenience. Unfortunately, many mobile devices are not capable of providing strong security assurances to end users and organizations; these devices lack the hardware-based roots of trust that are increasingly built into laptops and other types of hosts. Mobile devices are also vulnerable to "jailbreaking" and "rooting," which provide device owners with greater flexibility and control over the devices, but also bypass important security features which may introduce new vulnerabilities.

Organizations may wish to verify the integrity of a mobile device before granting it access to the organization's information. This verification provides some level of assurance that the organization's information will be properly protected—for example, the device's security adheres to policy, the device was not modified, the device is authorized to access the organization's information, and locally stored information from the organization has its confidentiality and integrity protected. This integrity verification is considered important for both organization-issued and personally-owned mobile devices.

The guidelines in this document are intended to introduce a baseline of security technologies that can be implemented across a wide range of mobile devices to help secure both organization-issued mobile and personally-owned mobile deployment scenarios. This document focuses on defining the fundamental security primitives and capabilities needed to securely enable mobile devices. This document is intended to accelerate industry efforts to implement those primitives and capabilities.

Several roles are defined as the basis for these recommendations. For example, a Device Owner is an entity that has purchased and maintains ownership of a mobile device. An Information Owner is an entity whose information is stored and/or processed on a device. An Information Owner can be an application-specific provider, a digital product provider, or an enterprise that allows access to resources from mobile devices. In this publication, it is assumed that each mobile device has a single Device Owner and one or more Information Owners.

Major recommendations contained in this document include the following:

Mobile device are required to implement the following security components.

Security components are foundational elements that can be leveraged by the device, the operating system (OS), and applications. The three required security components are Roots of Trust (RoTs), an application programming interface (API) to expose the RoTs to the platform, and a Policy Enforcement Engine (PEnE).

- **Roots of Trust (RoTs).** RoTs are security primitives composed of hardware, firmware and/or software that provide a set of trusted, security-critical functions. They must always behave in an expected manner because their misbehavior cannot be detected. As such, RoTs need to be secured by their design. Hardware RoTs are preferred over software RoTs due to their immutability, smaller attack surface, and more reliable behavior. To support device integrity, isolation, and protected storage, devices should implement the following RoTs:

    - Root of Trust for Storage (RTS)- provides a protected repository and a protected interface to store and manage keying material

  - Root of Trust for Verification (RTV)- provides a protected engine and interface to verify digital signatures associated with software/firmware and create assertions based on the results
  - Root of Trust for Integrity (RTI)- provides protected storage, integrity protection, and a protected interface to store and manage assertions
  - Root of Trust for Reporting (RTR)- provides a protected environment and interface to manage identities and sign assertions
  - Root of Trust for Measurement (RTM)- provides measurement used by assertions protected via the RTI and attested to with the RTR

- **An application programming interface (API) to expose the RoTs to the platform.** RoTs must be exposed to the device and OS in order to establish a chain of trust for user applications. Mobile applications interacting with services offered by various Information Owners will frequently utilize the security functions provided by the RoTs to locally store cryptographic keys, authentication credentials, and other sensitive data. Moreover, Information Owners will frequently rely on assertions from the device.

- **A Policy Enforcement Engine (PEnE).** The PEnE enables the processing, maintenance, and management of policies on the mobile device. The PEnE allows Information Owners to express the control they require over their information; it translates the desired requirements for storing and sharing their information into the appropriate device and network configurations and policy. The PEnE must be trusted to implement the Information Owner's requirements correctly and to prevent one Information Owner's requirements from adversely affecting another's. In order to perform key functions, the PEnE must be able to query the device's configuration state. While the PEnE is necessarily a component on the mobile device, it may interact with off-device components to assist in the policy generation and compliance decisions.

Mobile devices should implement the following three mobile security capabilities:

- **Device Integrity:** Device integrity is the absence of corruption in the hardware, firmware and software of a device. A mobile device can provide evidence that it has maintained device integrity if the state of the device can be shown to be in a state that is trusted by a relying party. A device has integrity if its software, firmware, and hardware configurations are in a state that is trusted by a relying party. The mechanism for communicating this trusted state is through one or more assertions that the Device Owner allows a device to make to the Information Owner.

- **Isolation:** Isolation prevents unintended interaction between Information Owners on the same device.

- **Protected Storage:** Protected storage preserves the confidentiality and integrity of sensitive data on the device while at rest, while in use (in the event an unauthorized application attempts to access an item in protected storage), and upon revocation of access.

1. Introduction

1.1 Purpose and Scope

This document is intended to accelerate industry efforts to implement security capabilities that can provide a higher degree of assurance of the trustworthiness of a device. Specifically, this document focuses on defining the fundamental security primitives and capabilities needed to securely enable mobile devices. This document defines essential security components and recommended security capabilities. The guidelines in this document should not limit implementation options or be considered the maximum capability or technology that can be used to address mobile security needs.

The guidelines in this document are intended to provide a baseline of security technologies that can be implemented across a wide range of mobile devices to help secure organization-issued mobile devices as well as outside devices brought into an organization, such as personally-owned devices used in enterprise environments (e.g., Bring Your Own Device, BYOD). The guidelines in this document are intended to be appropriate and reasonable for both enterprise-class devices as well as consumer-class devices. These guidelines may facilitate a convergence of enterprise and consumer grade devices, creating new devices that are acceptable for both consumers demanding state-of-the-art technologies and enterprises demanding critical security features.

1.2 Audience

The intended audience for this document includes mobile Operating System (OS) vendors, device manufacturers, security software vendors, carriers, application software developers and information system security professionals who are responsible for managing the mobile devices in an enterprise environment. The material may also be of use when developing enterprise-wide procurement and deployment strategies for mobile devices.

The material in this document is technically oriented, and it is assumed that readers have at least a basic understanding of system and network security. The document provides background information to help such readers understand the topics that are discussed. Readers are encouraged to take advantage of other resources (including those listed in this document) for more detailed information.

1.3 Document Structure

The remainder of this document is divided into the following sections and appendices:

- Section 2 discusses the need for hardware-rooted security in mobile devices.
- Section 3 defines essential security components needed for hardware-rooted security in mobile devices.
- Section 4 describes key mobile security capabilities that address the identified problems with mobile device security.
- Appendix A defines selected acronyms and abbreviations used in the document.
- Appendix B identifies references used in this document.

1.4 Document Conventions

Throughout this document Key Words are used to identify requirements. The Key Words “SHALL”, “SHALL NOT”, and “SHOULD” are used. These words are a subset of the IETF Request For Comments (RFC) 2119 key words, and have been chosen based on convention in other normative documents [RFC2119]. In addition to the Key Words, the words ‘need,’ ‘can,’ and ‘may’ are used in this document, but are not intended to be normative.

2. Background

As mobile technology matures, employees increasingly want to use both organization-issued and personally-owned mobile devices to access corporate enterprise services, data and resources to carry out work-related activities. Enterprises are under pressure to accept the associated security risks inherent in today's mobile devices due to, among other factors, perceived cost savings and employee desire for greater convenience.

Many mobile devices, particularly those that are personally-owned, are not capable of providing strong security assurances to end users and organizations. Current mobile devices lack the hardware-based root of trust features that are increasingly built into laptops and other types of hosts (e.g., Trusted Platform Modules, TPMs). Mobile devices are also vulnerable to "jailbreaking" and "rooting," which provide device owners with greater flexibility and control over the devices, but also bypass important security features which may introduce new vulnerabilities. See NIST SP 800-124 Revision 1, *Guidelines for Managing and Securing Mobile Devices in the Enterprise* [SP800-124], for more information on current threats and vulnerabilities involving mobile devices.

Organizations may wish to verify the integrity of a mobile device before granting it access to the organization's information. This verification provides assurance that the organization's information is properly protected—for example, the device's security is not breached, the device is authorized to access the organization's information, and locally stored information from the organization will have its confidentiality and integrity protected. This integrity verification is considered important for both organization-issued and personally-owned mobile devices; however, it is generally easier to achieve for organization-issued devices than personally-owned devices. Current mobile device platforms provide limited capabilities to perform such integrity verification. The guidelines in this document are intended to support stronger integrity verification features in devices that implement these guidelines.

This section provides a foundation for the rest of the document. Section 2.1 defines roles that are leveraged throughout the document. Section 2.2 discusses two scenarios of organization-issued devices and BYOD, and discusses how a role's perspective is affected by the scenario its enterprise chose to employ.

2.1 Roles

The roles and scenarios presented in this section provide the basis for the recommended security components and capabilities for mobile devices defined in Sections 3 and 4. This document defines seven roles: Device Owner, Information Owner, enterprise, device manufacturer, mobile operating system (OS) vendor, mobile carrier, and user.

2.1.1 Device Owner & Information Owner

A Device Owner is an entity that has purchased and maintains ownership of a mobile device. An Information Owner is an entity whose information is stored and/or processed on a device. An Information Owner can be an application-specific provider, a digital product provider, or an enterprise that allows access to resources from mobile devices, for example.

Every mobile device has a single Device Owner and one or more Information Owners. For an organization-issued mobile device, the organization is the Device Owner and the organization is also an Information Owner. For a personally-owned mobile device in a BYOD scenario, the user who owns the device is the Device Owner and one of the Information Owners (for the user's personal information), and the organization is another Information Owner (for the organization's information). While any device has

a single Device Owner, the device can have multiple users. Figure 1 shows some of the issues inherent in a BYOD scenario.

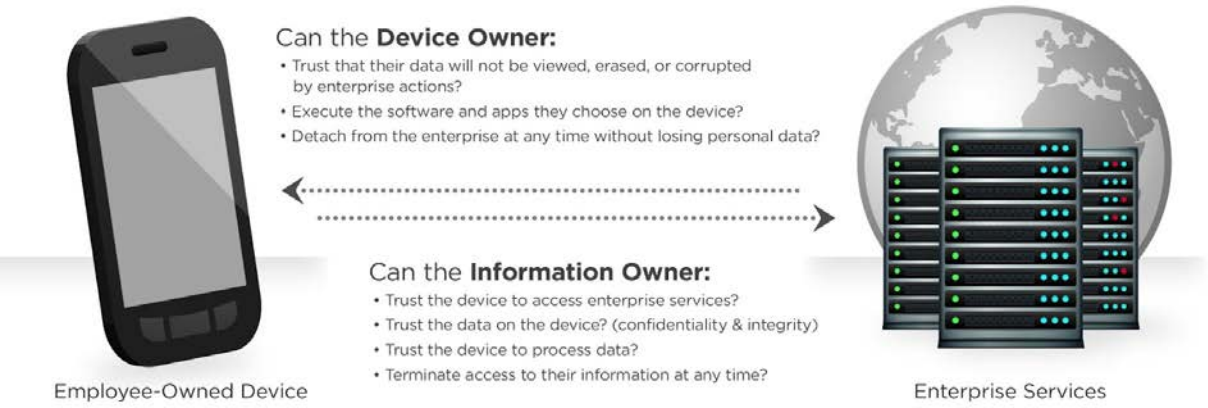

Figure 1: Mobile Scenario Owners and Motivations

Note that all of the material can also apply to organization-issued mobile devices simply by setting the Device Owner and the Information Owner to both be the organization.

2.1.2 Enterprises

An enterprise is an organization with users, computing resources and data, and is the primary type of Information Owner considered in this publication. Enterprises are organizations with specific missions to accomplish (business cases or services to citizens) where the confidentiality and integrity of their information is essential to accomplishing their mission. Enterprise data must be shared between computing resources and users in a confidential manner while maintaining integrity in order to support the mission. Enterprises need assurances that mobile devices accessing their information are trustworthy and meet their confidentiality, integrity and availability requirements. The capabilities offered to consumers in today's mobile devices cannot efficiently help the enterprise maintain device integrity. By leveraging the security components and capabilities presented here, enterprises are able to benefit from mobile devices in the enterprise while ensuring the necessary mechanisms are in place to secure their data and computing resources. Enterprises expect Device Owners to acquire devices through a trustworthy supply chain, maintain positive control over their devices and, as authorized users, are trusted to not intentionally subvert security protections on the device.

2.1.3 Mobile Carriers, Device Manufacturers and OS Vendors

Mobile carriers, device manufacturers and mobile OS vendors together provide the mobile platform used by organizations and personal users. These entities have a vested interest in enabling more secure consumer scenarios. In doing so they simultaneously satisfy the security requirements demanded by enterprises, and capitalize on the growing revenue potential from an increasingly consumer-driven market.

2.1.4 Users

A user is the person consuming or generating data on a mobile device. A typical user could be an enterprise employee with a smartphone or tablet. Users consume and generate data that is both personal in nature and related to the enterprise, and may simultaneously require access to both. The information users

consume may be malicious in nature or originate from untrusted or unreliable sources. Enterprises want to ensure that, to the best extent possible, their information is not put at risk by the actions of users. Many enterprises today find it beneficial to allow users 24/7 access to certain information such as email, contacts, and calendar. In the future this may broaden to information currently deemed unsuitable for mobile devices. The requirements of this document support the desire to use a single mobile device to view information from multiple information owners. This is accomplished through enterprises presenting policies to users, who then choose to accept or reject how enterprises wish to use the mobile device. By accepting the policy, users can consume and generate any information they wish on devices meeting the requirements of this document.

2.2 Scenarios

The following two scenarios provide context for the security components and capabilities discussed in Sections 3 and 4. The components and capabilities in this document can be applied to either scenario.

2.2.1 Organization-Issued Devices

In the traditional model, enterprises provide users with mobile devices and are the Device Owner and Information Owner. Outside the enterprise, organization-issued devices are used to access enterprise data and restrictions are placed on the device's capabilities to protect enterprise data and resources. The devices are centrally managed, sometimes using enterprise-owned software known as a Mobile Device Manager (MDM). A device policy is strictly enforced and one user is assigned to one or more mobile devices.

From the user's perspective, in addition to their personally-owned mobile device, the user is now responsible for another device. However, enterprise policy typically restricts personal use of organization-issued devices, reducing their utility to the users. The enterprise is the Information Owner for an overwhelming majority of the device's information.

2.2.2 Non-Organization Issued Devices (BYOD)

The concept of allowing a user's mobile device to store and interact with an enterprise's data is known as BYOD. A user (the Device Owner) purchases a mobile device and wants to access his or her organization's enterprise data using that device. The enterprise (an Information Owner) will want to restrict access to their data based on the capabilities and configuration of the user's personal device. The Device Owner expects to be able to access the device for personal use without limitations on device capability. This separation of personal data and enterprise data on a single mobile device is illustrated in Figure 2.

Ideally, a Device Owner should be able to consume arbitrary, non-enterprise data (e.g., Internet services) in one instance of a browser while simultaneously accessing protected data in another. Information owners will predicate access to data based on terms of a usage agreement expressed as a policy. For instance, Device Owners may determine that installing a company's line of business applications on their mobile device achieves a better home-work balance. Device Owners could log into their company's extranet site, select the applications of interest, and install them on their mobile device. As a prerequisite, an Information Owner's policy may require device encryption and a device PIN.

At the foundation of this transaction is a hardware root of trust (RoT) that is used to establish a chain of trust for the device and provide assertions about the device to the Information Owner. From the Device Owner's perspective, their company was authorized to learn basic information about the device and its configuration. As a result, the company is able to gain reasonable confidence in the security of their data

due to the inherent device integrity, isolation, and data-at-rest protection mechanisms supported by the device.

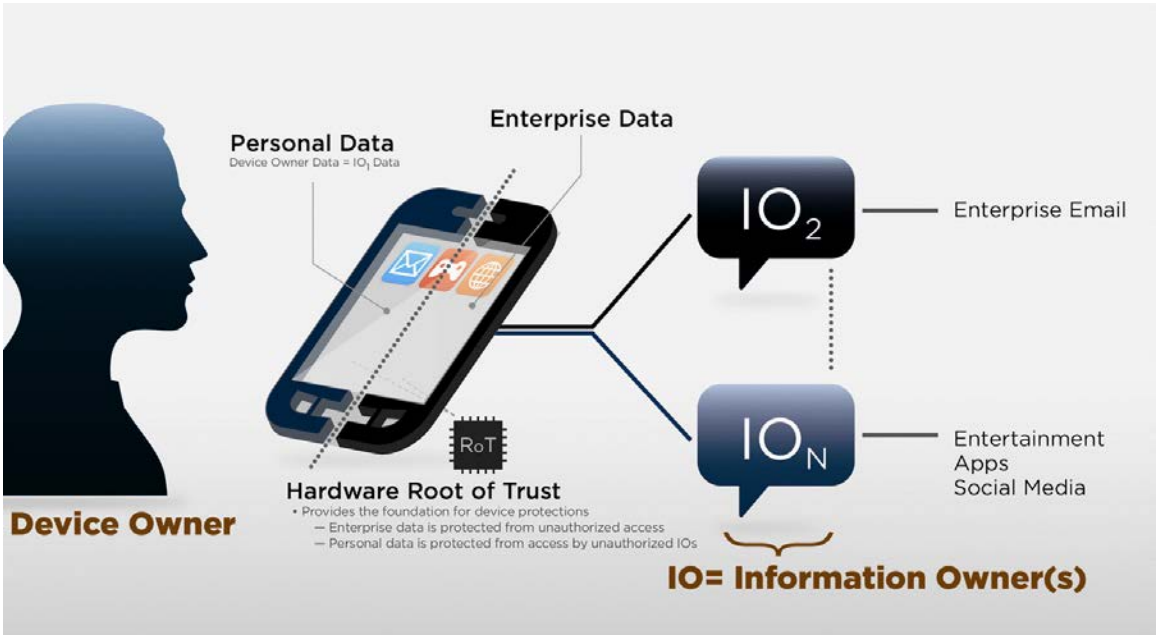

Figure 2: BYOD Scenario

Device Owners control all decisions concerning the use of their devices and can change device settings at any time, possibly resulting in the inability to access company data if new settings conflict with company policy. If the personally-owned device is lost, the Device Owner and the Information Owner both expect that the organization's data will be protected from disclosure. Note that the mechanisms that protect one data domain may differ from those that protect other domains. For example, the device may contain enterprise data that is locked, and also personal data of the Device Owner that remains unlocked at the Device Owner's discretion. An Information Owner retains the ability to control access (e.g., revoke access, wipe) to their data at all times, but does not have the capability to touch data it does not own (e.g., the Device Owner's personal information).

Information Owners expect that Device Owners acquired the device through a trustworthy supply chain and authorized users are trusted to not intentionally subvert security features on the device. An Information Owner also expects that if multiple users (i.e., independent Information Owners) share a device, the device will enforce access control over data such that only users approved to access an Information Owner's data will be able to access that data.

3. Mobile Devices Components and Architecture

3.1 Security Components

In order to provide the security capabilities needed to enable the scenarios described in Section 2, every mobile device requires a set of *security components*—foundational security elements that can be leveraged by the device, OS, and applications. This section provides a concise overview of the three required security components: Roots of Trust (RoTs), an application programming interface (API) to expose the RoTs to the platform, and a Policy Enforcement Engine (PEnE). Section 4 of this document describes how these primitives will be leveraged by three new security capabilities: device integrity, isolation, and protected storage.

3.1.1 Roots of Trust (RoTs)

Roots of Trust (RoTs) are the foundation of assurance of the trustworthiness of a mobile device. As such, RoTs are security primitives composed of hardware, firmware and/or software that provide a set of trusted, security-critical functions. They must always behave in an expected manner because their misbehavior cannot be detected.

Hardware RoTs are preferred over software RoTs due to their immutability, smaller attack surfaces, and more reliable behavior. They can provide a higher degree of assurance that they can be relied upon to perform their trusted function or functions. Software RoTs could provide the benefit of quick deployment to different platforms. To support device integrity, isolation, and protected storage, devices should implement the following RoTs:

- **Root of Trust for Storage (RTS)—**provides a protected repository and a protected interface to store and manage cryptographic keys and other critical security parameters. It may also include or process policy details for the use of the keys or critical security parameters. For higher assurance of the protection of sensitive data, it is preferred that the RTS be implemented in trusted hardware and protected software with a protected interface for accessing or managing the data protected by the RTS. The RTS will also typically contain limited cryptographic capabilities to use keys protected by the RTS without releasing them outside the RTS's logical boundary in plaintext.

- **Root of Trust for Verification (RTV)—**provides a protected interface and engine to verify digital signatures associated with software/firmware and create assertions based on the result. It executes the signature verification algorithm and has access to a key store that includes the public key needed to verify a signature. This key store may be stored internal to the RTV or it may rely on an RTS to protect and maintain the key store.

- **Root of Trust for Integrity (RTI)—**provides protected storage, integrity protection, and a protected interface to store and manage assertions. These capabilities are usually provided through the use and maintenance of tamper evident locations for the purpose of securely storing measurements and assertions. It also provides a limited protected interface for accessing and modifying these storage locations. For example, this interface may allow extensions of these storage locations through a hash chaining method, but disallow direct writes (such as the TPM method for extending Platform Configuration Registers, PCRs). Together the tamper evident locations and protected interface collectively form the Root of Trust for Integrity.

- **Root of Trust for Reporting (RTR)—**provides a protected environment and interface to manage identities and sign assertions for the purposes of generating device integrity reports. It has the capability to reliably cryptographically bind an entity to the information it provides. In the case of

device integrity reports, the RTR serves as the basis for the capabilities of integrity and non-repudiation of these reports. It necessarily leverages capabilities provided by other RoT in the system, including the RTM and RTI.

- **Root of Trust for Measurement (RTM)—**provides trusted measurement functions to be used by assertions that are protected via the RTI and attested to with the RTR. It has the capability to make inherently reliable integrity measurements and is the root of the chain of transitive trust for subsequent measurement agents. A small root of trust for measurement applied very soon after a re-initialization of an endpoint may have greater value than a root of trust for measurement instantiated later, mainly in minimizing the attack surface's exposure to subversion of the measurement process. The later the endpoint invokes the RTM, the more opportunity an adversary has to subvert the measurement trust chain.

The RoTs defined above are logical entities. A particular RoT may be implemented in several hardware and software components, but it is important that these components are protected as a group. Also, two or more RoTs may share components, effectively being implemented as a group of trusted capabilities. The key point is that mobile devices need to implement the security capabilities and services defined above in trustworthy hardware, firmware and software components.

3.1.2 Application Programming Interface (API) to RoTs

The capabilities provided by the RoTs need to be exposed to the platform so that mobile operating systems and applications can benefit from the stronger security assurances they typically provide over features implemented at the OS or application level. We expect mobile operating systems to utilize the capabilities provided by the RoTs to create and protect device integrity reports, verify and measure firmware and software, and protected locally stored cryptographic keys, authentication credentials, and other sensitive data.

In particular, the RoTs need to be exposed by the operating system to applications through an open application programming interface (API). This will provide application developers a set of security services and capabilities they can use to secure their applications and protect the data they process. By providing an abstracted layer of security services and capabilities, these APIs can reduce the burden on application developers to implement low-level security features, and instead allow them to reuse trusted components provided in the RoTs and the OS. To further reduce the burden on application developers, and to encourage adoption of these features, the APIs should be standardized within a given mobile platform and, to the extent possible, across platforms. This will make it easier for developers to write applications that make use of these security capabilities across a broad range of devices.

Applications can use the APIs, and the associated RoTs, to request device integrity reports, protect data through encryption services provided by the RTS, and store and retrieve authentication credentials and other sensitive data.

3.1.3 Policy Enforcement Engine (PEnE)

There needs to be secure mechanisms for the Device Owner to manage the different policy agreements that will be required to establish access with multiple Information Owners. The Policy Enforcement Engine (PEnE) enforces policies on the device with the help of other device components and enables the processing, maintenance, and management of policies on both the device and in the Information Owners' environments. The PEnE provides Information Owners with the ability to express the control they require

over their information. The PEnE interacts with all of the Information Owners and translates the desired requirements for storing and sharing their information into the appropriate device and network configurations and policies.

The PEnE needs to be trusted to implement the Information Owner's requirements correctly and to prevent one Information Owner's requirements from adversely affecting another's. The PEnE will enforce the policy with the strictest requirements. Where there is an unresolvable conflict, the PEnE will notify the Device Owner and enforce a default policy that prevents unauthorized access to data until the conflict is resolved.

In order to perform key functions, the PEnE needs to be able to query the device's configuration and state. For example, if device sensor events are part of a policy to be enforced, the PEnE needs access to those outputs in order to enforce the policy, independent of the Application Context. To have the necessary privileges to access the device state and enforce policies, we expect that the PEnE will generally be part of the mobile operating system.

3.2 Notional Architecture

Like other computers, mobile device architectures are composed of a stack of hardware, firmware, and software. Figure 3 describes the notional architecture for mobile devices used throughout this section. Generally, each level of the stack provides services and interfaces for the higher levels of the stack to use. Higher levels of the stack typically must trust lower levels of the stack to be trustworthy, as they often have limited insight into what occurs at those levels.

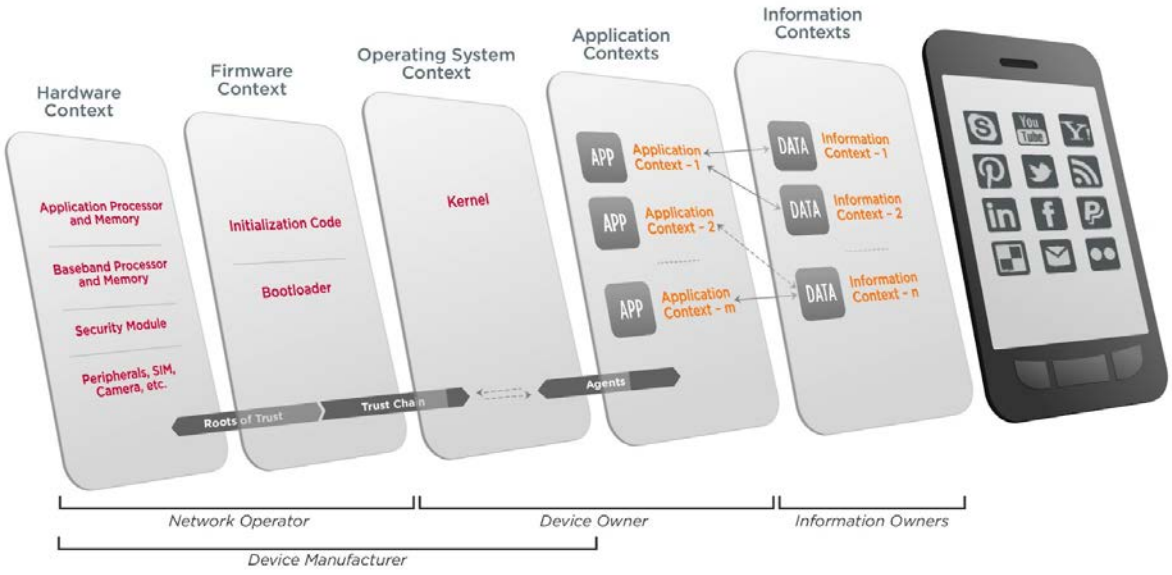

Figure 3: Notional Mobile Device Architecture

Hardware Context: The lowest level of the stack includes hardware components, which this document collectively refers to as components within the Hardware Context. Mobile devices are inherently portable, battery powered computing devices with one or more radios for network connectivity. It is increasingly common for smartphones to support multiple data connectivity methods including 3GPP and 3GPP2 cellular, IEEE 802.11 wireless local area networks (WLANs), Bluetooth, and near field communication (NFC). Most smartphones are equipped with rich media player capabilities, cameras, GPS receivers, and

various sensors (accelerometer, gyroscope, proximity, compass). Components within the Hardware Context are provided by the chipset and device manufacturer. The security components that operate at this level are critical, as they serve as the basis from which to build and extend trust to higher levels of the stack. These trusted components are RoTs. Examples of Hardware RoTs include a a separate chip that acts as some of the RoTs to store and sign assertions, as well as a key used to root confidentiality guarantees made by other RoTs. Hardware differs from firmware by being inherently immutable.

Firmware Context: The next level in the stack is the Firmware Context. Firmware is a protected set of special code that interacts closely with mobile device hardware. It includes boot firmware that initializes the hardware components and loads the OS, as well as other firmware components that control other hardware components, such as the baseband processor and radios. The Firmware Context may contain some of the RoTs, but it will typically take advantage of hardware capabilities in providing protection to those RoTs (e.g., flash write protection). Firmware typically is written by the device manufacturer, possibly with the help of the wireless provider. It also typically has controlled mutability, with any changes being produced by the manufacturer and verified and authorized by the provider before they are deployed. This firmware is also responsible for setting up the configuration of the hardware of the device and measuring and launching itself, any other ROM code that runs before the OS, and the boot record which in turn launches the main OS. Together this protects against pre-boot malware and bootkits that can cloak other malware on the device.

Operating System Context: The Operating System Context includes the OS kernel, PEnE, and system service components along with their configuration data. The OS is under the control of the Device Owner, although software updates to this context are typically provided and approved by the manufacturer in cooperation with the service provider. The components of the Operating System Context set up application environments, provide interfaces to access the device hardware, manage and enforce information policies, and offer various other services. The PEnE, whose functionality may be spread over multiple OS components, manages information policies and coordinates with the appropriate components in the architecture for their enforcement. The OS kernel typically provides application isolation, but protection requirements enforced through the use of either device integrity or protected storage may require the cooperation of hardware and firmware components as well.

Application Context: Applications execute in the Application Context. The OS kernel typically provides isolation capabilities between applications. The isolation mechanisms used to enforce this separation are intended to prevent applications from modifying or tampering with code, configuration data, or user data tied to other applications. Application isolation capabilities could allow devices to run untrusted applications along with trusted applications by providing assurances that the untrusted software should not be able to compromise other information processed, stored or transmitted by trusted applications.

Information Context: Applications running on top of the OS receive, store, and process data. Security capabilities provided by applications, OS components, and software/firmware components rooted in hardware protect this data, maintain separation, and ensure that only authorized applications have access to each piece of data. These protected data storage elements are information components contained within their own Information Context. Each Information Context is under the control of an Information Owner. Information Owners may define certain policies and security requirements that protect their information as it is stored and processed on the device. These policies are managed by the PEnE and enforced by the appropriate components in the architecture. An Information Owner may employ their own agents to assist in monitoring their Information Context and enforcing their policy. These agents would work with the PEnE and would not have any additional privileges. A more privileged agent used by multiple Information Owners could be installed with the approval of the Device Owner. This type of agent would be considered to be a part of the PEnE. In no case can one Information Owner gain access to another Information Context without the permission of the owner.

The trusted software components used to enforce information policies are in a transitive trust chain that is tied back to one or more of the RoTs. There may also be untrusted software components that communicate with the RoTs and are relied on to provide part of a security capability, but the security of the security services provided by the RoTs do not rely upon the correct behavior of these components as long as the RoTs are trustworthy. While Information Owners may not directly modify applications or OS components without cooperation from the Device Owner, Information Owners do need assurance that the device is still operating in a trustworthy state to have confidence that the platform's security capabilities are functioning properly in the context of data access. The device integrity guidelines outlined in this document provide the mechanisms necessary to provide this assurance to Information Owners.

4. Key Capabilities

Mobile devices should implement the following three mobile security capabilities to address the challenges with mobile device security: device integrity, isolation, and protected storage.

- **Device Integrity:** Device integrity is the absence of corruption in the hardware, firmware and software of a device. A mobile device can provide evidence that it has maintained device integrity if its software, firmware, and hardware configurations can be shown to be in a state that is trusted by a relying party. The mechanism for communicating this trusted state is through one or more assertions that the Device Owner allows a device to make to the Information Owner. A device with integrity has the ability to assert specific claims about its configuration, health, or operating status in a way that Information Owners can confidently rely upon to make decisions about interacting with the device and Device Owner.

- **Isolation:** Isolation prevents unintended interaction between applications and Information Contexts on the same device.

- **Protected Storage:** Protected storage preserves the confidentiality and integrity of data on the device while at rest, while in use (in the event an unauthorized application attempts to access an item in protected storage), and upon revocation of access.

4.1 Device Integrity

Device integrity is the absence of corruption in the hardware, firmware and software of a device. A mobile device can provide evidence that it has maintained device integrity if the state of the device can be shown to be in a state that is trusted by a relying party. An assessment of the state of the individual hardware, firmware, and software components in a device determines the integrity (i.e., soundness) of a device. An Information Owner needs be able to control access to its information or services based on integrity information asserted by the device.

This section describes a framework for a device to assess its own integrity, and how that integrity is asserted off the device. It assumes RoTs with an established set of trusted components previously introduced, which are extended into chains of trust and into trusted agents. The framework describes how a device's integrity is verified through verifying a signature or message authentication code generated over the software and the configuration data.

In the BYOD scenario a Device Owner wants to access information or services owned and controlled by another party. In order for the other party, the Information Owner, to grant access to that service or data, a Device Owner allows its device to assert to the Information Owner that the device has the capability to protect that data or service in ways defined as acceptable by the Information Owner. In order for the Information Owner to have confidence that the device being used to access that data or service is capable of providing the protections required, the device needs to be able to make certain reliable assertions about itself to the Information Owner. In some scenarios, Information Owners may want explicit assertions made about specific information components. Devices should possess the ability to make assertions both locally and remotely. Devices with RoTs can credibly report assertions on the integrity of the device.

4.1.1 Threat Addressed through Integrity

Mobile operating systems on devices, along with the underlying firmware and hardware, provide or manage many of the important security mechanisms necessary to secure mobile devices. A major threat to mobile devices is the unauthorized modification of security-critical software and firmware components

in mobile devices that provide these key capabilities. However, Information Owners need assurance that these components are in a trustworthy state in order to have confidence that their data will be protected once it is stored or processed on the device.

The device integrity assertion capabilities described in this section provide Information Owners will the ability to query the state of a mobile device, and make informed trust decisions based on the result. If the device is not in a trusted state, due to accidental or malicious corruption of security-critical components, the Information Owner could choose to not allow that device to access its information. If the device already possesses data from the Information Owner, the Information Owner could choose to invoke a selective wipe for that data, described in Section 4.3.3.3, to reduce their potential exposure to a security breach.

4.1.2 Considerations for Device Integrity

An assertion is a set of one or more attributes that represents the state of an entity in a transaction, usually online, but possibly between two different information domains on the same device. That entity can be a person, an object, a process, or a service.

Mobile devices may use assertions to represent the state of firmware as either verified or unverified, the state of an OS as either validated or not, the state of file encryption as either on or off, the state of the microphone as either on or off, etc. A PEnE collects assertions from a mobile platform and determines whether or not the platform is in compliance with a particular policy.

When a device asserts its state with cryptographic primitives, it will need signing keys. These signing keys represent the tangible identity of some entity, whether it is the device, the Device Owner, the Information Owner, or an application on the device serving as a proxy for one of them. Other characteristics, such as an account name or serial number, may optionally be a part of that identity. These principles serve as a guide to the establishment and use of these identities represented by signing keys.

- A device should establish at least one device identity at a minimum; this does not rule out the use of multiple identities for the device, however, the Device Owner maintains the usage policies for all device identities.
- A device may establish a unique device identity for the purpose of device authentication; however, a unique device identity is not necessary for assertions.
- A device should allow Information Owners to establish Information Owner identities, which are distinct from device identities.
- Information Owners may establish their own identities only if authorized by the Device Owners.
- A device uses either a device identity or an Information Owner identity to assert attributes and characteristics (i.e., the state) of the device.
- Device identities and Information Owner identities may assert only those attributes and characteristics explicitly authorized by its Device Owner.
- A device uses RoTs to provide integrity, authenticity, non-repudiation, and freshness to the assertion reporting process.
- A device uses standard mechanisms for establishing, protecting, and maintaining identities, generating and protecting assertions, and assuring the integrity, authenticity, non-repudiation, and freshness of its assertion reports.
- A Device Owner can monitor all attempts by device and Information Owner identities to assert device attributes and characteristics.

4.1.3 Device Integrity Capabilities

This section describes six facets of the assertion lifecycle: establishing identities, and generating, maintaining, sharing, collecting, and processing assertions.

4.1.3.1 Establishing Identities

Upon provisioning, a mobile device establishes identities for the device and the Device Owner. In order for the Device Owner to control access to assertions of device attributes and characteristics, the Device Owner should authorize the platform to establish identities for each Information Owner. Whether these identities are unique per platform or per Information Owner is left to the discretion of the Information Owner. Upon each application or service provisioning, the device will either use an existing identity for an Information Owner or establish a new one. The device uses policy to control the Information Owner's access to assertions about the device.

4.1.3.2 Generating Assertions

Mobile devices generate assertions during boot by verifying the signatures of the boot components. The mobile device verifies the signature of each component through the RTV. Upon a successful verification, it creates an assertion and extends it into an integrity register protected by an RTI. In this case, the mobile device is acting as an RTM. It is important that standards so that systems that generate and consume assertions can interoperate throughout their lifecycle.

> **Example boot code assertion generation implementation:**
> The device could cryptographically hash the certificate used to verify the signature (or policy assertion associated with the certificate) along with a binary 1: *Success = Hash (Certificate || 1)*. Then it extends the hash into an integrity register: *NewValue = Hash (OldValue ¦¦Success)*. If the verification is not successful, it creates an assertion by hashing the certificate used to verify the signature along with a binary 0: *Failure = Hash (Certificate || 0)* and extends it into the integrity register: *NewValue = Hash (OldValue || Failure)*. In each case the device records the result of assertion in the Device Integrity Log.

The above example illustrates how assertions can be used for any component on the device, including device drivers, OS kernels, and applications. Verified components can then assert the state of peripherals, such as whether or not a microphone is live or a camera is on. In the example, verified components assert their state by hashing a certificate along with a fixed value representing its state.

> **Example peripheral assertion generation implementation:**
> *PeripheralAssertion = Hash (Certificate || State)*. The component will then extend the assertion into an integrity register: *NewValue = Hash (OldValue || PeripheralAssertion)*. The verified component will also make the appropriate entry into the Device Integrity Log.

As the device and verified components create assertions, they shall also create policies that spell out which Information Owners (as represented by their identities) have the right to share them. For example, as the device boots, it contains few Information Owners (maybe only one, that of the Device Owner). As assertions are created they can be stored along with policies that control access to them. As new applications are provisioned and new Information Owner identities are created, the new applications may request access to existing assertions to which the Device Owner can grant or deny. When the applications are executed, the Information Owner identities are activated along with their assertion policies. If the

running applications generate additional assertions on behalf of the Information Owner, then the Information Owner controls the policies of those assertions. However, these assertions and policies are ephemeral in nature and should die when the application terminates, and be recreated when the application runs again.

4.1.3.3 Maintaining the Integrity of Assertions

An RTI maintains assertions in a protected repository. While raw assertion data could be stored in this repository, typically cryptographic hashes of assertions are stored in protected registers, with the raw assertion data stored in unprotected memory. Alternatively, the device and its applications may generate assertions and store them outside the RTI as long as a verifiable chain of trust exists from the RTI to the external storage location. In both cases, the assertions shall be protected from unauthorized writes. A protected interface that enforces authorization to extend values into the assertions provides this protection.

4.1.3.4 Sharing Assertions

The device RoTs and PEnE will enforce the policies for the sharing of assertions based on the policies provisioned at the time the assertions were created as well as the identity of the device or Information Owner requesting the assertion. The mobile device will use its RTR to sign assertions with signing keys associated with either a device identity or an Information Owner identity. The RTR will apply integrity and non-repudiation protections on the data so that subsequent software agents cannot alter or replay the data without detection. The established identities allow verifiers to verify the source of the data uniquely and without repudiation.

4.1.3.5 Collecting Assertions

A requester (e.g., an Information Owner) will generate a request for assertions of device attributes and characteristics. Software agents on the mobile device, acting in response to the request for assertions, will collect and sign them with a key associated with an identity of either the device or an Information Owner. The requester will have taken an active role (e.g., in supplying a nonce) to the software agent to guarantee freshness of the assertion signature. The software agent shall leverage the RoTs for storage, integrity, and identity to guarantee the integrity of the assertions and the authenticity of the source of the assertions to the requester. This is done by establishing and protecting identities within the RTS, generating and protecting assertions within the RTI, and signing the assertions in the RTR, all of which are located in the same trusted component.

4.1.3.6 Processing Assertions

The requester, presumably performing a remote assessment of the mobile device, participates in the protocols with the software agents on the device to ensure the confidentiality, integrity, and freshness of the signature as well as authenticating itself to the device. Once the requester receives the signed assertion, it will use the public certificate associated with the identity of the device or Information Owner used to sign the assertion to verify the signature and to obtain assurance of the freshness (e.g., using a nonce) of the signed assertion. The requester also authenticates the identity of the source of the assertions and can have confidence in the assertion just presented.

4.2 Isolation

Isolation is the capability to keep different data components and processes separate from one another. In particular, it is the ability to restrict the flow of information from one entity to another. In the case of a mobile device, some level of isolation is required between each of the different architectural contexts

described in Figure 3 in Section 4.1. This section addresses the mechanisms required to provide every Information Owner confidence that their application or data can be processed in an isolated state on the device without being adversely impacted by other applications. In order to provide this confidence, the isolation mechanisms need to be designed to leverage the RoT and PEnE.

The ability to provide an isolated context in which the Information Owner can operate or store data requires the device to be capable of executing a verifiably secure context as a starting point. Establishing this environment securely requires that the isolation mechanisms use the assertions made available from the RoTs. In the notional architecture, this environment consists of elements of the Application and Information Contexts.

The required contexts on a mobile device fall into one of two categories. Some will be unique to one Information Owner; some will be common between multiple Information Owners within the device. For the uniquely-owned context, it is easier to gain assurance in the isolation techniques that the context remains uniquely owned. However, when the context inherently straddles the boundary between owners, it becomes critical to understand the limits of the separation mechanisms built into the device. For example, a browser residing in an Application Context that connects to multiple Information Contexts (e.g., enterprise and Internet) today could transmit data across contexts. This exposes the need to safely manage the inevitable and necessary co-dependence between contexts (i.e., be multi-context aware). As the appropriateness and visibility of data sharing paths are resolved, they are then enforced by the PEnE.

4.2.1 Threats Addressed through Isolation

A primary threat to data on a mobile device today is malware exfiltrating sensitive data off of the device, resulting in the loss of confidentiality. Information Owner risk losing control of their data as soon as it is pushed to a mobile device, through intentional and unintentional compromises of that data that arise from instances of poorly behaving, compromised or malicious applications; compromised operating systems; tampered baseband processors; or improperly shared peripherals. Any complex software, including mobile applications, will have latent exploitable conditions (i.e., vulnerabilities) that could allow an unauthorized entity to gain control of the context within which the application runs, or otherwise cause the application to behave in a manner other than the Information Owner or application developer intended. Mobile devices that provide policy-managed, verifiable, and isolated contexts in which to process and store data provide confidence to Information Owners in the proper protection of their data.

4.2.2 Considerations for Data and Application Isolation

Isolation capabilities are dependent on a known, clean initial state and continuity of that state as demonstrated through device integrity assertions. However, confidence in the verity of any assertion diminishes with time because as an established context runs, the device state continues to change outside of the context. That change may or may not have a negative impact on the context. The implementation and strength of mechanisms used to isolate the contexts need to ensure that negative impacts introduced by platform changes are mitigated in a way that protects information stored or cached on the device. Additional mechanisms, such as refreshing the environment, or the assertions, can be used to maintain or regain Information Owner confidence of isolation capabilities.

A second, and critical, consideration while isolating applications and data on a mobile device is that most applications require cross-device communication or resources in order to execute. For instance, many applications may need to share, with Device Owner and Information Owner consent, a central contact list. In this case, logical sharing channels need to be established to appropriately share this information. To provide all Information Owners with a level of confidence that allows for these sharing channels to be created, the device will need to use RoTs in conjunction with a PEnE to provide the Device Owner and

Information Owners with enhanced access control and enforcement of the authorized sharing policies for communications channels.

Most mobile operating systems support some form of multitasking. As such, the operating system and multiple applications will likely be acting on multiple Information Contexts simultaneously. To properly secure this case, it is important that the platform support application hardening against control flow kinds of attacks such as (but not limited to) Address Space Layout Randomization, stack/heap canaries, no-execute permissions—a minimum bar to prevent control hijacking. However, when an Application Context becomes corrupted, the isolation mechanisms need to be robust enough that the corrupted Information Context gains no more privilege and access than it originally had, and also doesn't grant (directly or indirectly) other contexts access to information beyond their intended access levels. This global view is critical because there are no easy ways to detect or eliminate bugs in running code, especially in complex systems. Anti-exploitation mechanisms alone, however, may not provide sufficient protection. Sound isolation above the minimum bar stated here can be achieved in part through implementing the foundational isolation capabilities described in Section 4.2.3.

4.2.3 Foundational Data and Application Isolation Capabilities

In order to meet the objectives of the identified scenarios, every Information Owner on the device needs be able to establish a completely isolated environment in which to process, transmit, receive and store the data they own. The isolated environment needs to be able to provide the capabilities outlined in this section.

4.2.3.1 Creating an Environment

The OS along with the underlying firmware and hardware of the mobile device will be the basis for the secure environment that will provide many of the isolation capabilities. The device shall use the RTV to establish the secure environment used to store and process data from an Information Owner. This implies that all contexts and objects (e.g., OS services, application processes, and device hardware, etc.) within the new trusted computing base were confirmed to match the specifications of the Information Owner's policy. This may be done with assertions of the isolation primitives being used by the platform.

4.2.3.2 Device Integrity to Support Isolation

Information Owners need to be provided with assurance that the mobile device attempting to access their data is capable of providing an environment that will support their data and application isolation requirements. This assurance will be provided in the form of device integrity assertions. The mobile device shall be capable of asserting to Information Owners that the software controlling the launch of the isolation context and the management of its resources has not been modified without the Information Owner's consent. The device assertions shall meet the device integrity assertion requirements outlined in Section 4.1.

4.2.3.3 Authorized Data Sharing Channels

Some applications require network access or inter-process communication capabilities to function properly. In other cases, users may find it desirable to share data between applications, such as a shared contact list or file viewers for opening e-mail attachments. These examples show that application and data isolation cannot be absolute, and that there is a need for controlled sharing of data on devices. However, these sharing mechanisms could introduce vulnerabilities that could allow a threat to exfiltrate data off the device. To help ensure their data remains protected on the mobile device, Information Owners need to have the ability to place restrictions on the flow of their information on the mobile device.

To support this capability, the mobile device or platform shall be able to generate assertions that list all sharing channels (e.g., network interfaces, applications with network access, contexts, inter-process communications, etc.). The PEnE shall enforce an Information Owners' requirements (i.e., their policy) with respect to which channels information can be shared across. Access to information via unauthorized sharing channels shall be denied.

4.2.3.4 Policy Negotiation

Information Owners need the ability to create policies surrounding the handling of their data on mobile devices. These policies, once accepted by the Device Owner, shall be enforced by the PEnE. The isolation capabilities described above provide two examples of policies that need to be supported by the device: what state or states the mobile device must be in for the Information Owner's data to be available, and what logical channels can be used to share their data.

However, the Device Owner needs to maintain control of his or her device, and in particular, the Information Owner should not be able to assert control over the mobile device without the knowledge and consent of the Device Owner. The PEnE shall provide the Device Owner with all policies and conditions the Information Owner levies on the device for the protection of the Information Owner's data.

- The Device Owner shall be able to explicitly accept or reject those conditions prior to the instantiation of the isolation context.
- The acceptance or rejection of that condition shall be securely asserted to the Information Owner.
- The Information Owner shall be able to decide whether or not to grant access based on that exchange.

4.2.3.5 Maintaining Trust

Maintaining the trust status of the firmware and software used to enforce data and application isolation is very critical. Attackers can explore vulnerabilities in that firmware and software to compromise the isolation mechanisms in a mobile device. The device integrity capabilities identified in this document are intended to reduce the risk of a persistent malware presence (e.g., survive a reboot), but may not be capable of providing protection against temporary exploits. To mitigate this, a periodic restart of the device or the application, a re-query of the assertion, or something equivalent shall be required to maintain isolation.

4.2.3.6 Revoking Trust

Relationships between Device Owners and Information Owners are not necessarily permanent. The Information Owner shall be able to remotely remove access to the data they own on the device at any time. This capability is important for responding to lost or stolen devices, as described in Section 4.3.3.3. It may also be useful when the relationship between the Information Owner and the Device Owner is terminated. For example, the Information Owner may terminate the relationship if a Device Owner changes jobs. The Device Owner might terminate the relationship if he or she no longer finds the policy by the Information Owner acceptable.

4.2.3.7 Data Isolation

The RTS in the mobile device should be used to segregate data from various Information Owners stored on the device. This would typically be achieved through cryptographic mechanisms. If data isolation

capabilities are not provided by the platform, the capabilities described in Section 4.3 would provide applications with the ability to create and enforce this segregation of data.

4.3 Protected Storage

Mobile devices need to maintain the confidentiality and integrity of Information Owners' data while operating in a diverse environment consisting of potentially untrusted applications and a range of possible levels of authorization. In order to provide proper data protection in this diverse operating environment, protected storage mechanisms are required. Protected storage typically depends heavily on encryption and integrity protection, leveraging standardized algorithms and functions to protect data and the associated keys with the authentication credentials of authorized entities. Some of the most significant risks to protected storage in a diverse operating environment that leverages encryption include insecure storage of keys and the exposure of keys in non-volatile storage during or after use.

In order to support enhanced assurance for key storage, retrieval, and use, a mobile device can implement a service that supports storing, retrieving, and using keys for encryption, decryption, integrity protection, and verification while minimizing exposure of the keys outside of the service. Rather than depending on each application to perform key wrapping securely with each user's credentials, the use of the RTS facilitates a secure wrapping infrastructure that is common across applications. The availability of the RTS may also reduce the level of effort and expertise required for application developers, and enable the proper use and storage of encryption keys as a standard practice. Protected storage and key protection services both ensure proper authorization to retrieve keys and minimize exposure of keys to a localized level. The RTS also provides a central place for an Information Owner to revoke access (by removing access to the encryption keys).

In addition to cryptographic means, protected storage provides confidentiality and integrity by restricting access to information to authorized system entities (users, process, devices) through physical and logical mechanisms. Physical protections restrict access to keys to approved operations by authorized entities. Logical protections are access control rules, typically implemented in software or firmware that permits only authorized access to an object by an entity.

4.3.1 Threats Addressed through Protected Storage

Protected storage defends against a range of threats. One of the most significant threats to data on mobile devices is theft and loss resulting in exposure of sensitive data. Data at Rest (DAR) solutions predicated on robust protected storage mechanisms counter this threat. More information regarding the risk environment and considerations for broad classes of devices can be found in NIST SP 800-111, *Guide to Storage Encryption Technologies for End User Devices* [SP800-111].

DAR solutions apply to data on storage media that is not currently traversing a network, residing in processor or memory space, or being read or written. The device may be in one of multiple possible locking states when the device is separated from the Device Owner, and the access and authentication behavior for each power state is a critical policy decision that must balance security and usability. A comprehensive policy also should address the strength of the authentication factors to ensure that the values cannot be guessed or found through brute-force attacks. The principal DAR protections are cryptographic, and the assurance depends on the use of standardized and generally accepted mechanisms for cryptographic operations, as well as independent validation, such as [FIPS-140].

The second threat is software applications (possibly malicious) unauthorized access to critical security parameters resulting in loss of confidentiality and data integrity. Protected storage supports the principle

of application isolation discussed above. The RTS allows only authorized entities to access the cryptographic services, and may prevent even authorized entities from being able to gain direct access to retrieve keys generated and stored by the RTS (and therefore only allow those authorized entities to retrieve exportable keys protected by the wrapping key). The RTS can also provide features such as secure storage and use of asymmetric keys used for device identification and integrity assertions.

A third consideration is when a change in the threat or usage environment necessitates revocation of access to information in protected storage. For example, a mobile device may fall into the hands of an unauthorized user who attempts to repeatedly guess a password, or the authorized user may have their access revoked for whatever reason (e.g., termination of employment). In these cases (and a range of others), the confidentiality of the data is enhanced by the ability to selectively sanitize the Information Owner's data from the device.

4.3.2 Considerations for Protected Storage

An Information Context is considered locked if the user cannot access its data. Conversely, an Information Context is considered unlocked if the user can access it. Upon power-up, Information Contexts should originate in a locked state. Information Contexts may transition from the locked state to the unlocked state if the Information Owner provides the correct authentication and the PEnE is satisfied. Information Contexts may transition from the unlocked state to the locked state in a variety of scenarios, e.g., whenever the user decides, the PEnE decides as a result of policy enforcement, or the device is powered off. Whenever an Information Context is locked, memory-present data encryption keys or key encryption keys, if any, shall be zeroized. Upon power-down, devices should transition unlocked Information Contexts to the locked state. The Information Owner may set policy through the PEnE to define storage protection behavior as the device transitions between each pair of different states.

4.3.3 Foundations for Protected Storage

In order to effectively apply policy-based protected storage, the Information Owner needs to identify data protection needs and policies. The Information Owner would begin by categorizing the confidentiality level for the types of data processed by the information system using a categorization standard such as FIPS 199 [FIPS-199]. NIST SP 800-60 is also available to help support mapping the types of information to security categories [SP800-60]. This document assumes that the Information Owner has categorized the data types to be processed by mobile devices and identified any special handling needs such as for the protection of healthcare data, financial data, or personally identifiable information (PII) as identified in NIST SP 800-122 [SP800-122].

4.3.3.1 Key Generation

The strength of cryptographic mechanisms relies upon two basic components: the algorithm used to encrypt or protect the data, and the cryptographic key used by the algorithm. NIST has developed a wide variety of Federal Information Processing Standards (FIPS) and NIST Special Publications (SPs) that specify and approve cryptographic algorithms for Federal government use. These algorithms, when implemented appropriately with properly generated cryptographic keys and other parameters, are capable of providing the required protection for the data.

Proper generation of cryptographic keys is critical to the security of the cryptographic mechanisms, as the security of the cryptographic algorithm relies upon entropy provided in the key. NIST SP800-133, *Recommendation for Cryptographic Key Generation*, provides guidelines on the generation of keys used with symmetric and asymmetric cryptographic algorithms [SP800-133]. All cryptographic keys shall be generated based directly or indirectly on the output of an approved Random Bit Generator, such as those

specified in SP800-90A, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators* [SP800-90A] seeded with sufficient entropy to support the target security strength of the key.

While it is preferable to store cryptographic keys in protected hardware, or encrypt them with keys that are stored in protected hardware, mobile devices may, in some cases, use user-entered passwords to protect cryptographic keys stored on the device. Because such passwords typically have low entropy and weak randomness properties, these passwords shall not be used directly as cryptographic keys. NIST SP 800-132, *Recommendation for Password-Based Key Derivation*, provides guidelines on the use of a family of password-based key derivation functions that may be used to derive cryptographic keys from passwords for the protection of data encryption keys or key encryption keys [SP800-132].

4.3.3.2 Key Protection

Key protection shall be provided by the mobile device as a service, starting with one or more device keys that are contained in the RTS. In general, we consider two classes of keys : data encryption keys (DEKs), which are are keys that encrypt data, and key encryption keys (KEKs), which are keys that wrap other keys. Some KEKs are RTS-protected keys that in return protect DEKs and other KEKs. DEKs may be used for per-context encryption (e.g., separate keys for each Information Owner's data), per-application encryption, data or file encryption, or any other encryption operation where an application-unique key is desired. The key protection service provides confidentiality and integrity through functions for wrapping, unwrapping, and loading cryptographic keys for use. The service could be implemented as a purely software-based service or with support from hardware such as the Trusted Platform Module described in the multipart ISO/IEC 11889 standard [ISO11889].

While an RTS is well suited for protecting KEKs, DEKs are at risk for exposure since they are often unwrapped for use outside the RoTs. For example, bulk encryption/decryption operations are executed outside the RTS by the application processor on a mobile device. Isolation, as described in Section 4.2, mitigates the risk of exposure for DEKs.

If an RTS has the capability to generate keys, they could be either exportable or non-exportable. All keys generated outside an RTS are considered exportable with respect to the RTS. An RTS can protect both exportable and non-exportable keys. Additionally, the RTS shall not export non-exportable keys under any condition. In the case where a malicious user-installed application is running alongside an application processing the Information Owner's data, there may be a risk of exposure of DEKs to the user-installed application. The other DEKs protected by non-exportable KEKs within the RTS remain protected.

The RTS also provides a central place for revocation of access to the Information Owner's data. In the event an employee is separated from employment, the Information Owner could issue a command to remove access to the non-exportable wrapping key(s) for that user. The Information Owner has assurance that the key could not have been escrowed by the user (because they were never able to retrieve it directly) and the data encrypted by keys that are protected by the non-exportable key is subsequently inaccessible to the user.

4.3.3.3 Remote Sanitization

Sanitization is the process of rendering access to data stored on media infeasible. NIST SP800-88rev1, *Media Sanitization Guidelines*, identifies three methods for sanitizing data that offer different levels of protection [SP800-88]:

- **Clear**- a method that applies logical techniques to sanitize data for protection against non-invasive attacks. These methods typically involve the use of standard read/write commands to overwrite data.
- **Purge**- a method that applies physical or logical techniques to render data recovery infeasible against state-of-the-art laboratory techniques.
- **Destroy**- a method that renders the media unusable and renders data recovery infeasible against state-of-the-art laboratory techniques. These methods typically involve physical destruction of media through incineration, shredding, pulverizing or disintegration.

In the context of this document, remote sanitization refers to the ability of the Information Owner to render its data stored on the mobile device inaccessible to the Device Owner or any other entity with logical or physical access to the device. In this case, what is needed is the ability for Information Owner to render their data, and only their data, inaccessible. Data under the control of the Device Owner, or other Information Owners, is not impacted by a sanitization by an Information Owner. This ability is sometimes referred to as a selective wipe. The platform shall provide the Information Owner with the ability to remotely sanitize their data stored on the device, making use of capabilities and services provided by the RTS and PEnE.

Clear or Purge methods could be employed to accomplish this form of selective sanitization. NIST SP800-88r1 provides guidelines on the use of the clear and purge sanitization methods. These guidelines include material covering the technique of overwriting data to sanitize data according to the clear method, as well as the use of cryptographic erase as a sanitization technique that involves wiping the cryptographic keys that encrypt the Information Owner's data [SP800-88].

4.3.3.4 Data Lifecycle

If an Information Owner wants to protect confidential information on mobile devices, they may choose to encrypt it. They could generate keys on the mobile device or generate them centrally and provision them to the mobile device. In either case, the mobile device utilizes the RTS to directly or indirectly protect both KEKs and DEKs. The PEnE enforces the binding of these KEKs and DEKs to their particular Information Contexts. That is, the Information Owner controls access to its data by PEnE-enforced control of the KEKs and DEKs within the Information Context.

Device Owners shall not be able to export keys without the explicit authorization of the Information Owner. Finally, Information Owners shall retain the ability to revoke access to data in the Information Context utilizing a data sanitization method described in Section 4.3.3.3.

Appendix A—Acronyms

Selected acronyms and abbreviations used in the guide are defined below.

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
API	Application Programming Interface
BYOD	Bring Your Own Device
DAR	Data at Rest
DEK	Data Encryption Key
DIL	Device Integrity Log
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Management Act
GPS	Global Positioning System
IT	Information Technology
ITL	Information Technology Laboratory
KEK	Key Encryption Key
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
OMB	Office of Management and Budget
OS	Operating System
PEnE	Policy Enforcement Engine
PII	Personally Identifiable Information
PIN	Personal Identification Number
RFC	Request for Comments
ROM	Read Only Memory
RoT	Root of Trust
RTM	Root of Trust for Measurement
RTI	Root of Trust for Integrity
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
RTV	Root of Trust for Verification
SP	Special Publication
WLAN	Wireless Local Area Network

Appendix B—References

Selected acronyms and abbreviations used in the guide are defined below.

[FIPS-140] FIPS 140-2, *Security Requirements for Cryptographic Modules*. May 2001.

[FIPS-199] FIPS 199, *Standards for Security Categorization of Federal Information Systems*. March 2004.

[ISO11889] ISO/IEC 11889-1:2009. *Information technology- Trusted Platform Module- Part 1: Overview*. 2009.

[RFC2119] Request for Comment 2119, Key words for use in RFCs to Indicate Requirement Levels, Network Working Group. IETF Current Best Practices Track, March 1997.

[SP800-60] NIST SP 800-60rev1, *Guide for Mapping Types of Information and Information Systems to Security Categories*. August 2008.

[SP800-88] NIST SP 800-88rev1 (Draft), *Guidelines for Media Sanitization.* September 2012.

[SP800-90A] NIST SP 800-90A, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*. January 2012.

[SP800-111] NIST SP 800-111, *Guide to Storage Encryption Technologies for End User Devices.* November 2007.

[SP800-122] NIST SP800-122, *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII),* April 2010.

[SP800-131A] NIST SP 800-131A, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*. January 2011.

[SP800-132] NIST SP800-90A, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*. January 2012.

[SP800-133] NIST SP 800-133 (Draft), *Recommendation for Cryptographic Key Generation*. August 2011.