SECURE TECHNOLOGY ALLIANCE

A SECURE TECHNOLOGY ALLIANCE MOBILE COUNCIL WHITE PAPER

Trusted Execution Environment (TEE) 101: A Primer

Version 1.0

Publication Date: April 2018

Secure Technology Alliance
191 Clarksville Road
Princeton Junction, NJ 08550

www.securetechnologyalliance.org

About the Secure Technology Alliance

The Secure Technology Alliance is a not-for-profit, multi-industry association working to stimulate the understanding, adoption and widespread application of secure solutions, including smart cards, embedded chip technology, and related hardware and software across a variety of markets including authentication, commerce and Internet of Things (IoT).

The Secure Technology Alliance, formerly known as the Smart Card Alliance, invests heavily in education on the appropriate uses of secure technologies to enable privacy and data protection. The Secure Technology Alliance delivers on its mission through training, research, publications, industry outreach and open forums for end users and industry stakeholders in payments, mobile, healthcare, identity and access, transportation, and the IoT in the U.S. and Latin America.

For additional information, please visit www.securetechalliance.org.

The Secure Technology Alliance has used best efforts to ensure, but cannot guarantee, that the information described in this report is accurate as of the publication date. The Secure Technology Alliance disclaims all warranties as to the accuracy, completeness or adequacy of information in this report. This white paper does not endorse any specific product or service. Product or service references are provided to illustrate the points being made.

Table of Contents

1 TEE Overview

The Trusted Execution Environment (TEE) is designed to allow mobile and other connected devices to meet their unique requirements for speed and security. The expansion of the Internet, mobile computing, and the proliferation of connected devices have led to increased opportunities for data and identity theft. In the mobile ecosystem, the number of mobile applications is growing exponentially, and mobile devices can access services without explicit user intervention, which means the device may be sending sensitive data to an untrusted third-party without proper protection or authorization. In addition, users access Internet resources using untrusted mobile applications and browsers, increasing the probability of propagating malware to their devices. And while the widespread availability of WiFi is convenient for users, it opens the door to unfettered attacks on mobile devices and the unauthorized collection of sensitive data.

Mobile computing is currently so pervasive that besides storage of personal data, personal financial applications and social media activities, corporate applications often coexist on the same device. The device can also serve as an online identity tool and as an additional factor of authentication enabling access to highly sensitive domains and resources. Malicious software can invade a mobile device as a result of user activities that originate from an approved device, but the potential for damage increases significantly with practices such as rooting, jailbreaking, and side loading untrusted applications. Avoiding or delaying device security updates can also make a device an easy target for vulnerabilities.

Various instances of mobile operating system (OS) and platform security features make managing device application security even more complicated for application providers. Security breaches can result from viruses, malware, and ransomware. Such breaches can result in financial losses and damage to the reputations of individuals and corporations alike; and for corporations, the costs can eventually outweigh the benefits of doing business.

Neither the services offered by device operating systems nor traditional software protection techniques are necessarily robust enough to combat current security vulnerabilities. The security offered by solutions such as the hardware secure element (SE) also has drawbacks. While the SE offers excellent protection for sensitive code and data, emerging use cases such as mobile payment solutions, content protection, credentials management, and corporate applications for consumer devices require significant amounts of memory space and speed as well as secure access to peripherals. In addition, the cost and complexity of SE implementation is significantly higher than the rich OS that comes with the mobile device.

These challenges encourage exploration of alternate forms of device security for fast, cost-efficient, and convenient solutions. This white paper describes the Trusted Execution Environment (TEE) as a candidate for a mobile security solution that supports a wide range of use cases, such as payment apps, content protection, corporate applications, and loyalty. While various TEE models are emerging, this white paper focuses on GlobalPlatform-based TEE models for mobile devices, which combine the power of hardware with a software-based solution.

2 TEE Evolution

Since the mid-2000s, TEE implementation has evolved from proprietary solutions to a standards-based approach and from mobile devices to a wide variety of Internet-connected devices. This section traces the development of TEE solutions and standards and introduces its potential future applications.

In 2004, Trusted Logic and Texas Instruments pioneered a generic trusted environment. In 2006, ARM developed TrustZone, and the Trusted Logic software became the TrustZone software, licensed by ARM (which then became Trusted Foundations) and commercialized by Trusted Logic.

In 2006, the Open Mobile Terminal Platform (now held within GSMA) published the first set of requirements for a trusted environment, OMTP TR0. In 2008, the requirements were revised to define security requirements for mobile devices. OMTP TR1, also released in 2008, defined a TEE built on top of the TR0 security requirements. In 2010, Giesecke & Devrient (G&D) created their own TEE software, called Mobicore.

In 2012, two major events made TEE a more common standard:

- ARM, Gemalto, and G&D formed Trustonic to boost TEE usage through an open TEE.
- GlobalPlatform and the Trusted Computer Group (TCG) founded a joint working group focusing on TEE specifications and TEE use with the Trusted Platform Module (TPM).

Since 2010, GlobalPlatform has been responsible for driving TEE standardization on behalf of the industry.[1] GlobalPlatform has published numerous TEE-related specifications and offers TEE functional and security certification programs to provide assurances to application and software developers and hardware manufacturers that a TEE product will perform in line with the GlobalPlatform specifications and as intended.

The first major business case for use of a TEE surfaced in 2011, for Netflix: protection of high-definition premium content on smartphones and tablets with a secure digital rights management implementation. Content owners (such as the movie studios) required hardware security before allowing a service provider to display high resolution content on an Android mobile device. Only the TEE could satisfy all of the requirements of this business case, particularly the following:

- Extremely high computing power (to download, decrypt, and display streaming content in real time)
- Hardware-independent content decryption and processing
- Hardware-independent content display (through privileged and secure access to the device output)
- Hardware-protected secure storage of sensitive data (e.g., decryption keys and license files)
- Enforced separation of applications (data cannot be copied or intercepted by other applications)
- Standardized APIs (application portability)

The TEE has subsequently been used for consumer, financial, enterprise, media, government, and Internet of Things (IoT) applications.

[1] https://www.globalplatform.org/specificationsdevice.asp

2.1 TEE Beyond Mobile

Initial deployments of the GlobalPlatform TEE were intended for the mobile space, addressing such applications as secure video paths or secure payments. However, over the past few years, the need for a similar capability has been demonstrated by other applications, such as consumer electronics, smart home appliances, residential gateways, and drones.

Many devices traditionally did not require any security because they were “closed” platforms. However, a number of these devices are now connected to the Internet and support third-party downloadable software, raising legitimate concerns as to whether the devices are secure and trusted. As Internet connectivity moves beyond the PC and mobile phone, security is now a primary requirement for many applications.

2.2 Multitrust TEE

The IoT is reshaping the connected landscape. Mobile phones and tablets are no longer the dominant Internet-connected devices. The other newly-connected devices vary greatly in processing power, amount of memory, and communication speed. The more robust of the newly connected devices (for example, digital video recorders) may provide multiple TEE environments.

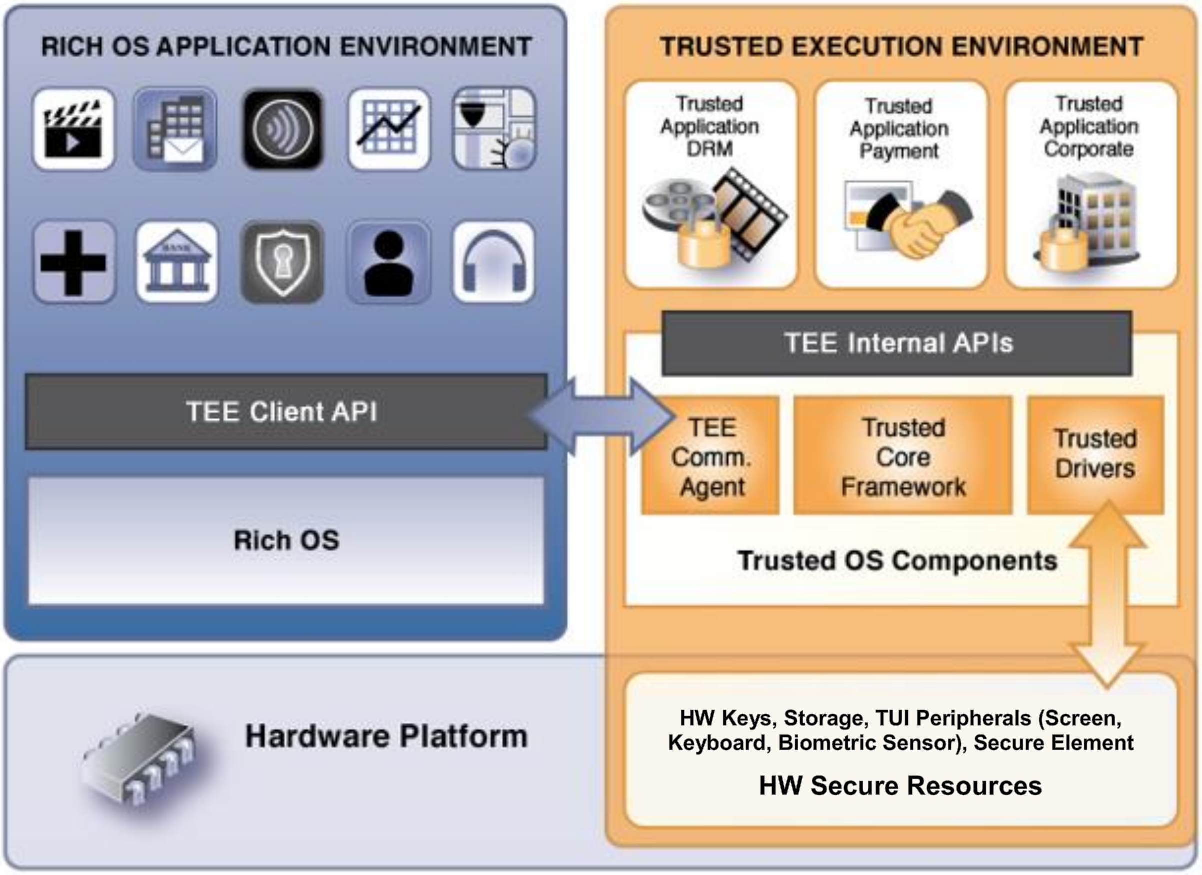
A multitrust TEE enables multiple TEEs to co-exist on a single system. Each TEE can be dedicated to specific services or applications; each trusted application (TA) or suite of TAs can claim its own trusted environment. A multitrust TEE also allows TEEs to be started and stopped dynamically, as needs dictate. Additionally, each TEE can set its individual management policies and life-cycle states to better meet the separate needs of each service’s ownership model.

3 TEE High-Level Architecture and Security

The fundamental principle of the architecture of a device using a TEE is hardware isolation between the TEE and the mobile device's operating environment. This section provides an overview of a possible TEE architecture along the security principles outlined in Section 3.3.

3.1 TEE High-Level Architecture

Figure 1 illustrates a simplified architecture representation of the TEE. As the figure shows, two environments are involved: the rich OS application environment (also called the rich execution environment, or REE) and the TEE. GlobalPlatform specifications require that a TEE be separated from the REE by a hardware-based system. This separation enables cost-effective hardware-based security, since there is no requirement to integrate an extra hardware component into the device to deliver unique, strong security features.


RICH OS APPLICATION ENVIRONMENT
TRUSTED EXECUTION ENVIRONMENT
Trusted Application DRM
Trusted Application Payment
Trusted Application Corporate
TEE Internal APIs
TEE Client API
TEE Comm. Agent
Trusted Core Framework
Trusted Drivers
Trusted OS Components
Rich OS
Hardware Platform
HW Keys, Storage, TUI Peripherals (Screen, Keyboard, Biometric Sensor), Secure Element
HW Secure Resources

Source: GlobalPlatform Inc., *The Trusted Execution Environment: Delivering Enhanced Security at a Lower Cost to the Mobile Market,* June 2015.

Figure 1. Architecture of the TEE

A TEE can run multiple applications, called trusted applications (TAs). Apps in the REE send commands and requests to the TAs through a TEE client API, which connects through a hardware system to a TEE communications agent (see the horizontal arrow in Figure 1). How the hardware connection between the two communications agents is implemented is left up to the TEE provider. The TEE communications agent then forwards these commands and requests to the TAs through the TEE internal APIs.

The trusted OS in the TEE can connect to touchscreens, keyboards, cameras, secured storage, SEs, and other peripherals through trusted drivers (see the vertical arrow in Figure 1). The peripheral services are then available to the TA. Two types of peripherals can be used:

- Peripherals that are accessible only to the TEE (e.g., secure storage and biometric sensors)
- Peripherals that are shared with the rich OS (e.g., screens and keyboards)

When a peripheral is shared, the TEE locks it whenever a TA wants to use it. All communications to and from the shared peripheral are therefore secure and confidential to the TEE.

The GlobalPlatform specifications require the TEE implementation to be separated from the REE by hardware platform protections. A TEE provider can run the TEE implementation on the device's main hardware platform, using the same processor and memory for both the REE and TEE systems (e.g., TrustZone). As an alternative, a TEE provider can use a separate processor and separate resources.

GlobalPlatform also requires that the TEE boot process start before the REE boot process. The boot process loads the security keys from a root of trust. However, GlobalPlatform does not specify how to initiate the boot process. The example in Figure 1 shows the TEE extending into the hardware platform. A hardware platform boots from a first-level bootloader that is read-only code, and the TEE usually starts its boot process from subsequent bootloader software (a secure boot chain). The TEE may also boot from and run on its own processor in the hardware platform.

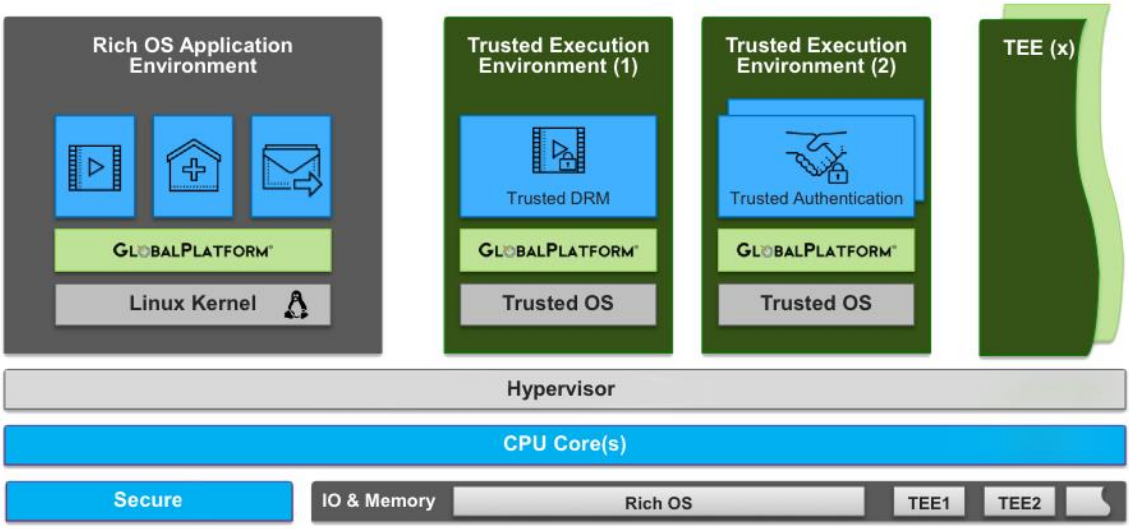
3.2 Multitrust TEE

The multitrust TEE is not currently part of any mobile or tablet system. While there is nothing stopping a vendor from implementing a multitrust TEE on such devices, the main purpose of the multitrust TEE is to separate services and applications into multiple TEE environments so that each can be managed independently of the others. Multitrust TEE systems are therefore used in multi-user and multi-player systems.

A multitrust TEE can be built on a processor with hardware virtualization capability. Hardware virtualization provides a foundation for creating "security by separation," preventing cross-contamination and leaks. In addition, this technology enables the separation and protection of critical assets such as device communication interfaces (and software stacks), as well as storage and other resources.

The TEE environment settings enable multiple configurations consisting of security access requirements, service management, hardware interface connection, and application execution control. A TEE owner determines these settings. The multitrust TEE facilitates different environment configurations to meet the specific needs of each service provider. An example is a video streaming service having exclusive access to a display system and the account management service restricting access.

Figure 2 illustrates one way to implement a multitrust TEE. The figure shows a GlobalPlatform-compliant TEE for devices with a RISC-based CPU. The TEE uses the CPU's virtualization technology to enforce hardware-level isolation and enable operation of a single TEE as well as a multitrust TEE.



Source: MIPS

Figure 2. Multi-Trust TEE Example[2]

In a system with multiple domains, the hypervisor enables isolation of multiple concurrent guest systems (domains). The combination of the hypervisor and the TEE allows full operation of the TEE and the rich environments isolated in each domain. Each domain is referenced with a unique ID (UID) that is managed by the hypervisor; exchanges such as the GP API between the REE and TEE are based on the UID. Each TEE memory and respective resources are sandboxed to provide the isolation for the rich and other trusted environments (trusted OS). The hypervisor maintains an isolated communication exchange between the rich environment and the corresponding TEE.

3.3 TEE Security Principles

A TEE must adhere to certain basic security principles:

- Be part of the device secure boot chain (based on a root of trust) and verify code integrity during each device boot
- Provide hardware-based isolation from the device's rich OS environment to execute sensitive code
- Isolate TAs from each other
- Provide secure data storage, using a hardware-unique key accessible only by the TEE operating system to prevent unauthorized access and modification and any possibility of exploiting the data in other devices
- Provide privileged and secure access to peripherals

Device peripherals (such as fingerprint sensors, displays, touchpads) can be hardware-isolated from the rich OS environment and controlled only by the TEE during specific actions. Access is from inside the TEE, with no visibility or access from the rich OS environment, so that malware running within the rich OS cannot access those peripherals.

[2] The "Secure" box in the figure references the secure element within the system on a chip (SoC)/platform.

Some TEE implementations that have already been deployed also support remote device identity verification and secure management of TAs. These implementations offer service providers the flexibility and scalability needed to deploy and manage the lifecycles of their mobile solutions independently from the chipset and device manufacturers.

4 Role of the Rich OS, SE, and TEE

Many devices today run a variety of applications. A mobile device, for example, may run applications such as e-mail, weather, maps, banking, and payment wallets. Some of these applications may involve sensitive data that must be protected while on the device or in transit to the backend processing systems.

An application provider determines how much protection data requires while on the device or during transmission as part of a risk management assessment. Some applications, such as viewing the weather, may not involve any sensitive data. Others, such as a payment wallet, involve sensitive information such as payment credentials and authentication information, and the application provider will require some level of security for storing, processing, or transmitting the data.

This section describes the options available for protecting data, to enable application providers to determine what approach best addresses the security needs identified in their risk management appraisals. For devices that only support a single application (such as wearables or IoT devices such as appliances), the information in this section can be used to determine what security components, if any, should be included in the device.

4.1 Rich OS

A device’s OS provides both the general environment in which apps operate and the resources available for their use. This rich OS provides the functionality to address multiple use cases but was not designed with security in mind and is vulnerable. One only needs to look at how often the OS on a PC or mobile device is updated or patched to address security vulnerabilities to realize that the OS does not offer a secure environment for sensitive applications.

Application providers use the functionality offered by the OS to produce the best product for their intended customers. However, like the data residing on our personal computers, the information generated, stored, or used by an app is exposed unless special precautions are taken as part of implementation. For some apps this may not be a concern, but apps working with confidential data may seek ways to protect sensitive information. In some cases, the app may choose to store sensitive information in the backend processing system or on the cloud, rather than on the device itself; however, tokens or keys stored in the app may still be required for access to the information. Active management of this reference information can mitigate, but not prevent, potential compromise and exposure of the sensitive information.

Therefore, while the features offered by the rich OS are attractive to application developers and offer great flexibility, developers may need to explore other options to protect confidential data and minimize administrative controls.

4.2 Secure Element

Some devices contain an SE—a tamper-resistant chip capable of securely hosting an application and any associated confidential and cryptographic data (e.g., for key management). An example of a secure element is the chip in a payment card in which the EMV application and data are stored. The benefit of the hardware-based SE is that it is purpose-specific, with functionality that can undergo standard security evaluations. Such components can therefore provide not only a repository for confidential data but also a secure processing container for applications or a set of operations that work with sensitive data. In addition, any cryptographic keys or certificates that may be needed for these operations can also be provisioned securely and used within the component without the need to expose them to an OS.

In devices where the SE is available, apps can control risk by using the SE to perform operations that involve sensitive data and keys and exposing only the result of the operations to the OS.

However, while an SE offers a secure data repository and application host, it has memory space and processing limitations. The SE chip would not be a practical environment for a data-rich application with graphical displays due to its size limitations. In addition, not all devices include an SE.

4.3 TEE

When more security than what an OS can offer is desired, and either the device does not include an SE or the SE cannot meet the requirements of the hosted application, an alternative approach is necessary. In this case, the TEE offers a flexible, secure solution.

A TEE can deliver a trusted environment with hardware-based security. Like the SE, a TEE is a purpose-specific security element that can undergo security evaluations and deliver assurance for applications or data. The TEE can be used on its own or conjunction with an SE on the device, not only to secure direct access to hardware resources (such as user interfaces for input/output) but also to enable secure communication with the SE itself. The TEE can also run multiple trusted applications isolated from each other.

4.4 Summary

Table 1 summarizes the different approaches to implementing apps involving sensitive data.

Table 1. Summary of Suggested Approaches

Property	SE	TEE	OS with Software Protection
Level of code and data protection	Best Tamper-resistant	Better Hardware secured	Good
Memory and computation performance	Limited	Maximum	Maximum*
Executed on the main processor	No	Yes**	Yes
Secure peripheral access	No	Yes	No
Provides device attestation	Limited	Yes	No
Software ecosystem	Limited	Yes	Yes
Use case support	Limited	Unlimited	Unlimited

*Some software protection mechanisms impact performance.
**May run on a separate processor.
Source: Trustonic.

5 TEE Use Cases and Implementation Examples

This section describes the following use cases and example implementations for a TEE:

- Mobile payments
- Mobile identity credentials
- IoT
- Content protection

5.1 Mobile Payments Use Case

As mobile payments become more convenient, security requirements must be augmented to thwart malware attacks. For consumers, mobile payments can take place: at a merchant's point of sale (POS) using a number of communications technologies (e.g., Near Field Communication (NFC), Magnetic Secure Transmission (MST), QR code, Bluetooth); through a peer-peer app; through a merchant's app (i.e., in-app); or through the mobile browser.

For consumers, use of payments through mobile is getting more prevalent. It can be through NFC, QR code, in-app and In-browser payments. For all these scenarios, payment credentials are being stored and transmitted from one entity to the other. The payment application resides on the same hardware and OS layers as that of other applications, which can include a malicious application and thus thwart the security of the stored payment credentials. This is where TEE can help and provide a better way of securing such credentials at rest and during presentment.

For merchants, mobile point-of-sale (mPOS) solutions enable merchants to accept payments using mobile devices such as smartphones, tablets, or proprietary wireless acceptance devices. mPOS devices not only transmit the payment transactions, they can also enable signatures and PIN entry. Capturing payment data and sending the data wirelessly requires security above and beyond traditional OS-based support. TEE can provide a platform in mPOS payment acceptance devices for a trusted user interface and data handling, using TAs that are generated specifically for each payment brand. End-to-end encryption of the data transmission can be achieved using hardware protection. Some of the mPOS platforms allow users to develop custom applications. Value-added services such as coupons, offers, and loyalty can be enabled and targeted to consumers securely, creating a richer user experience without relying on paper or plastic.

5.1.1 Implementation

This implementation example considers a consumer mobile payments wallet use case, which involves storage of credentials, user authentication, transaction management, and application security. Credentials for making proximity and online payments are stored in the device. During the transaction, the application retrieves the credentials to enable payment and stores the transaction details. Credentials usually involve tokens or payment account numbers and cryptographic keys to generate cryptograms. The implementation steps include provisioning, lifecycle management, and transaction management.

Provisioning is the process of storing payment credentials and cryptographic keys in a secure storage location such as an SE or TEE. A provisioning system must also transport the keys and credentials securely. In addition, provisioning requires secure communication between backend systems and the device. Both end-to-end encryption and secure credentials storage can be achieved by a TA with privileged access to a TEE.

Lifecycle management is the process by which credentials are managed after they are provisioned to the device. Lifecycle management can include suspending, resuming, and deleting credentials, and refreshing temporary credentials when credentials cannot be stored for the life of the payment instrument. TEE offers a secure storage mechanism by encrypting the credentials using hardware-enabled keys.

During the transaction process, mobile apps need to authenticate the user and payment credentials, create payment network-specific data elements, compute cryptograms, and send the data to the payment point of interaction (POI) for further processing. This process requires validating credentials retrieved from secure storage, preventing malware attacks, and storing certain transaction details. During transactions, users may have to authenticate themselves using mobile PINs or biometric identification. Using TEE's trusted user interface APIs, user authentication can be enabled securely, preventing malware from obtaining user credentials.

5.1.2 Challenges

TEEs have faced challenges in mobile payments implementations.

Unlike SEs which have a standard process for certification, TEE certification processes are still being formalized. In addition, EMV-based contactless mobile payment solutions were designed originally to use the SE and are now being adapted to the TEE.

However, a more recent mobile approach combines host card emulation (HCE), which allows an application residing in the application processor to access the NFC functionality rather than requiring an SE, and tokenization (substituting a token for the payment card number). This latter approach is well-suited for a TEE.

Software protection requires credentials stored in the device to be refreshed periodically based on dynamic parameters, whereas TEE allows for static key storage, a better alternative. Furthermore, using a TEE provides unique security features not available when using an SE: biometrics and a trusted user interface, enabling secure user authentication and secure confirmation by the user.

In addition, developers may face challenges as implementation of the TEE may differ based on device, OS and implementation approach.

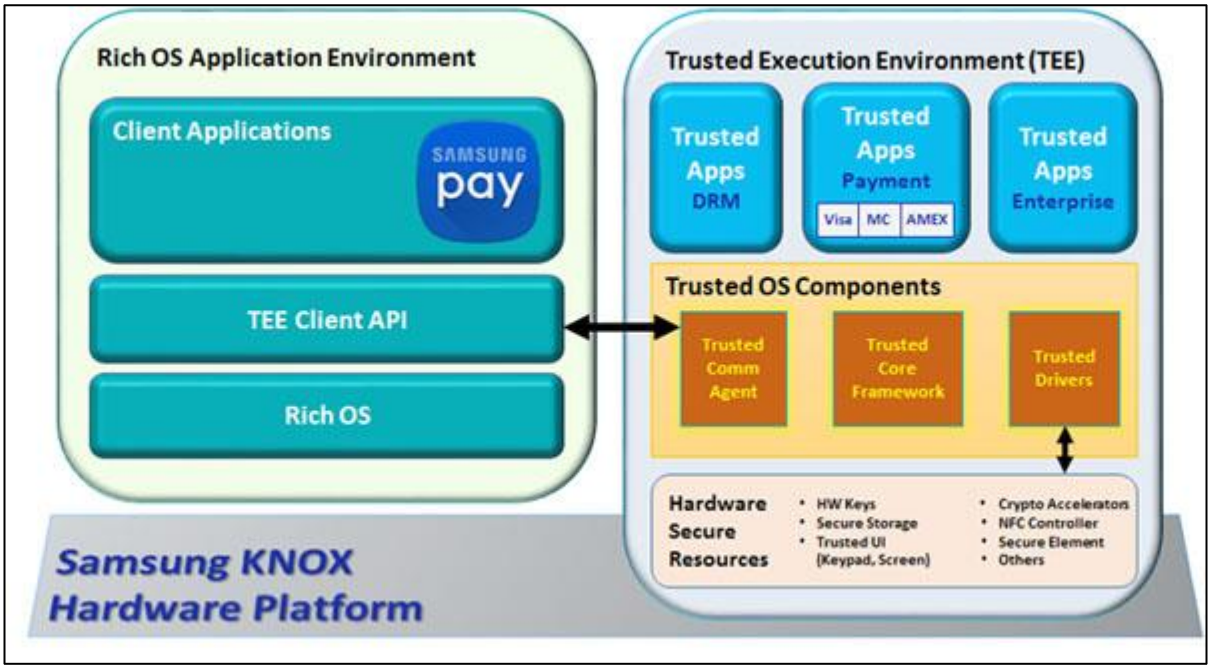
5.1.3 Examples

Samsung mobile devices use a TEE implementation to support Samsung Pay.

Leveraging the Samsung Knox security platform, Samsung Pay's TEE implementation employs TrustZone-based hardware isolation to segregate secure storage and data processing from non-secure operations. Trusted boot and a trusted OS further secure the wallet app's TEE (Figure 3).

As shown in Figure 3, TAs run inside the TEE. They are responsible for user authentication, data encryption keys and key management, and communication with the payment networks. The trusted drivers inside the TEE control access to the device's biometric sensors, trusted user interface, and PIN pad, as well as to the NFC controller and an MST antenna that securely transmits payment tokens and cryptograms to the merchant's POS system. The cryptogram is computed only after successful user authentication and only once per authentication.

Whenever a trusted app is loaded into memory, the TEE performs cryptographic verification of the binary (the app's executable program) to further ensure that only authentic Samsung Pay TAs are executed and allowed to access payment credentials. During the device boot process, the bootloaders, the TEE, and the hardened Android kernel are verified through code signing. The Knox framework supports other defense-in-depth measures, ensuring comprehensive app, OS, and device integrity.



Source: http://developer.samsung.com/tech-insights/pay/device-side-security

Figure 3. Samsung Pay with Samsung Knox and TEE

5.2 Mobile Identity Use Case

Mobile identity can take a variety of forms. For financial institutions, a user's identity may be credentials stored on the phone and associated with a PIN or fingerprint. Government entities may accept derived credentials stored on mobile devices to authenticate employees; citizens may be authenticated using a mobile passport or driver's license. For corporations, a login-password combination, a one-time password, or biometrics may be required to access corporate applications. The mobile phone can also be used to access a building or unlock or control a car.

In this use case, mobile identity refers to a solution that handles user credentials. As more and more operations occur in a mobile environment, it also becomes critical to authenticate the device from which the user is authenticating to prevent fraud.

5.2.1 Implementation

Mobile identity solutions usually rely on the following security principles:

- Secure lifecycle management of the application and credentials
- A secure communication channel between the device and the remote entity (e.g., a building, a car, a server)
- Secure user credential storage and authentication on the device, such as certificates, a one-time password, a PIN, or biometrics
- Trusted device authentication

The TEE protects the application during lifecycle management and execution on the device and also protects user credentials, with both hardware-backed storage and hardware isolation of credential processing. The TEE secures the user interaction with the device, supporting fingerprint recognition, PIN entry, one-time password display, or similar authentication mechanisms.

The TEE also provides and protects a unique device identity (that can, for example, be associated with a user or a group of users). It also can secure communications with the remote entities (for example, by protecting the credentials used for mutual authentication).

5.2.2 Challenges

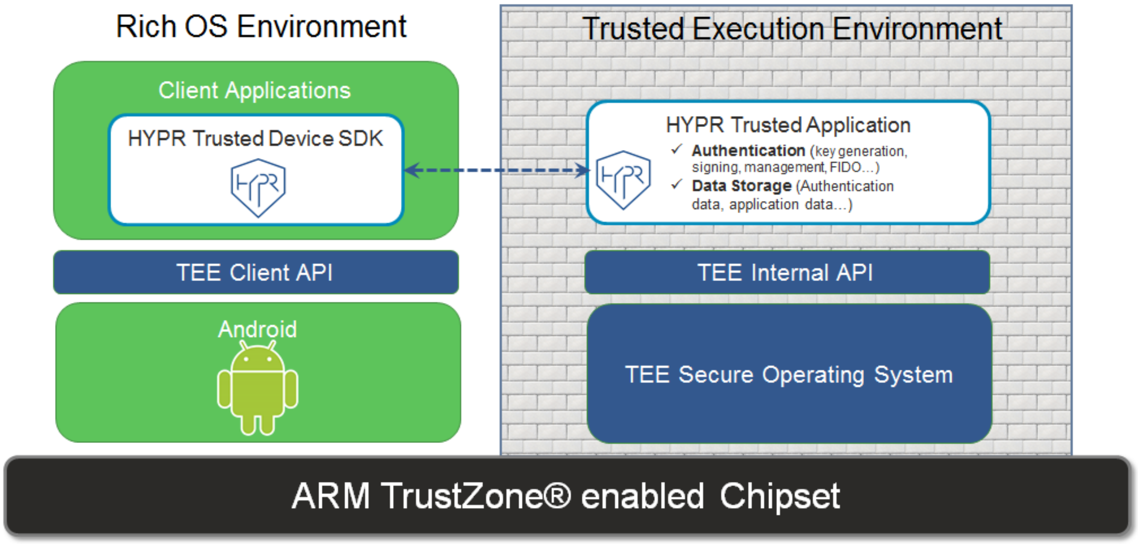
Some markets demanding high security (such as the government) may require the use of an SE to store certain credentials. Use of the TEE can complement use of an SE by securing user interaction and communication with the SE.

It is also important for hardware manufacturers to allow third parties to securely access peripherals via the TEE. Not all devices currently do, creating some challenges for developers that need to support secure user interaction across all devices.

5.2.3 Examples

One example is HYPR Corp's[3] development of the HYPR Mobile Client leveraging the Trusted Device Software Development Kit (SDK). This client provides decentralized biometric user authentication on the device using advanced device-level security including the TEE for enhanced protection. Some of the key features of this SDK include secure data storage, ECC/RSA 2048-bit encryption, and biometrics verification.

The SDK is used in solutions for various markets including financial services, insurance, healthcare and automotive.


Rich OS Environment
Trusted Execution Environment
Client Applications
HYPR Trusted Device SDK
HYPR Trusted Application
✓ Authentication (key generation, signing, management, FIDO…)
✓ Data Storage (Authentication data, application data…)
TEE Client API
TEE Internal API
Android
TEE Secure Operating System
ARM TrustZone® enabled Chipset

Source: HYPR Corp.

Figure 4. High-Level Architecture Example of HYPR Trusted Device SDK on Android

[3] https://www.hypr.com/

5.3 IoT Use Case

The TEE and the concept of hardware isolation are applicable to devices that are part of the IoT. More and more devices are connected (Figure 5) and aggregate, share, or process sensitive data, making it crucial to protect the integrity and origin of the data.

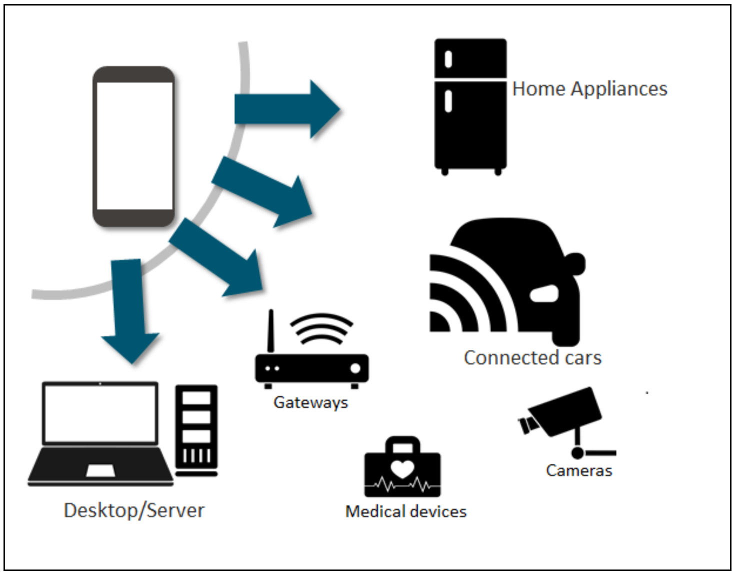


Figure 5. Landscape of the Internet of Things

IoT implementations today include:

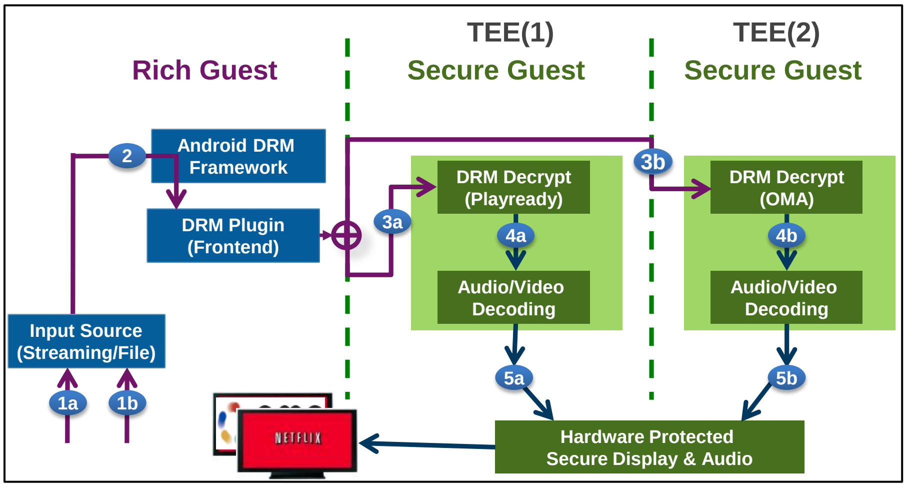
- Smart cities—public safety, energy, transportation
- Smart homes—surveillance, energy management, smart locks
- Industrial—production streamlining, inventory management, robotics, building access
- Automotive —driverless cars, telematics, infotainment, in-car payment
- Healthcare—telehealth, remote drug prescription and administration

Use of the TEE can enable secure solutions in the following areas:

- Software/firmware management
- User and device enrollment
- Data analytics
- Data transmission
- Device to device communication
- Device to cloud communication
- Device authentication
- Device counterfeit protection
- Device tracking
- Payment
- User authentication

One use of a multitrust TEE is to implement multiple secure data paths for different tasks. Many host applications are supported by a number of concurrently running, dissimilar security tasks, each of which needs to be autonomous. For example, the control path for a secure video flow must be kept isolated from an adjacent payment data flow. Each data flow must conform to specific certifications, involves different lifecycle operations, and is subject to remote management from different third parties. Comingling the data could compromise and contaminate the assets that belong to each task.

A multitrust TEE supports multiple concurrent TAs, which are isolated from each other and operate in their own trusted environments (Figure 6). Multiple TEEs also support independent management and maintenance tasks, such as patching software, updating, and managing lifecycle events.



Source: MIPS

Figure 6. Multitrust TEE

5.4 Content Protection Use Case

Securing premium content was the initial major business case for the TEE, and some major digital right management (DRM) solutions are implemented in the TEE (e.g., Google Widevine).[4] DRM implementations offer a variety of ways to protect content and detect fraud (e.g., licensing, watermarking). This section describes how DRM can be used to protect premium video content.

Premium content can refer to high definition content and also to the time between which a movie is released in theatres and when it is distributed by a content provider (e.g., Netflix), also called early-window content. The shorter the time, the more valuable the content and the more it needs to be secured. Hardware security becomes mandatory.

5.4.1 Implementation

Securing premium content requires the following:

- Protect the DRM code and the credentials (ensuring that the keys and algorithm cannot be duplicated by third parties)
- Securely send the content (encrypted) from the server to the device (establishing an end-to-end tunnel that third parties cannot intercept or capture)
- Securely apply the licensing rights in the device (verifying that a certain user or device can watch or read the content)

[4] For more information, see https://storage.googleapis.com/wvdocs/Widevine_DRM_Architecture_Overview.pdf.

- Securely decrypt the content (protecting the content after it is decrypted)
- Securely display the content (ensuring that the content cannot be captured by a third party such as a screen grabber or scraper)

The TEE can secure premium content as follows:

- Has enough computing power to receive, decrypt, and send content securely to the hardware decoder/renderer without impacting the user experience
- Can host, execute, and manage the DRM TA securely
- Enables personalization of the TA with unique credentials
- Can protect the licensing scheme (e.g., anti-rollback) and cryptographic keys used to decrypt the content
- Enables secure content display independent of the rich OS (privileged access to the hardware decoder/renderer), preventing malware or users from stealing the content when decrypted

5.4.2 Challenges

One challenge is how to enable the secure display of content for DRM trusted applications installed remotely. As of today, to benefit from a TEE-enabled secure display of video content, the content providers or DRM solution providers would have to work directly with the device manufacturers and have their solution pre-installed within the device. Device manufacturers should ensure that secure video display is available for remotely installed applications.

6 TEE Challenges

Some TEE implementations face certain challenges to enable a vibrant third-party ecosystem, including: enabling remote access; isolating applications from different solution providers; enabling secure peripheral access to a wide range of devices.

Like the chip on a payment card, the TEE is embedded in a device, which means it has to be integrated within the device before the device is shipped. Unlike payment cards, which are typically delivered to customers with applications and data already resident on the chip, devices with embedded TEEs are usually consumer-owned devices on which applications and data must be installed remotely. The process used to load data securely therefore has unique requirements. A unique device identity (root of trust) must be embedded to enable remote attestation that the device is genuine before sensitive assets, such as application data, can be installed in the TEE, and to enable secure data transfers and remote management of TAs. TEE providers have to work directly with each of the chipset manufacturers and device makers to make this happen. The process is not very different from the process for restricting and managing access to the chips on payment cards but must be more secure to compensate for the distributed environment and number of parties that may be involved.

The TEE can support multiple applications with complete separation of content and security. Therefore, the provisioning process for applications and their data needs to manage the separation and include permissions for loading the different applications to maintain segregation. GlobalPlatform has defined a TEE Management Framework (TMF)[5] to administer TEE environments and manage TAs and security domains. The TMF supports the concept of multiple security domains,[6] enabling the provisioning and administration of each TA in its own isolated environment.

As described in Section 3.3, the TEE can provide privileged access to peripherals, such as a touchscreen with a trusted user interface. This feature is already available in many Android devices but is neither ubiquitous nor standardized, as each TEE manufacturer provides its own trusted driver. Like provisioning the root of trust, secure privileged access to peripherals requires integration within the device before it is shipped. GlobalPlatform is currently completing a trusted user interface specification standardizing a generic API for the trusted peripherals.

The majority of Android smartphones and tablets support the hardware required for a TEE environment. However, the integration of the secure TEE operating system is left to the device manufacturers. While the GlobalPlatform specifications define the functionality expected of a TEE implementation, not all manufacturers have adopted it yet. On non-Android devices, the TEE is not open to third parties. For example, iOS devices embed a form of TEE, but application developers cannot develop and manage TAs inside of it. On Android devices, if a device that supports a TEE does not allow remote management of a TA, then a third party can only deploy and manage TAs in conjunction with the device maker, which complicates the go-to-market and scalability of the solution. Therefore, developers should consider developing and deploying TEE-based solutions with TEE providers that enable remote management, particularly in the mobile space. Fortunately, a large number of TEE-based mobile devices already support remote TA management.

Currently, development and management of the TAs are hardware/OS specific. In order for TEE applications to become pervasive in the mass market, a common API should be standardized and

[5] *TEE Management Framework Version 1*, GPD_SPE_120, Dec. 2016.

[6] The security domain for a TEE uses the same concepts as the security domain in an SE.

deployed across all TEE implementations. Similar to JavaCard for SEs, a common platform API would provide a cross-platform solution that would allow applications to be written once for all target devices. This would allow a large population of application developers to utilize the technology and remain agnostic to the TEE solution. Thus, there would be no need for multiple implementations of the same application based on the target platform.

7 Mobile Implementation Considerations

By definition, the rich OS in a mobile device cannot be trusted. It can be rooted (jailbroken), cannot be certified, and is too large to be bug free. The role of the TEE is to isolate and protect sensitive data.

7.1 Developing a TEE Solution

A mobile solution leveraging the TEE will be composed of two distinct applications: the rich OS application and the TA. To design and develop a solution leveraging the TEE, an application developer should follow these basic guidelines:

- The solution design should not trust the rich OS. A TA should never send unprotected sensitive data to the rich OS application.
- Sensitive data (such cryptographic processing or sensitive data storage and processing) should be ported to the TEE. The purpose of the TEE is not to host and execute the full rich OS application, so the amount of porting should be minimal.
- The rich OS is a bridge used to communicate securely between a TA and a remote entity (e.g., a server or SE). However, the TA should always be able to have a secured and authenticated data exchange with the remote entity that is implemented on top of the transport layer provided by the rich OS.
- Capture and display of sensitive data should be routed through the TEE.

Application developers should also anticipate devices that do not support the TEE and use software protection for those cases (e.g., whitebox cryptography, code protection techniques). Implementing the fallback to software protection is easy and is already available from some TEE providers.

7.2 Managing a TEE Solution

A TA can be pre-embedded in the rich OS application (e.g., Samsung Pay). The application bundle can be posted on any app store and downloaded by a customer.

TA management requires an application provider and the use of a specific protocol defined in the GlobalPlatform TMF. The GlobalPlatform TMF specifications require the application provider to have the proper security mechanisms to connect to the TEE or to the provider's TA. TMF can use PKI-based security (easy implementation) or the same secure channel protocol used by an SE for more robust security. A TEE will reject connections that do not meet the minimum security requirements.

The application provider can be hosted on premise or leverage cloud-based environments. In any case, a business entity willing to manage TAs should collaborate with the TEE provider to make sure that the minimal security requirements are met. As described in Section 3, the TEE uses security domains to give application providers a virtual presence in the TEE. Security domains enable application providers to manage their TAs directly, without having to go through another party such as the TEE owner or the device manufacturer.

8 Conclusion

Widespread data breaches have made it very important to harden security measures when sensitive data are being processed and stored. This is resulting in an evolution of tokenization and enhanced device security mechanisms in personal computing and connected devices technologies. As in-device payment transactions, mobile identity and authentication, corporate applications and media streaming become more popular, transaction speed and faster time to market are of utmost importance. The wide variety of other connected devices also require security to protect sensitive data. While use of an SE undoubtedly provides a higher form of security, modern devices require more speed, memory, and cost effectiveness than use of an SE can offer. In addition, the SE solution may be too complex for many use cases.

TEE offers a dependable security platform. When built using GlobalPlatform standards, TEE offers scalability and an easily deployable solution for any device that supports the TEE architecture. In the mobile space in particular, the TEE provides a simple and quick-to-market solution across all devices that support the technology to providers willing to develop, deploy, and manage trusted applications independently from the hardware manufacturers. Using the TEE imposes no additional limitations on speed, memory, or computing power. The TEE relies on the device's main application processor and the device's native memory space.

TEE is achieving a foothold in various electronic devices, including personal computing, mobile, and IoT-connected devices. While SEs continue to make sense for certain use cases, TEEs can work independently as well as with SEs to enhance processing abilities. Used alone, TEE is a prime contender for use cases where security is of high importance, such as HCE-based payments, mobile wallets, mobile POS, content protection, and connected devices. And while TEE is maturing, the next generation of mobile security can leverage TEE-based solutions to protect devices from multiple threats and vulnerabilities.

Though there are challenges in creating a scalable TEE eco-system, many of these challenges can be overcome by participation in industry standardization efforts from industry players such as chip manufacturers, device manufacturers and TEE solution providers.

9 Publication Acknowledgements

This white paper was developed by the Secure Technology Alliance Mobile Council to provide an educational resource on the Trusted Execution Environment and relevant use cases.

Publication of this document by the Secure Technology Alliance does not imply the endorsement of any of the member organizations of the Alliance.

The Secure Technology Alliance wishes to thank Council members for their contributions. Participants involved in the development and review of this white paper included: Cubic Transportation Systems; Discover Financial Services; First Data; GlobalPlatform; JPMorgan Chase; Mastercard; MIPS; Thales e-Security; Trustonic; and Visa.

The Secure Technology Alliance thanks **Sridher Swaminathan**, First Data, for leading the project and the following Council members and guests who participated in the project team to write the document:

- **Majid Bemanian**, MIPS
- **Hank Chavers**, GlobalPlatform
- **David deKozan**, Cubic
- **Jose Diaz**, Thales e-Security
- **Florent Joubert**, Trustonic
- **Sadiq Mohammed**, Mastercard
- **Tony Sabetti**, JPMorgan Chase
- **Sridher Swaminathan**, First Data
- **Hans Van Tilberg**, Visa

The Secure Technology Alliance also thanks Council members who participated in the review of the white paper including:

- **Hank Chavers**, GlobalPlatform
- **Florent Joubert**, Trustonic
- **Cathy Medich**, Secure Technology Alliance
- **Cheryl Mish**, Discover Financial Services
- **Sadiq Mohammed**, Mastercard
- **Hitesh Shah**, Discover Financial Services
- **Mike Strock**, Secure Technology Alliance
- **Sridher Swaminathan**, First Data
- **Christopher Tomczak**, Discover Financial Services

Trademark Notice

About the Secure Technology Alliance Mobile Council

The Secure Technology Alliance Mobile Council aims to build industry awareness around the business and security impacts of utilizing different technologies for distributing, storing and using secure credentials on personal mobile and tethered wearable devices. The Council believes raising awareness will facilitate broader discussion on creating standards. The Council will create resources to help implementations and accelerate the adoption of payments, loyalty, marketing, peer-to-peer, identity, and access control applications using mobile and tethered wearable devices. The Council focuses on activities that will help to educate the industry on implementation and security considerations and will act as a bridge between technology development/ specification and the applications that can deliver business benefits to industry stakeholders. Additional information on the Mobile Council can be found at https://www.securetechalliance.org/activities-councils-mobile-council/