# Evolution of Trust Systems from PCs to Mobile Devices

Dolores Howry, Yogendra Shah PhD, *Member, IEEE,* InterDigital Inc.
Andreas U. Schmidt , *Member IEEE,* Novalyst IT AG

***Abstract*— A trust ecosystem for mobile devices based on trusted computing principles is presented. The trust ecosystem model from the latest PC platform provides a reference for the evolution of a trust model for the diverse mobile device market. By balancing the trusted computing burden between the mobile device and network, and separating service delivery/access control decisions and error reporting from remediation, the processing in the network is hugely simplified together with an increased efficiency in the use of network resources. Additionally, certification and accreditation of the mobile device's trust security capabilities provides a white box view into a mobile device's security architecture and a means to gain a high level of trust assurance in the mobile device.**

***Index Terms*— trusted computing, trusted ecosystem, device integrity, trust certification, trusted execution environment, secure boot, access control, remediation**

## I. Introduction

Security and trust have become a top priority for PC platforms in recent years prompting companies such as Intel and Microsoft to strive to provide more secure platforms for their users. The ability to ensure privacy of end user information as well as the assurance of trusted operation of a PC is of paramount importance in the design of future PC systems. As users move to mobile devices (e.g. tablet computers and smart phones) and become more dependent on the use of Internet technologies to access data and services, the value of trust security for mobile systems must also keep pace with the increased threats to user data and the secure use of Internet services. Established trust security from PC platforms may be extrapolated and then incorporated into mobile devices to develop a complete trust security ecosystem for mobile devices.

As wireless networks become more and more distributed, new security threats are emerging. Endpoint protection is becoming increasingly necessary in terms of the security architecture for distributed networks. Unprotected network edge components are susceptible to attacks and will require stronger security as part of an overall network security architecture. For example, cellular systems employ Femto cells, relay nodes and gateways to interface with mobile devices for access to mobile networks. These edge devices are not protected by the typical garden wall security afforded to other network nodes, leaving them vulnerable to multiple vectors of attack, impacting the availability and operational aspects of the wireless networks.

Likewise, mobile device security also needs to be bolstered. Data stored on mobile devices and the services they access require protection. Mobile applications provide a variety of services including services that require a significant level of security such as online banking and mobile payments. A device infected by malware undermines the trust assurances the user expects and can have costly consequences to both the user and the service provider. Innovations in the PC security model provide a reference for the evolution of platform trust security for mobile systems. PC security architectures are anchored on the Trusted Platform Module (TPM) defined by the Trusted Computing Group (TCG) [1], which is based on trusted computing technologies, enabling safer and consistent behavior that offers improved security for PC systems [2].

Transitioning the conventional PC trust security model to a mobile system requires an adaptation of the security architecture since TCG techniques remain impractical from the standpoint of application to broad consumer markets. The TCG Mobile Phone Working Group (MPWG) is tasked to transition trusted computing technologies into a mobile TPM (mTPM) architecture suitable for mobile systems. However, the work to date has not addressed network interactions which place a heavy burden on the network. Wireless network operators are already burdened by heavy traffic volume that will only worsen as consumers move to more and more dependence on wireless communications for their connectivity and with the growth of the Internet of Things. Mobile devices can benefit from the advanced trust assurance security features architected for the PC environment, by evolving and incorporating these into a trusted ecosystem for mobile systems. Creation of a more practical approach to trust security is possible by separating access control decisions from remediation in the network and by balancing trust processing between the mobile device and the network.

## II. CURRENT PC ARCHITECTURE

### *A. Overview*

The latest PC security architecture is based on the Unified Extensible Framework Interface (UEFI) specification and TCG TPM technologies, utilizing hardware and software enhancements leveraged by the Windows 8 OS framework [3]. This model provides an ideal reference to transition to mobile systems. Fig. 1 illustrates the Windows 8 platform integrity architecture as a series of stages. A secure boot process initiates the platform startup. The trust assurance in the secure boot process is based on an immutable Hardware Root of Trust (RoT). The platform incorporates a TPM to provide a Root of Trust for Measurement (RTM) and a Root

of Trust for Storage (RTS). At boot time, the platform utilizes the TPM to provide secure measurements and storage for various architecture components [4]. As identified in Fig. 1, the TPM computes the measurements for the Pre-OS stage components and then protects the measurements cryptographically for storage outside the TPM in the measurement log file. At completion of Stage 1, control is transferred to the kernel and an Early Launch Anti-Malware (ELAM) component in Stage 2. Additional measurements are taken according to policies defined for the platform. These measurements are also protected by the TPM and stored in the measurement log file. At the completion of the Windows logon process the platform anti-malware (AM) client is provided with the measurement log protected by the TPM in Stage 3. The detailed component measurement report is then sent to a remote entity to complete the platform attestation process in Stage 4.

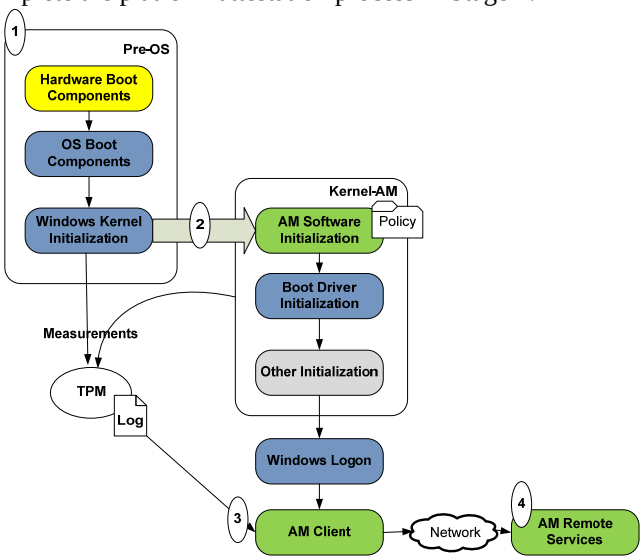


Fig. 1. Measured Boot and Anti Malware Components

### B. *UEFI Advantages*

The PC platform architecture has been further enhanced in Windows 8 by the mandatory implementation of UEFI based firmware security backed by a hardware anchored RoT. The UEFI firmware resides between the hardware and OS level, replacing the conventional BIOS in the Pre-OS boot stage in Fig. 1. The UEFI firmware represents a collection of drivers, OS loaders, tables and applications. It enables the measurement and reporting of the platform firmware integrity. The UEFI firmware provides several advantages including the provisioning of a flexible pre-OS environment enabling applications and drivers to execute in the boot environment with few constraints [5], [6]. Additionally, with no functional OS available, UEFI can provide a full networking protocol stack to enable diagnostics and firmware upgrades as well as repairs to the operating system, by contacting and authenticating to a remote server.

### C. *Early Launch Anti-Malware (ELAM)*

The ELAM feature provides a Microsoft-supported certificate based launch of AM software prior to the loading of any third-party components. The benefit of activating the AM drivers first is that loading of subsequent drivers is under the control of the AM software. The ELAM is the first application loaded by the OS and controls the staged bring-up of the remaining applications on the platform. The ELAM provides the platform an ability to control the loading of applications whose integrity may be in question, as assessed through direct integrity checks of the firmware and certificates. The ELAM boot driver measurements are stored in the TPM for later attestation with a remote AM server.

### D. *Anti-Malware Client*

The AM client on the platform is the component that works in conjunction with the ELAM software to facilitate measurement reporting. The AM client provides communication with a remote AM server. The AM client sends a log of the measured components, collected during the boot process and stored in the TPM, to the AM Server for remote attestation. The stored measurements include measurements taken in the Pre-OS stage, facilitated by UEFI firmware, as well those taken by the ELAM. The report provided by the AM client contains detailed component information that is specific to a particular platform.

## III. MOBILE DEVICE TRUST SECURITY ARCHITECTURE

### A. *Requirements for Trust Security in Mobile Devices*

The argument for a strong, yet flexible, trust security foundation for mobile devices derives from a few key drivers:

- The diffusion of malware on mobile devices today poses a significant threat to the take-up of secure services on mobile platforms [7].
- Although resource constraints in terms of processing power and memory are mostly alleviated on current mobile devices, cost and battery power consumption remain an issue, which impedes the diffusion of traditional signature based malware detectors.
- With the growth of Bring-Your-Own-Device (BYOD) scenarios, a lost or stolen mobile device puts sensitive corporate and user data at risk [8].
- User experience to access data and online services on mobile devices is different than on PCs. Mobile users are constantly connected to multiple services simultaneously with small and frequent transactions rather than a focused activity moving from one service to another.
- Bandwidth remains an issue for mobile devices and access networks (ANs), which should be considered in any security architecture, even in 4G and LTE system contexts.
- Mobile devices may be compromised on a large scale (millions) and therefore pose a risk to maintaining normal network services and operations [9].

The trust security architecture requirements for mobile devices can therefore be outlined as:

- A strong trust foundation encompassing the architectural and functional design of mobile systems, including evaluation and certification of the mobile devices.
- Trusted Execution Environments (already present on current devices, such as the UICC or ARM's TrustZone architecture, and upcoming potential candidates such as the mTPM) and hardware trust anchors to effectively establish device trust security.
- Fine-grained diagnostics of mobile devices to enable trust to be established at various levels, including baseband, OS, and user space Apps.
- Efficiently perform diagnostics and allow access to services to be handled remotely, e.g., upon network connection or service access.
- Remote remediation of diagnosed security issues, and/or containment of compromised software components preferably automated.
- Balance required resources between device and network. Minimize the impact of the trust security framework on the different AN components.

### *B. Practical Limitations*

Standard platforms such as PCs have a security architecture based on TCG technologies. The traditional TCG Trusted Network Connect (TNC) architecture requires the network to obtain detailed knowledge of the platform component measurements, which enables both access control and remediation [10]. The network must process the reported component measurement log from the PC platform to assess the trust state of the platform, in order to make a service delivery or access control decision, a task similar to the diagnostics for a remediation process. This process results in a heavyweight communications interface between the platform and the network with extensive processing on the network side. Though this burden may not be of great concern in PC based architectures, a migration to high volume consumer markets such as mobile devices reporting such measurements would place a tremendous traffic and processing burden on the network. Furthermore, with hundreds of mobile devices currently available and new handset designs coming onto the market every 12 to 18 months, the task of maintaining component measurement databases becomes impractical.

While the TCG TNC technologies have merit in a standard Intel/Microsoft Windows based PC architecture, the same does not apply to mobile phone platforms where there is a large diversity of processors and architectures. Mobile devices have many varieties of hardware architectures with each chipset manufacturer having a proprietary processor architecture which included various ARM variants, Intel ATOM, TI OMAP, Qualcomm Snapdragon, etc. Therefore the network operator lacks external visibility into a mobile platform's proprietary security architecture and how a particular manufacturer has implemented trust security capabilities in their mobile platform. In addition, no evaluation criteria exist to determine the security architecture of a device or how a secure boot process has been implemented, resulting in a low level of trust assurance in the mobile device.

### *C. Evolution of PC Trust Security to Mobile Devices*

Decoupling the service delivery/access control decisions from the remediation function and distributing the trust processing between the network and mobile device in a balanced manner provides for an improved trust processing and access control mechanism. Evolution of the PC trust security model to such device trust assessment architecture requires adaptation of the TCG approach to measurement and reporting. Equipping the mobile device with the ability to measure the security state of the platform in a trusted manner and report information suitable for access control policy decisions reduces the burden the network operator. Platform validation is the ability for a device to measure its own "health" (or its integrity) and report the results to the network operator to efficiently assess the trust state of the platform. Following an access control decision, failure reports are forwarded to the device management server. Blending the remediation function into the device management (DM) server, where intimate knowledge about the device configuration already exists, provides for a more efficient mechanism for detailed platform remediation.

Similar to the UEFI and measured boot process in a PC architecture, platform validation is a process where a secure boot brings-up the platform through performing a series of trustworthy self-checks on components of the system and then reports them to the network to complete the trust assessment and perform access control decisions. The mobile device performs secure self-administration of policy based component load control and validation measurement reporting as part of the secure boot process, which simplifies the reporting of trust state information and reduces the burden on the network in performing an assessment of the trust state. Moreover, the network can notify the device management server to trigger remediation of a compromised device. The device management server can further interrogate the mobile device to isolate component failures reported during the access control process to perform remediation under conditions that are beneficial for both the device and the network load.

The proposed mobile device trust security architecture distributes trust based processing on two core conceptual elements - a secure boot process anchored on a hardware RoT and a trusted execution environment (TEE) for trust measurement, evaluation and reporting. The TEE provides an environment that can be trusted to perform security sensitive operations and store data securely. The ARM Trust Zone is an example of a prime candidate for inclusion in a mobile device architecture. The mTPM, which is a TPM tailored for mobile applications, and the Global Platform TEE are other suitable technologies [11]. The Trusted Software Stack (TSS) that provides a standardized API to access the TPM is another technology that can be adapted towards the mobile embedded system environment [12].

Finally, the lack of visibility into the mobile device security architecture can be addressed through white box certification. Certification provides a means to circumvent the issue of visibility into the security architecture and secure boot process giving network operators a higher level of trust assurance for the mobile device.

## IV. ELEMENTS OF PROPOSED TRUST SECURITY ECOSYSTEM

Fig. 2 provides a view of an ecosystem for remote trust assessment in mobile devices. At the heart of the mobile device trust security is a Trust Module incorporating a SEE anchored on an immutable hardware Root of Trust. This provides the core architectural basis for the mobile device trust security model. The strength of the Trust Module is its ability to execute a policy based platform bring-up through a chain of trust. The secure mobile device incorporates independently certified hardware and software components enabling external entities such as the network operator to establish trust in the device. A certified hardware Trust Module provides a layered approach to device trust certification and a path to meet the rapid development cycles for products targeted towards the mobile market. Certified software components provide additional means to establish trust in the devices to remote entities.

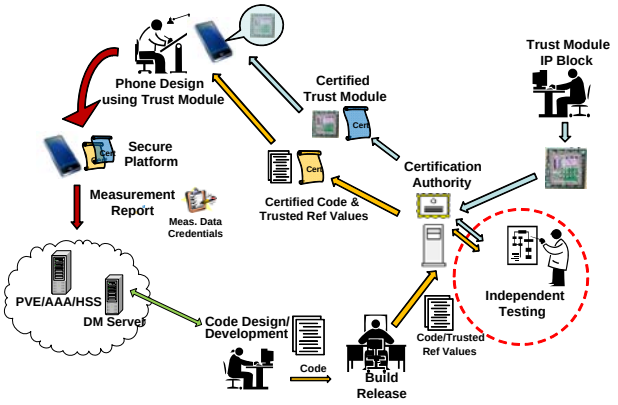


Fig. 2. Ecosystem for Establishment of Trust in Mobile Devices

The second element of the trusted ecosystem is the access control entity. In a cellular wireless system the proposed access control entity is called a Platform Validation Entity (PVE). The PVE can be considered an extension to the wireless network's Authentication, Authorization and Accounting (AAA) and Home Subscriber Server (HSS) function and tightly coupled with authentication procedures. The role of the PVE is to validate the trust state of the mobile device and make fine-grained access control decisions based on the results of the integrity checking measurements and certification information obtained from by the mobile device. The result of the validation process enables service and content providers to gain trust in the mobile device to provide the requested services and content. A description of the reported functionality is provided in a later section.

The third element in the trusted ecosystem is the device management server. When performing device validation, in the event of a measurement report indicating a failed functionality, the PVE informs the device management server of the failure. The device management server is enhanced to perform remediation through interrogation with the device management client on the mobile device to isolate specific failed software components and provide secure software updates. A detailed description of the remediation process is provided in a later section.

## V. SECURE MOBILE DEVICES

Moving beyond the current mobile architectures, the secure mobile device design provides trust assurance through a white box view into the internal security architecture of the mobile device, traceable to trusted third parties. The high level of trust assurance is achieved through a defined trust evaluation and certification process for both hardware and software. The integration of certified hardware and software elements enables the delivery of secure and trusted mobile devices. Pre-certification of hardware and software enables mobile device manufacturers to develop trust certified products while maintaining the rapid development cycles for a diversity of mobile products.

### *A. Trust Module Certification*

In order to establish trust in a mobile device, the security architecture of the platform needs to be understood and assessed. Establishing trust in a mobile device requires an ability to peer into the platform trust security processing including the RoT and trust measurement and reporting capabilities. Trust in the Trust Module may be obtained through an independent trusted third party test and certification authority. The Trust Module is capable of performing trusted measurements, secure storage and maintains the ability to trigger alarms and send reports to the network based on the integrity check results in the event a full protocol stack is unavailable, much like the UEFI in a PC architecture. Additionally, the device management client is tightly integrated into the trust system to enable remote network driven interrogation and remediation.

The certification of the Trust Module can be executed prior to and independent from the design of the secure mobile device. Independent hardware testing and certification creates a white box view of the Trust Module security which can be done in parallel to the software certification process. In many cases a new mobile device may not require a new trust module so by layering the certification process for the mobile devices, manufacturers can use pre-certified elements followed by an accreditation process to build new platforms leading to faster time to certification for the end product. Certifications including FIPS for cryptographic functions and Common Criteria for platforms provide standardized evaluation methods that enable a white box view of the trust security capabilities in a mobile platform.

### *B. Software Certification and Signing*

Similar to the hardware certification process, the software certification and signing process also contributes to the complete white box view of the mobile device's trust security architecture. The certification process includes the hashing of software components to be integrity checked and embedding the signed values in the form of trusted reference values (TRVs), which are typically a cryptographic hash of a code block, into the code release. As in the case of the hardware certification, independent software testing and certification provides trust assurances of the complete software security architecture.

The integrity validation procedures require the software to be compared to TRVs. Similar to the certified measurement of the RoT in a secure boot operation, the trusted reference

values are the expected results of the measurement of a software component. The software Integrated Development Environment (IDE) and Build Release tools are enhanced to provide an automated mechanism for embedding trust information into software releases. The software build and code release tools enable the device processor to extract trust information for software integrity checking and error reporting [13]. The boot and program loader use the TRV information embedded in the code image/components during the build process to perform integrity check measurements and compare to expected measurements. Measurements are compared to the embedded TRVs and the measurement results are logged and associated with functions being tested through tracing a software component to a function.

### *C. Trusted Assurance through Measurement Reports*

The combination of hardware and software certification processes on the mobile device enables the network to obtain a detailed view into the security architecture of the device. Therefore the ability to provide detailed security architecture information of the mobile device mirrors the security architecture for Windows 8 based PC platforms. This information can be used by the Mobile Network Operator (MNO) to establish trust in a particular mobile device through evaluation of the certification and tracing back to the trusted third parties and certificate authorities. In the proposed method the secure platform extracts embedded trusted reference values and uses the trust module to verify the integrity of the hardware and software. The secure self-measurement result reporting along with hardware and software security credentials establishes trust between the device and MNOs. The proposed measurement reports differ from the traditional TCG measurement logs. Instead of sending a measurement log or a list of component hashes for the network to further evaluate, a list of platform functionalities is reported. Each functionality maps to a collection of measureable components. For example, an integrity check failure of software components of a Netflix application will result in a failure result being reported for the Netflix functionality.

This mapping process from a specific component to a function enables the device to send a platform independent “self-health check” with trust assurances provided by the security certificates to enable the network to perform a platform trust validation and an access control policy decision. This process replaces the network centric TNC attestation process with a balance of trust measurement and assessment between device and network. The network does not need to perform attestation on a reported measurement log in order to perform an access control decision. In moving from a remote attestation process to a more autonomous attestation procedure reduces network traffic and the network processing burden [14]. More importantly the need for the network to have a detailed knowledge of the process involved in creating a measurement log are hugely simplified by having the mobile device perform this task itself

### *D. Trust Measurement and Fine Grained Access Control*

The certification process defined earlier enables the mobile device to send trusted measurement reports to the network that enable the network to make fine-grained access control decisions. The measurement reports differ from traditional attestation reporting in TNC in that there are no measurements sent in the report, rather the status of the integrity check. As part of the secure boot process, the device performs local measurements of software components. Each component measurement is compared to the corresponding TRV and a pass or fail result is stored in the secure execution environment. The trust module applies pre-installed policies to enable loading of software components based on the trust measurements. Components that have failed the integrity check are not loaded and marked by the trust module as failed. At the completion of the secure boot process, the device sends the measurement report to the network as part of the authentication process. The measurement report does not contain a measurement log as in traditional remote attestation, but rather a list of failed functionalities represented through a mapping of failed components [15]. The notion of component to functionality mapping can be viewed as a physical to logical view of the software as shown in Fig. 3. The physical view corresponds to the representation of the components in the devices memory map which includes a start address and a size. The logical view maps a group of components to the corresponding functionality performed. For example, the Netflix application can comprise numerous measureable components that support the Netflix functionality on the mobile device and any one of these components may be compromised and fail an integrity check. It should be sufficient for an access control entity to know that the Netflix application was compromised rather than the specific component of the Netflix application. This approach distills the detailed measurements into sufficient information to enable access control decisions to be completed and at the same time relieves the network from performing a detailed analysis of the measurement log to identify the severity of an integrity check failure and then apply access control policy decisions.

The mapping of component to functionality may not have a one to one correspondence, but rather multiple components can map to the same functionality. The component mapping can be device manufacturer specific since each device can have a unique component build and memory allocation which represents a standard functionality across many mobile devices. The network operator may define the device policies regarding loading and reporting of failed functionalities and actions to be taken when encountering an integrity check failure of a component. The functionalities supported by the mobile device can be grouped to define a policy required for the trusted and non-trusted execution of the device as defined by the network operator. For example, the mobile device may execute trusted and non-trusted applications in separate environments. The result of the integrity checking enables gating of components to be loaded. Components that pass integrity checks are considered trusted, while those that fail integrity checks are considered non-trusted. Applications can consist of multiple components but identified by the functionality they provide. The identification of

functionality enables a level of abstract reporting from the mobile device without providing unnecessary details of the component specifics to the network. The results of the integrity measurements are securely reported along with detailed certification information to an access control entity in network.

At the end of the boot process, the OS loader performs a translation from software component to functionality and reports a detailed functionality measurement report to the Platform Validation Entity (PVE) in the network including the hardware and software certification information [16]. Providing the network with the security capabilities of the mobile device, through the certifications, assures the network that the mobile device can be trusted to perform a secure platform bring-up and that the reported information is trustworthy. By extension, real-time integrity checking procedures may use the same functionality measurement reporting to provide an on demand policy function.

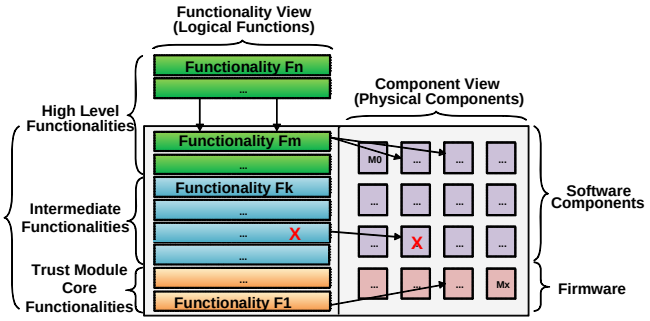


Fig. 3. Functionality to Component Mapping

The PVE uses the fine-grained trust evaluation report to establish trust in the device and provide access control information to the network. The access control options may include granting full or partial access to content and services (e.g. emergency calling only) or to quarantine the device (no access) and subsequently perform remediation. The network does not receive any detailed information regarding the measurements of components on the device but rather a logical report on the "health" of the device anchored by a hardware anchored RoT. This information is sufficient to perform access control on a mobile device without burdening the network.

The benefit of reporting the status of a functionality versus a measurement log provides substantial relief to the network in performing service delivery/access control decisions and a reduction in network traffic whilst maintaining the ability to access the measurement log information through the separated out device remediation function.

### *E. Efficient Device Management and Remediation*

In addition to making service delivery/access control decisions, the PVE may also contain policies that can trigger or schedule remediation procedures in the case of a failure report. In the event remediation is triggered, the PVE forwards the trust measurement report to the device management server that reverse maps the functionality measurement errors to a set of components on the device. Embedded trust measurement information in the code build process allows the existing device management server to expand its capabilities to readily identify failed components and remediate only the failed components. The expanded device management server in turn performs additional interrogation of the device to isolate the specific failed component(s). The device management client on the mobile device is tightly coupled to the trust system architecture, which enables the device to provide additional trustworthy integrity measurements at the request of the device management server. The device management server may chose to provide a full component update or to further interrogate the device to isolate errors. These additional integrity measurements enable the device management server to isolate specific subcomponents of the failed component. In many cases the overhead incurred in the interrogation process may be smaller than a full component update making the process more efficient in terms of traffic overhead than a complete component update. Fig. 4 illustrates the access control and remediation aspects of the ecosystem.

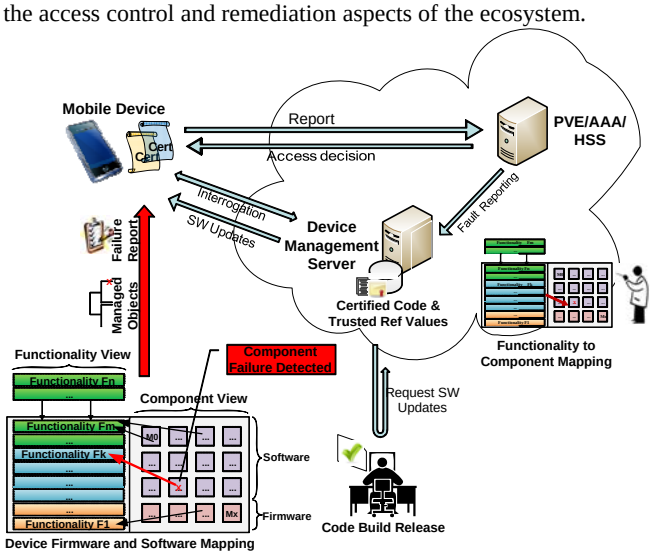


Fig. 4. Device Measurement Reporting and Remediation

## VI. CONCLUSION

The trusted ecosystem described in this paper provides numerous benefits for the various system stakeholders. Migration PC trusted platform architecture to a secure mobile device provides the core architectural basis for an embedded mobile device trust ecosystem. Trust in a mobile device is created through a white box view, anchored on an immutable and verifiable hardware Root of Trust that is assured through independent testing and certification of the platform security architecture. A layered certification and accreditation process enables platform developers to achieve the aggressive time to market requirements for mobile devices through a faster end product certification. Remotely configured policies, governed by the network operators, allow flexibility in trust establishment during the early stages of platform bring-up. A distributed and balanced trust measurement capability where the device performs a trustworthy self-assessment along with higher level measurement reporting to the network unburdens the network in terms of processing and radio resources. The mobile device provides the proof of self-test and compromised functionality in a measurement report to facilitate the network to perform decisions on fine-grained access control and service provisioning. Furthermore, the reporting of failed functionality instead of failed

components reduces the size of the measurement reports and enables the separation of the access control function from the remediation process. On the device side, the device management client is tightly integrated into the trust system to enable remote network driven interrogation and remediation. The device management server utilizes the measurement report from the Platform Validation Entity to perform targeted interrogation to identify which specific software components are compromised and then to perform efficient remediation. The overall approach results in a lightweight network protocol, simplified network processing, and reduced down time for mobile devices.

The present and the related publications [14 – 16] provide a comprehensive conceptual framework for trust security management on mobile devices. Beyond theory, we have begun validation of these concepts in practice. In [17], we exhibit the remote validation and remediation of an Android-based device, augmented by trust security capabilities to enable granular fault detection, reporting, and remediation with a PVE and DM server. We have also designed and demonstrated a method, using hash trees [18], to boost performance and save bandwidth, during the interactive diagnostic procedure between the PVE, DM server, and mobile device.

## REFERENCES


[1] Trusted Computing Group, “TCG Specification Architecture Overview TCG Specification Revision 1.4”, Trusted Computing Group: August 2, 2007.

[2] Sadeghi, Ahmad-Reza, Stuble, Christian, “Property-based Attestation for Computing Platforms: Caring about properties, not mechanics”, Proceeding of the 2004 workshop on New Security Paradigms, pp. 66-77.

[3] Microsoft Corporation, “Trusted Boot: Hardening Early Boot Components against Malware”, whitepaper, September 13, 2011.

[4] Trusted Computing Group, “TCG infrastructure working group, architecture-Part II- Integrity management TCG Specification version 1.0 Revision 1.0”, Trusted Computing Group 2008.

[5] Microsoft Corporation , “UEFI and Windows”, whitepaper, March 28, 2012.

[6] UEFI, “Unified Extensible Firmware Interface Specification”, Version 2.3 Errata E, April 26, 2011.

[7] C. Orthacker, P. Teufl, S. Kraxberger, G. Lackner, M. Gissing, A. MArsalek, J. LEibetseder, O. Prevenhuber. Android Security Permissions – Can We Trust Tehm?. In: A.U. Schmidt, et al. (eds.) Proc. MbiSec2011, LNICST 94, Springer (2012).

[8] Aruba Networks Inc., BYOD Adoption Is Growing Despite Security Concerns (May 2012), http://www.arubanetworks.com/news-releases/byod-adoption-is-growing

[9] R. Power, et al., Mobility and Security: Dazzling Opportunities, Profound Challenges. Technical Report. McAfee (2011)

[10] Trusted Computing Group, “Trusted Network Connect TNC Architecture for Interoperability”, TCG Specification 1.4 Revision 4, 18 May, 2009.

[11] Global Platform Device Technology, “TEE System Architecture”, Version 1.0, December 2011.

[12] Trusted Computing Group, “TCG Mobile Abstraction Layer, TCG Specification Version 1.0 revision 2.03”, April 28, 2011.

[13] DuVarney, Daniel C., Bhatkar, Sandeep, and Venkatakrishnan, V.N. 2003. “SELF: a Transparent Security Extension for ELF Binaries”., Proceedings of the 2003 workshop on New Security Paradigms. Ascona, Switzerland, August 18 - 21, 2003.

[14] A. U. Schmidt, A. Leicher,, I. Cha, 2010, Scaling Concepts Between Trust and Enforcement, Z. Yan (Ed.), Trust Modeling and Management in Digital Environments: From Social Concept to System Development, IGI Global, Hershey, PA, USA, 20-57.

[15] Cha, Inhyok, Shah, Yogendra, Schmidt, Andreas U., Leicher, Andreas, and Meyerstein, Mike. Trust in M2M Communications: Addressing New Security Threats. IEEE Vehicular Technology Magazine.(Sept. 2009) Vol. 4 Issue 3, 69-75

[16] A. U. Schmidt, A. Leicher, I. Cha, Shah, Yogendra, 2010, Trusted Platform Validation and Management, In International Journal of Dependable and Trustworthy Information Systems (IJDTIS).

[17] A. Leicher, A. U. Schmidt, Y. Shah, I. Cha, D. Howry. Interactive Remote Validation and Management of Trusted Platforms . In: F. Hartung, T. Kalker, S. Lian (eds.) Digital Rights Management: Technology, Standards and Applications. CRC Publishers. Forthcoming.

[18] A. U. Schmidt, A. Leicher, A. Brett, Y. Shah, I. Cha. Tree-formed verification data for trusted platforms, Computers & Security (2012), http://dx.doi.org/10.1016/j.cose.2012.09.004