Qualcomm Technologies, Inc.

Qualcomm® Trusted Execution Environment (TEE) v5.8 on Qualcomm® Snapdragon™ 865 Security Target Lite

80-NR875-21 Rev. AA

July 20, 2021

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
AA	March 4, 2021	Initial draft from Security Target with confidential information removed in Section 2.4, 2.5, 8.1.4, 8.3 and A.2. Updated guidance in section 2.4.2.1.
AA	May 20, 2021	Section 2.2 was updated with correct hashes for binaries. Section 2.4.3 was updated with correct GP reference versions. Remove generic application notes in Section 7.
AA	May 25, 2021	Changes in Section 7.1.3, 8.4 and 8.5.
AA	June 17, 2021	Updated revision history. Addressed copyright and trademark comments. Minor update in Section 8.2.1 and Section 2.3.2.
AA	June 24, 2021	Update in Section 2.4.2.1
AA	July 13, 2021	Section 2.4.2.1 was updated to add rev number for KBA-191027204619
AA	July 20, 2021	Section 2.3.3 was updated to remove inconsistency for qwes.mbn.

Contents

Figures

Tables

1 Introduction to Security Target

This Security Target is defined for the Qualcomm® Trusted Execution Environment (TEE) version 5.8 on Qualcomm® Snapdragon™ 865. SM8250 is the part number for Snapdragon 865.

1.1 Conventions

The security functional requirements (SFRs) have the following conventions:

- Phrases in **boldface** indicate the an assignment or selection already performed by the Protection profile (PP).
- Phrases in blue or **blue boldface** indicate an assignment or selection performed in this security target.
- The same color convention applies to application notes where additional ones are set in blue.
- Phrases in *italic* indicate a title of a referenced document, specification or article, irrespective of color convention.

1.2 Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at https://createpoint.qti.qualcomm.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

1.3 Acknowledgements

QTI would like to acknowledge Riscure North America for creating the initial draft of this document.

2 Security Target

This chapter describes the Security Target, target of evaluation (TOE), and the key features.

2.1 ST reference

The evaluated version of the Security Target is identified by the document number 80-NR875-18 Rev AA.

2.2 TOE reference

The TOE identifies commercially as Qualcomm® Trusted Execution Environment (TEE) v5.8 on Qualcomm® Snapdragon™ 865.

From the perspective of the integrator, the software parts of the TOE are uniquely identified by the release numbers and the SHA256 hash of the binaries:

- TZ.XF.5.8-00041.3-SM8250AAAAANAZT-3

Binary	Hash
tz.mbn	7ea7ee3137e4b8f89a94143c056d97286bbdd991122fa4c9bfaf637bc85d8cd7

- TZ.APPS.1.8.c1-00005-SM8250AAAAANAZT-1

Binaries	Hash
applib.lib (32bit)	4928b3ba49e6639e060288f929dbd3442647d2cf441dff8068d5b1d7851b4d2e
applib.lib (64bit)	29ccc59d0a88af9b1b7b36a080874a2feba61d9897ff04aef2583eb1583ad11f
common_applib.o (32bit)	da4044c1c0c50aecaa82af64f80b25d28ae470103be0a15de8182241fc3d35f9
common_applib.o (64bit)	33be65f3b1e602d9683e2d86ef5df120ed9859f9e7c7f9fa436c514567bf3d56
eseservice.mbn	d6118143e5fbd1fc500ed3805bb656699a68a569843706a393301e9296b373e1
scp11cry.mbn	4530aaacb38b9baa7ad4b8ec0bccbd61750be3f3c719fbc6cc9e0914c1dcc9fb
tee_se_api.lib (32bit)	bed821bcdf00de4e40919f7b7764c7206ba04f9a3a7ca2bb78d3da3723fa68b5
tee_se_api.lib (64bit)	a3638101eb1533031e695e7364127d175063bcc7db46a47e949bb59e28baa8e5
qwes.mbn	93c65bb73bf8d34416caae3aec003c4c1ee6ac34bcf8570a9a287a8d2bb989e7

- BOOT.XF.3.2-00295-SM8250-1

Binary	Hash
xbl_sec.mbn	1c01e3e8cbfd98290910e2d9adda74dc150bb1825b4af7ae4412008a78237c22

Similarly, the hardware parts of the TOE are uniquely identified by the release identifier, that is, Snapdragon 865.

In addition to this, the TOE comprises the guidance documents detailed in *2.4, TOE description*, which are identified uniquely by their document versions.

2.3 TOE overview

2.3.1 TOE type

The TOE is a trusted execution environment (TEE) which is intended to operate in parallel to a rich execution environment (REE). It allows for executing trusted applications (TA) in a secured manner isolated from any applications running in REE. The TEE is integrated into a system-on-chip (SoC) and utilizes hardware components of the SoC as outlined in more detail in the description of the physical scope of the TOE.

The TOE offers a comprehensive set of services to the TAs including integrity of execution, secure communication with the client applications (CA) running in REE, trusted storage, key management and cryptographic algorithms, time management, and arithmetical application programming interfaces (API).

The TOE is an open environment where TAs can be loaded and installed post-issuance (in the end-user phase of the TEE-enabled device).

The TOE also incorporates the Qualcomm® Hypervisor Execution Environment (HEE) component that seamlessly interacts with the TEE to manage resources shared between the REE and the TEE. The Qualcomm HEE operates with the highest (hypervisor) privileges (EL2) within the REE.

2.3.2 Usage and major security features of TOE

The TOE consists of the TEE software executing on an SoC. The secure execution environment operates in hardware enforced isolation of the REE. The system consists of the following main components:

- A secure boot mechanism in which the hardware-based root of trust (RoT) uses cryptographically strong signature verification over the software components loaded for secure initialization of the TOE hardware, Qualcomm TEE and REE.
- Qualcom TEE v5.8, the secure execution environment and common services layer provided to TAs .
- GlobalPlatform TEE APIs support for trusted applications

Qualcomm TEE v5.8 provides application programming interfaces to the TAs and a communication interface to REE. In addition to this, the system offers debugging interfaces, but the OEM can effectively disable these debugging interfaces in the secure operational configuration of the product.

The TOE provides also functionality to the TAs to establish a *trusted path to an embedded secure element* (eSE). The eSE itself is not part of the TOE.

Finally, the TOE provides the functionality to support *device attestation* of the TEE.

The TOE is intended to be used for scenarios where processing and storage of sensitive data is assured even if the rich execution environment is compromised. Examples are:

- Transaction processing in payment solutions, both online and offline.
- TEE-SIM application in internet of things (IoT).
- Processing of entitlement messages in digital rights management (DRM) and content protection.

2.3.3 Required non-TOE hardware, software, and firmware

The TOE depends on the following non-TOE hardware components for proper functioning:

- External dynamic random-access memory (DRAM, either external or deployed as package-on-package).
- Replay protected memory block (RPMB, to support secure file system (SFS) version and TrustZone applications version anti-rollback).

The implementation of the trusted path to an embedded secure element feature connects to an external secure element which is a non-TOE component, that is, NFC Controller and eSE NXP SN100 / SN110.

The TOE depends on the following non-TOE software components for proper functioning:

The developer delivers the target of evaluation (TOE) in the form of binary images to the OEM. Therefore, the product is technically not deployed with preloaded TAs. However, there are several system extensions implemented, such as other TAs in the user-privilege level, which are mandatory for the proper functioning of the system. The OEM is responsible for the proper loading of these applications before using the system.

- TA identifier – Device configuration (Devcfg.mbn)

  Main developer – QTI

  Role: Supports the proper configuration of the device.

- TA identifier – Keymaster (Keymaster.mbn)

  Main developer – QTI

  Role: Supports the secure boot of the REE using the keymaster feature in Android.

The detailed descriptions of the physical and logical scope of TOE and further details are provided in *2.4, TOE description*.

2.4 TOE description

2.4.1 Physical scope of TOE – Physical components

The following figure shows the external interfaces of the SM8250 SoC.

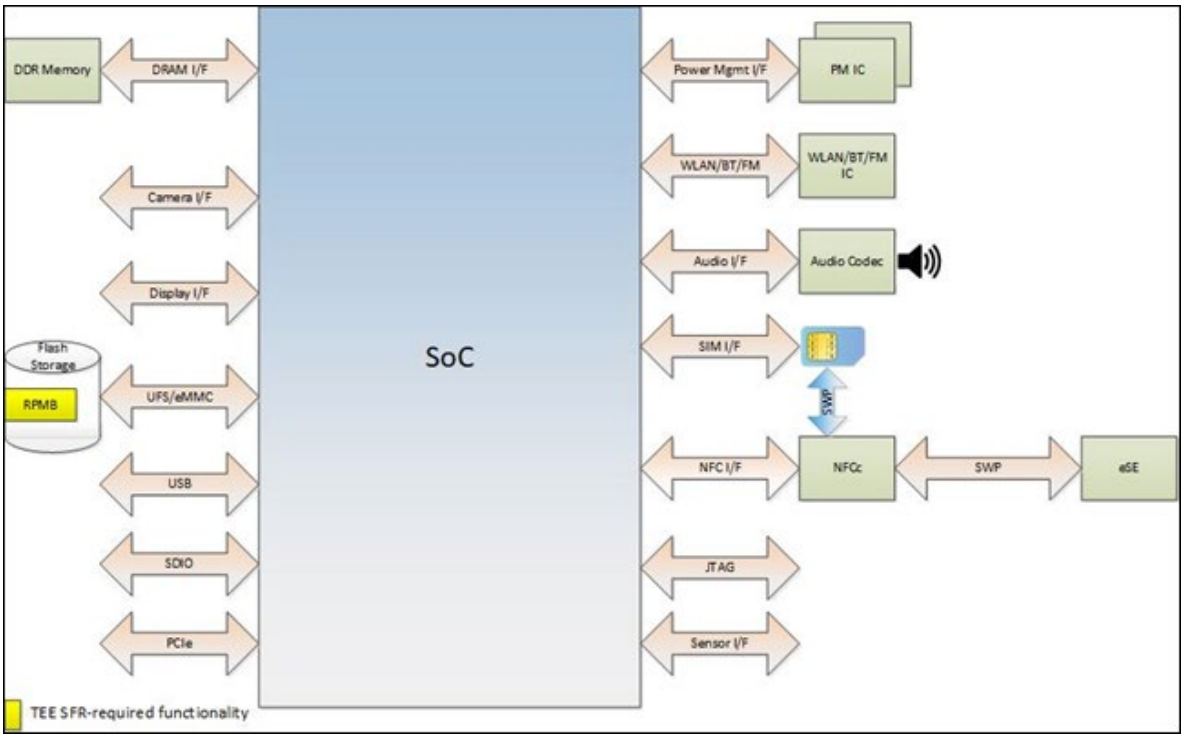

Figure 2-1 External interface for SM8250 SoC

RPMB is the only feature outside the SoC that is directly required by some of the SFR implementation, that is the roll-back protection. This component is cryptographically tied to the SoC by a shared key that is used to authenticate all messages between the SoC and RPMB. The integrator has to choose a component that adheres to the provided specifications and follow the integrator guidance, but the functional contribution of the RPMB is limited so that it can be considered a non-TOE component.

The other peripherals are required so that the system can function correctly, but they are only indirectly required by the security design of the system. For example, the external DRAM memory is used to store sensitive data in encrypted and authenticated form only so that it can safely and transparently be read by Qualcomm TEE because the system can detect any corruption while in storage.

In any case, the proper integration of the SoC with suitable peripherals is under responsibility of the OEM who constructs the final device. Qualcomm TEE hosted on the SM8250 SoC provides many access control mechanisms that are addressed in detail by the SoC evaluation and which can be flexibly configured to meet the requirements of a large set of target device configurations. The evaluation assesses that all of these security configuration options are properly described and communicated to the integrator. However, assessing concrete device configurations is out of the scope of this SoC evaluation.

From the hardware perspective, the SM8250 SoC consists of the components in the light-blue box surrounding most components, except the External DRAM and the UFS (external flash).

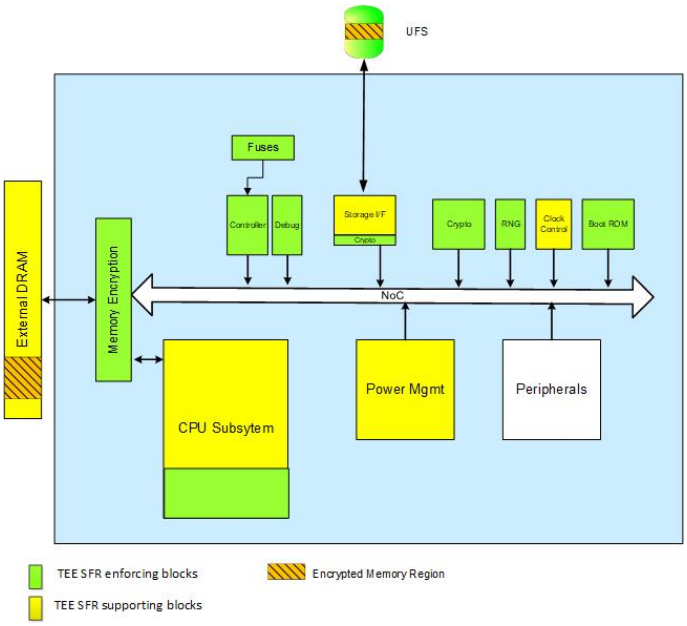

Figure 2-2 TEE hardware block diagram

The following components enforce the security implementation:

- The SoC contains a TrustZone-enabled CPU, offering hardware enforced isolation of memory, register and peripheral access, and controlled switching between secure and non-secure modes.
- The CPU offers central features such as the CPU registers, MMU features, and so on.
- The external memory encryption transparently encrypts and decrypts information written to and read from the external DRAM.

- The one-time programmable security fuses that control the security configuration of the system and the main security anchors of the system
- The hardware cryptography support and a deterministic random number generator that is seeded by a physical random number generator.
- The boot ROM that contains the primary boot loader, which bootstraps the secure loading process.
- The system memory management units (SMMU) that control access to memory, and the hardware access control features that control access to the hardware components.

In addition to this the following components are directly or indirectly required non-TOE components within the SoC:

The storage interface, the external memory controller, the CPU in general, the clock control and the power management all are indirectly required non-TOE components the TOE depends on to ensure correct processing of the security functionality. What is in scope of the TOE are the hardware access controls which ensure the proper firewalling between secure and non-secure operation.

In addition, the SoC contains further components that interface with cameras, displays, WLAN adapter, and other peripherals; these are not in the scope of the core Qualcomm TEE evaluation.

2.4.1.1 Hardware/packaging variants

Qualcomm TEE is capable to operate on two different packaging variants. The hardware architecture description (See *2.3.3, Required non-TOE hardware, software, and firmware*) describes how external DRAM is connected to the SoC using the memory encryption block in the SoC to transparently encrypt all information written to and read from the protected part of the external DRAM.

In a second packaging variant, the DRAM can also be integrated in the SoC package (PoP). In this case, the physical protection of the I/O lines to the DRAM provided by the package is considered sufficient, so that the memory operations are not routed through the encryption block.

Thus, the TOE can be deployed on two different SoC hardware variants:

- SoC only (with external DRAM connected during integration).
- SoC with DRAM as package-on-package (PoP).

2.4.1.2 Physical delivery

The TOE is physically delivered as listed in the following table.

are listed in the following table.

Table 2-1 TOE physical delivery

Delivery item type	Identifier	Version	Form of delivery
Hardware	SM8250 Hardware	TCSR_SOC_HW_VERSION register (0x60080201) 0x6080 – chip family ID 0201 – version 2.1	Moulded Electronic Package (MEP)
Hardware	SM8250 ROM Code containing the Primary Bootloader (PBL)	TCSR_SOC_HW_VERSION register	Burned into ROM
QTI signed Binary Load Images	Qualcomm TEE kernel v5.8, CommonLib (tz.mbn)	TZ.XF.5.8-00041.3-SM8250AAAAANAZT-3	Electronic delivery via CreatePoint
QTI signed binary image.	XBL-SEC xbl_sec.mbn	BOOT.XF.3.2-00295-SM8250-1	Electronic delivery via CreatePoint
Library intended to be statically linked into TAs	applib.lib - 32 bit applib.lib - 64 bit common_applib.o - 32bit common_applib.o - 64bit	TZ.APPS.1.8.c1-00005-SM8250AAAAANAZT-1	Same than above
Binary Load Images	Mandatory QTI System TAs	See *2.3.3, Required non-TOE hardware, software, and firmware*	

The delivery item for additional feature set is listed in the following table.

Table 2-2 Delivery feature set

Feature set	Delivery item type	Identifier	Version	Form of delivery
Attestation	Binary file	qwes.mbn	TZ.APPS.1.8.c1-00005-SM8250AAAAANAZT-1	Electronic delivery via CreatePoint
eSE	Binary file	eseservice.mbn SCP11cry.mbn tee_se_api.lib	TZ.APPS.1.8.c1-00005-SM8250AAAAANAZT-1	Electronic delivery via CreatePoint

2.4.2 Physical scope of the TOE – Guidance documentation

2.4.2.1 Guidance for SoC integrators

The following documents are guidance for SoC integrators:

- *SM8250 Security Overview* (80-PK882-10 Rev C)
- *SM8250 Secure Boot Enablement User Guide* (80-PK882-9 Rev B)
- *Debug Policy Version 2 User Guide* (80-NV396-72 Rev E)
- *Provisioning Encryption Tool User Guide* (80-P1824-1 Rev B)
- *SM8250 QFPROM Programming Reference Guide* (80-PL546-97 Rev B)
- *Sectools: FuseBlower Tool User Guide* (80-NM248-3 Rev K)
- *Sectools: SecImage Tool (Version 5.20 and Later) User Guide* (80-NM248-8 Rev C)
- *Sectools: KeyProvision Tool User Guide* (80-NM248-5 Rev B)
- *Sectools: Debug Policy Tool User Guide* (80-NM248-6 Rev K)
- *Security Quick Start* (80-PF777-103 Rev J)
- *TEE-based Mobile Payment Security Guidelines for OEMs* (80-NR875-15 Rev J)
- *SM8250+SDX55M Software User Manual* (SP80-PK882-4 Rev G)
- *SM8250 Linux Peripheral (UART, SPI, I2C, I3C) Overview* (80-PK882-6 Rev C)
- *Secure Channel Protocol 11 Configuration Guide* (80-P2484-142 Rev A)
- KBA-191027204619*, eSE enablement* (Rev 7)
- *Qualcomm Trusted Execution Environment (TEE) Product Delivery User Guide* (80-NR875-20 Rev AA)

2.4.2.2 Guidance for TA developers

The following document is guidance for TA developers:

- *Secure Coding Guidelines for TrustZone QTEE Applications* (80-NK069-1 Rev A)
- *Qualcomm Trusted Execution Environment (TEE) Reference Manual* (80-NH537-4 Rev K)
- *Qualcomm TEE TA Software Developers Kit* (80-PF777-58 Rev A)
- *Qualcomm Wireless Edge Services (WES) API Reference* (80-PL230-2 Rev H)

2.4.2.3 Guidance for TEE final users

The OEM is responsible for communicating the proper use of the integrated device to the final TEE users.

This guidance might not be a document but a commercial notice. The OEM provides an explanation if there is no action or behavior expected from the TEE final user.

2.4.3 GlobalPlatform API functional compliance

The following table indicates the type of compliance with GlobalPlatform specifications (CF, DF, DP, or NI) for each TEE API:

- CF – *Certified Full functional compliance:* the TOE fully implements an approved version of the API and the TOE has successfully passed GlobalPlatform functional compliance testing for this API. The vendor shall provide the GlobalPlatform Letter of Qualification (LOQ).
- DF – *Declared Full functional compliance:* the TOE fully implements an approved version of the API but the compliance has not been qualified by GlobalPlatform.
- DP – *Declared Partial functional compliance:* the TOE partially implements an approved version of the API. The vendor shall identify the compliant/noncompliant parts of the API.
- NI – *Not Implemented:* TOE does not implement the API.

Table 2-3 Compliance: GlobalPlatform API specifications

Reference	GlobalPlatform device technology	Version	Compliance type
GPD_SPE_007	TEE Client API Specification	1.0	DP
GPD_EPR_028	TEE Client API Specification v1.0 Errata and Precisions	2.0	DF
GPD_SPE_010	TEE Internal Core API Specification	1.1.1	DP
GPD_SPE_024	TEE Secure Element API Specification	1.1	DP
GPD_SPE_020	Trusted User Interface API Specification	1.0	NI
GPD_SPE_025	TEE TA Debug Specification	1.0	NI
GPD_SPE_013	Secure Element Access Control	1.1	NI
(other)	Other GlobalPlatform specifications or versions (Examples are available to members only)	–	NI

The following table lists the compliant GlobalPlatform API test suite.

Table 2-4 Compliance: GlobalPlatform API test suite

Reference	GlobalPlatform device technology	Version	LOQ issuance date
–	TEE Initial Configuration Test Suite	1.1.0.1	N/A
(other)	Other GlobalPlatform test suites (Examples are available to members only)	–	–

2.4.4 Logical scope of the TOE

The application programming interfaces offered to the TAs consists of the following:

- GlobalPlatform-compliant API according to the standards detailed in *2.4.3, GlobalPlatform API functional compliance*.

- Proprietary Qualcomm TEE APIs according to the internal specifications detailed in *ISO 9001:2008 – Certificate Number: 110371.001; valid as of Jul 25, 2014 (DEKRA Certification Group)*.

The interface of Qualcomm TEE v5.8 to the REE consists of the following:

- Secure monitor call (SMC) interface that provides the services by which the non-secure CAs can communicate and interact with the Tas.
- Additional service APIs as detailed in *ISO 9001:2008 – Certificate Number: 110371.001; valid as of Jul 25, 2014 (DEKRA Certification Group)*.

The system is structured as shown in following figure.

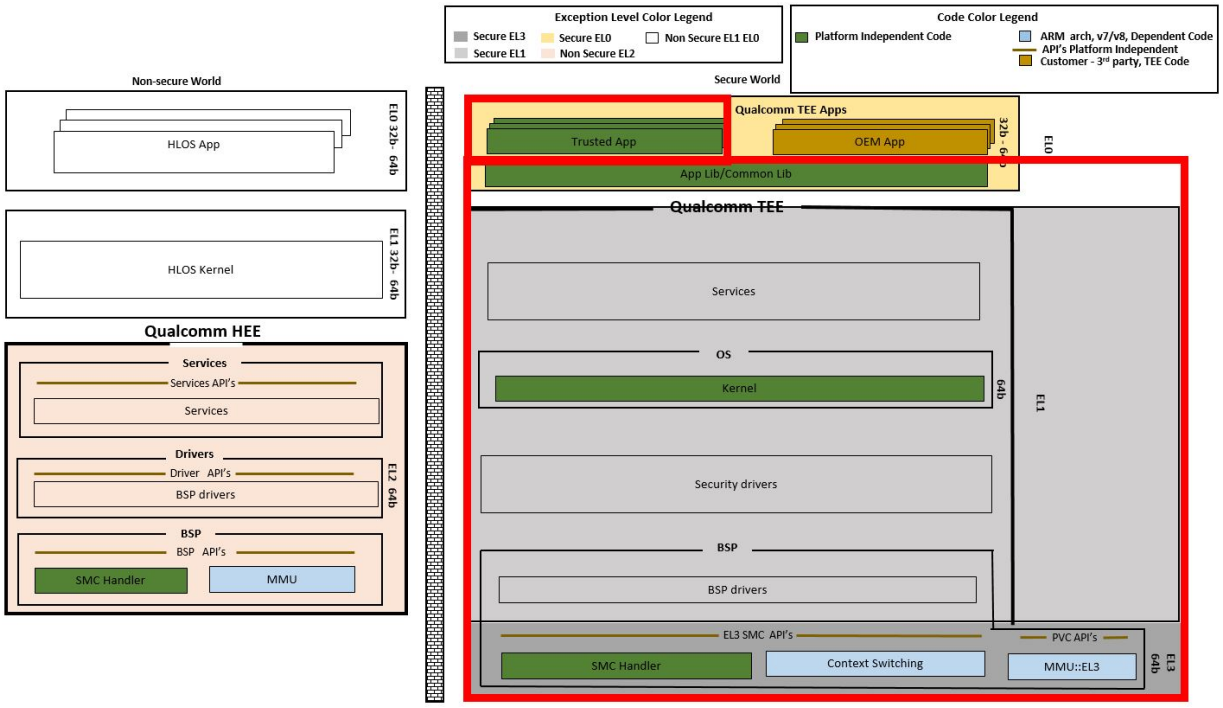

Figure 2-3 Qualcomm TEE v5.8 architecture

Referring to the figure:

- The components of the Qualcomm TEE core system are the components running in the high-privilege levels of the secure execution environment, which are depicted by the gray-shaded blocks. The kernel is implemented on top of the board support package (BSP), which in particular includes the SMC handler and functionality for context switching, as well as the memory management units (MMU).
- Additionally, the evaluation scope contains the CommonLib component loaded by the unified extensible firmware interface (UEFI), the AppLib intended to be compiled into the TAs depicted in the upper part of the red box and the TAs for the optional functionality referenced in *2.4.5, Optional functionality*.

- The additional QTI applications are shown in dark green in the red rectangle in the topmost level of the figure. They are not relevant from the security evaluation perspective because they are sandboxed like other TAs implemented by the OEMs. However, the mandatory applications are relevant for the proper functioning of the system, which is why they are listed as additional non-TOE components in *2.3.3, Required non-TOE hardware, software, and firmware.*

Additionally, the OEM still must properly configure several components according to the functional and security requirements of the system.

The Qualcomm HEE component is a privileged component that runs on a high-privileged level in the REE. Qualcomm HEE is intended to act as a secure proxy between REE and relieve Qualcomm TEE from non-security-sensitive tasks, such as controlling the application loading during the load process (observe that the loading itself is conducted by Qualcomm TEE).

2.4.5 Optional functionality

The TOE includes the following functionality that can be removed from the TOE:

- eSE Trusted Path,
- Device Attestation.

2.5 TOE life-cycle overview

The main components of the system as described in *2.4, TOE description* are developed, manufactured, integrated, deployed, and administered as follows.

1. Software design and development

   QTI designs and develops the secondary boot loader, the Qualcomm TEE core system, CommonLib, and the mandatory TAs (as specified in *2.4.1.2, Physical delivery*) and supplies them as images to the OEM. QTI signs the Qualcomm TEE kernel and the XBL-SEC image.

2. Hardware design and manufacturing
   a. QTI designs and manufactures the ROM code for the primary boot loader.
   b. QTI designs and manufactures the SoC, which includes the integration of the primary boot loader in the ROM of the SoC.

      Several verification steps, such as checking the ROM content and testing the integrated components, ensure the proper execution of the production steps and protect the integrity of the SoC.

   c. QTI preconfigures the SoC and burns QTI fuses.
3. Device integration and import of security anchors
   a. The OEM integrates all the hardware components, including RPMB, and is responsible for all subsequent production, preparation, and deployment steps.
   b. The OEM is responsible for integrating a suitable RPMB with the SoC during device integration.

c. The OEM is responsible for generating/deriving the secret key shared between the SoC and RPMB, and ensuring that the key material is properly and securely deployed to the SoC and RPMB during production.

d. The OEM blows security fuses.

4. Load image preparation and deployment

   The OEM signs the firmware images. Afterwards, the OEM deploys the Qualcomm TEE kernel to the nonvolatile memory of REE from where it can be securely bootstrapped by the secondary boot loader. The OEM deploys the signed TA images in a similar way so that they can be securely bootstrapped by the Qualcom TEE kernel.

5. Static library preparation and deployment by QTI and OEM

Developer/ manufacturer company name	Legal address	TOE-related sites	Site audits/date
QTI	5775 Morehouse Drive San Diego, CA 92121-1714 United States	QTI holds a groupwide ISO 9001 certification that includes all research and development as well as the manufacturing sites. The lead development and manufacturing site is as follows: 5775 Morehouse Drive San Diego, CA 92121 USA	ISO 9001, DEKRA Certification, June 3, 2020

Details can be found in the ISO 9001 certificate.

NOTE: ISO 9001:2015 – Certificate Number: 110371.001; valid as of June 02, 2020(DEKRA Certification Group).

In addition to the sites and activities in the table above, the following subsequent production steps are required to properly integrate the product.

- Integration of an RPMB that exhibits the required security properties (e.g., providing anti-rollback features.)
- Integration of external DRAM (for the non-PoP variant) and other non-TOE components that are required for proper functioning of the product.

These subsequent integration steps are not conducted under responsibility of QTI but under responsibility of the OEM, which is why no details on the manufacturing and development sites are given here.

3 Conformance claims

This Security Target claims conformance (CC) to:

- CC version claim: Conformance to CC Version 3.1 R5
- CC version claim in Security Target of base components: N/A since the TOE is not a composite TOE
- CC part 2 conformance: extended
- CC part 3 conformance: extended
- PP claims: strict conformance to *GlobalPlatform Device Committee TEE Protection Profile, Version 1.2.1 November 2016* (GPD_SPE_021), including the PP modules TEE TIME AND ROLLBACK and TEE DEBUG.
- Package claims: EAL2 augmented with AVA_TEE.2

The PP defines a TOE type is identical to the TOE type described in this ST. The strict conformance claim is emphasized by presenting the security problem definition taken from the PP and its modules without modification. The security problem definitions for the features that are unique to this TOE are presented in separate sections and augment the PP SPD without modification.

NOTE: See *GlobalPlatform Device Committee TEE Protection Profile, Version 1.2.1* (GPD_SPE_021) for details on PP.

4 Security problem definition

4.1 Security problem definition – TEE Base-PP and Modules

The security problem definition including the security problem definition extensions of the Time and Rollback PP-Module and the TEE Debug PP-Module are taken over into this security target as outlined in the following subsections. For detailed information see the PP.

4.1.1 Threats – TEE Base-PP and Modules

- T.ABUSE_FUNCT
- T.CLONE
- T.FLASH_DUMP
- T.IMPERSONATION
- T.PERTURBATION
- T.RAM
- T.RNG
- T.ROGUE_CODE_EXECUTION
- T.SPY
- T.STORAGE_CORRUPTION
- T.TEE_FIRMWARE_DOWNGRADE
- T.ROLLBACK
- T.TA_PERSISTENT_TIME_ROLLBACK
- T.ABUSE_DEBUG

4.1.2 Organizational security policies – TEE Base-PP and Modules

- OSP.INTEGRATION_CONFIGURATION
- OSP.SECRETS

4.1.3 Assumptions – TEE Base-PP and Modules

- A.PROTECTION_AFTER_DELIVERY
- A.TA_DEVELOPMENT

Observe that (in accordance to the PP) the assumption A.ROLLBACK is not taken over because it is addressed by the Time and Rollback PP Module (that is, the threat T.ROLLBACK).

4.2 Security problem definition – eSE

This section defined the security problem addressed by the trusted path between the TEE and an embedded secure element (eSE) and its operational environment. The operational environment is interpreted as it is interpreted in the PP. The TEE integration and maintenance environment and the TA development environment. Likewise, the security problem consists of the threats the TEE enabled device may face in the field, the assumptions on its operational environment and the organizational policies that have to be implemented by the TEE or within the operational environment.

4.2.1 Assumptions – eSE

This section presents the assumptions made for the eSE trusted path. One assumption is defined.

A.PROPER_eSE_COMMUNICATION_KEY_MATERIAL_HANDLING

It is assumed that both the developer and the OEM handles the key material involved in establishing the trusted path between the TEE and the (external) secure element in a secure manner.

4.2.2 Organizational security policies – eSE

There are no organizational security policies related to the establishment of a trusted path to the secure element.

4.2.3 Threats – eSE

This section presents the threats covered by the trusted path to the secure element. One threat is defined.

T.MODIFY_OR_DISCLOSE_eSE_COMMUNCIATION

A malicious TA or a malicious entity in the REE modify or disclose sensitive information exchanged between a TA and the (external) embedded secure element.

4.3 Security problem definition – Device Attestation

This section defined the security problem addressed by the Device Attestation security feature in the TEE and by the operational environment. The operational environment is interpreted as it is interpreted in the PP: The TEE integration and maintenance environment and the TA development environment. Likewise, the security problem consists of the threats the TEE enabled device may face in the field, the assumptions on its operational environment and the organizational policies that have to be implemented by the TEE or within the operational environment.

4.3.1 Assumptions – Device Attestation

There are no assumptions defined for device attestation.

4.3.2 Organizational security policies – Device Attestation

There are no organizational security policies related to the device attestation feature.

4.3.3 Threats – Device Attestation

One threats are defined that are related to the device attestation feature.

T.MODIFY_OR_DISCLOSE_ATTESTATION_DATA

An attacker tries to forge, modify or disclose attestation data supplied by the TOE.

5 Extended component definition

This Security Target uses the extended requirements as specified in PP:

- FCS_RNG.1
- FPT_INI.1
- AVA_TEE.2

NOTE: See *Chapter 6 of GlobalPlatform Device Committee TEE Protection Profile, Version 1.2.1* (GPD_SPE_021) for details.

The Security Target does not define any other extended requirements.

6 Security objectives

6.1 Security objectives – TEE Base-PP and Modules

The security objectives of the TEE PP, including the security objectives of the Time and Rollback PP-Module and the TEE Debug PP-Module are taken over into this security target as outlined in the following subsections.

6.1.1 Security objectives for environment – TEE Base-PP and Modules

- OE.INTEGRATION_CONFIGURATION
- OE.PROTECTION_AFTER_DELIVERY
- OE.SECRETS
- OE.TA_DEVELOPMENT

Observe, that OE.ROLLBACK is not taken over, because it is covered by the Time and Rollback PP-Module (that is, by O.ROLLBACK_PROTECTION).

6.1.2 Security objectives for TOE – TEE Base-PP and Modules

- O.CA_TA_IDENTIFICATION
- O.INITIALIZATION
- O.INSTANCE_TIME
- O.KEYS_USAGE
- O.OPERATION
- O.RNG
- O.RUNTIME_CONFIDENTIALITY
- O.RUNTIME_INTEGRITY
- O.TA_AUTHENTICITY
- O.TA_ISOLATION
- O.TEE_DATA_PROTECTION
- O.TEE_ID
- O.TEE_ISOLATION
- O.TRUSTED_STORAGE

- O.ROLLBACK_PROTECTION
- O.TA_PERSISTENT_TIME
- O.DEBUG

6.1.3 Security objectives rationale – TEE Base-PP and Modules

See Section 5.3 in *GlobalPlatform Device Committee TEE Protection Profile, Version 1.2.1* (GPD_SPE_021) for details on security objective rationale.

6.2 Security objectives – eSE

6.2.1 Security objectives for environment of TOE – eSE

OE.KEY_PROVISIONING_KEY_HANDLING

The device developer ensures that the environment in which keys are provisioned, handled and stored ensure confidentiality of the key at the same level as the TOE or eSE.

6.2.2 Security objectives for TOE – eSE

O.NO_INTERFERENCE_ON_eSE_PATH

The TOE shall manage the communication between TAs and the embedded Secure Element in a way that there is not interference, i.e. that communication data of one TA is not exposed to another TA communicating with the eSE.

O.TRUSTED_CHANNEL_TO_eSE

The TOE shall offer a service by which Trusted Applications can open a trusted channel to an (external) embedded Secure Element that protects the communication between the TA and the eSE against modification and disclosure.

6.2.3 Security objectives rationale – eSE

The following figure shows the mapping between the element of the security problem definition to the objectives for the TOE and the objectives for the environment related to the establishment of a trusted path to an external secure element.

O.NO_INTERFERENCE_ON_eSE_PATH addresses the threat T.MODIFY_OR_DISCLOSE_eSE_COMMUNICATIONS by isolating eSE communications per TA.

O.TRUSTED_CHANNEL_TO_eSE addresses the threat T_MODIFY_OR_DISCLOSE_eSE_COMMUNCATIONS by providing confidententiality, authenticity, integrity or combinations thereof by cryptographically secure messaging algorithms.

OE.KEY_PROVISIONING_KEY_HANDLING fulfills the assumption that key materials are managed in a secure environment.

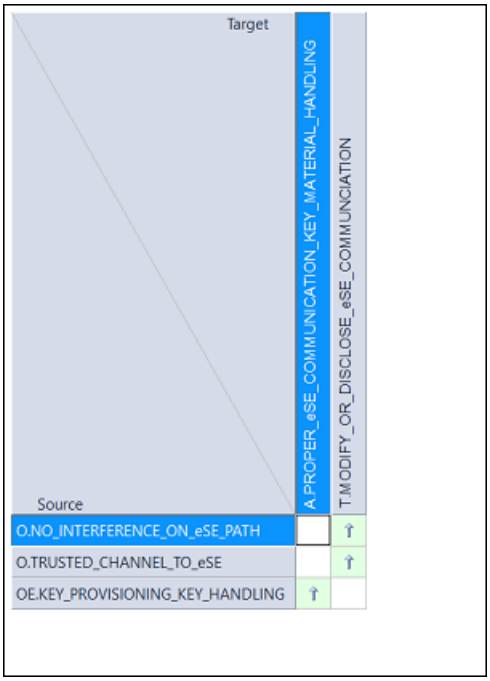

Figure 6-1 Security objectives rationale – eSE

6.3 Security objectives – Device Attestation

6.3.1 Security objectives for environment of TOE – Device Attestation

There are no security objectives for the environment defined for Device Attestation.

6.3.2 Security objectives for TOE – Device Attestation

O.ATTESTATION_DATA_GENERATION

The TOE shall generate attestation data that enables an external entity to reliably check the following pieces of information:

- Hardware versions
- Device ID
- OEM ID
- Integrity information about the REE
- Telemetry data
- Location information
- Qualcomm TEE software version

In addition to this the TOE shall also enable REE and trusted applications to extend the set of attestation information by additional pieces of information.

O.TRUSTWORTHY_AND_CONFIDENTIAL_ATTESTATION_DATA

The TOE shall allow for the export of the attestation data in way that the attestation data is trustworthy. Furthermore, the export functionality shall protect the confidentiality of the attestation data.

6.3.3 Security objectives rationale – Device Attestation

The following figure shows the mapping between the element of the security problem definition to the objectives for the TOE and the objectives for the environment related to the device attestation feature.

O.ATTESTATION_DATA_GENERATION addresses the threat T.MODIFY_OR_DISCLOSE_ATTESTATION_DATA by ensuring that attestation data is generated and prepared in the TEE.

O.TRUSTWORTHY_AND_CONFIDENTIAL_ATTESTATION_DATA addresses the threat T.MODIFY_OR_DISCLOSE_ATTESTATION_DATA by applying encryption and cryptographically strong signatures to the attestation data.

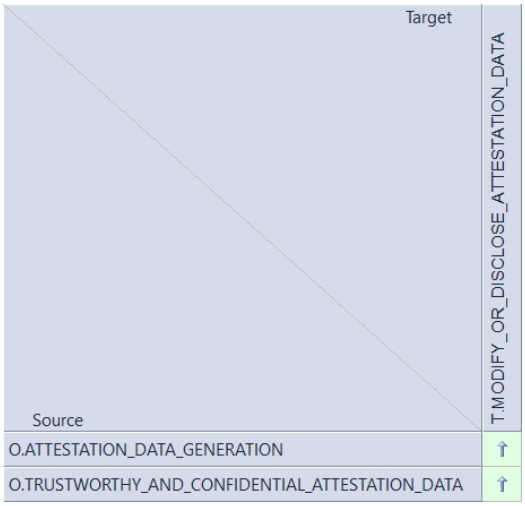

Figure 6-2 Security objectives rationale – Device Attestation

7 Security functional requirements

7.1 Security requirements for TEE base-PP

The security requirements presented here are taken verbatim from the PP. The SFRs that require an operation or refinement in the Security Target are marked by an asterisk (*).

Additional SFRs are marked by a plus (+).

NOTE: See *GlobalPlatform Device Committee TEE Protection Profile, Version 1.2.1* (GPD_SPE_021) for more information security requirements.

7.1.1 Identification and session management

7.1.1.1 FIA_ATD.1 User attribute definition *

FIA_ATD.1.1: The TSF shall maintain the following list of security attributes belonging to the individual users: **CA_identity, TA_identity, TA_properties , no additional attributes**

7.1.1.2 FIA_UID.2 User identification before any action

FIA_UID.2.1: The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

7.1.1.3 FIA_USB.1 User-subject binding *

- **FIA_USB.1.1**: The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:
  - **Client (CA or TA) identity is codified into the client_identity of the requested TA session**.
  - **No additional user security attributes**
- **FIA_USB.1.2**: The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:
  - **If the client is a TA, then the client_identity must be equal to the TA_identity of the TA subject, that is the client.**
  - **No other rules for initial association of attributes**

- **FIA_USB.1.3**: The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:
  - **No modification of client_identity is allowed after initialization.**
  - **No other rules for changing of attributes**

7.1.1.4 FMT_SMR.1 Security roles *

- **FMT_SMR.1.1**: The TSF shall maintain the roles.
  - **TSF**
  - **TA_User**
  - **No other authorized identified roles**
- **FMT_SMR.1.2**: The TSF shall be able to associate users with roles.

7.1.2 Confidentiality, integrity, and isolation (of runtime data in RAM and transfer)

7.1.2.1 FDP_IFC.2/Runtime Complete information flow control

- **FDP_IFC.2.1/Runtime**: The TSF shall enforce the **Runtime Data Information Flow Control SFP** on:
  - **Subjects: S.TA_INSTANCE, S.TA_INSTANCE_SESSION, S.API, S.COMM_AGENT, S.RESOURCE, S.RAM_UNIT**
  - **Information: I.RUNTIME_DATA**

  and all operations that cause that information to flow to and from subjects covered by the SFP.
- **FDP_IFC.2.2/Runtime**: The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

7.1.2.2 FDP_IFF.1/Runtime Simple security attributes *

- **FDP_IFF.1.1/Runtime**: The TSF shall enforce the **Runtime Data Information Flow Control SFP** based on the following types of subject and information security attributes: **S.RESOURCE.state, S.RAM_UNIT.rights and S.API.caller**.
- **FDP_IFF.1.2/Runtime**: The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

  Rules for information flow between S.TA_INSTANCE and S.RAM_UNIT:
  - **Flow of I.RUNTIME_DATA from S.TA_INSTANCE to S.RAM_UNIT is allowed only if S.RAM_UNIT.rights(S.TA_INSTANCE) is Write or ReadWrite**
  - **Flow of I.RUNTIME_DATA from S.RAM_UNIT to S.TA_INSTANCE is allowed only if S.RAM_UNIT.rights(S.TA_INSTANCE) is Read or ReadWrite**

Rules for information flow from and to S.COMM_AGENT:

- **Flow of I.RUNTIME_DATA from S.COMM_AGENT to S.RAM_UNIT is allowed only if S.RAM_UNIT.rights(REE) is Write or ReadWrite**
- **Flow of I.RUNTIME_DATA from S.RAM_UNIT to S.COMM_AGENT is allowed only if S.RAM_UNIT.rights(REE) is Read or ReadWrite**

Rules for information flow from and to S.API:

- **Flow of I.RUNTIME_DATA from S.API to S.RAM_UNIT is allowed only if S.RAM_UNIT.rights(S.API.caller) is Write or ReadWrite**
- **Flow of I.RUNTIME_DATA from S.RAM_UNIT to S.API is allowed only if S.RAM_UNIT.rights(S.API.caller) is Read or ReadWrite**

Rules for information flow from and to S.RESOURCE:

- **Flow of I.RUNTIME_DATA between S.API and S.RESOURCE is allowed only if the resource is under TEE control (S.RESOURCE.state = TEE).**

- **FDP_IFF.1.3/Runtime**: The TSF shall enforce **the following additional rules for information flow from and to S.TA_INSTANCE via shared buffers:**
  - **Flow of I.RUNTIME_DATA from S.TA_INSTANCE to S.RAM_UNIT and further on from S.RAM_UNIT to a CA or TA_TARGET_INSTANCE is allowed only if S.RAM_UNIT.rights(S.TA_INSTANCE) is Write or ReadWrite and S.RAM_UNIT.rights(REE or TA_TARGET_INSTANCE) is Read or ReadWrite**
  - **Flow of I.RUNTIME_DATA from S.RAM_UNIT from a CA or TA_SOURCE_INSTANCE and further on to S.RAM_UNIT to S.TA_INSTANCE is allowed only if S.RAM_UNIT.rights (REE or TA_TARGET_INSTANCE) is Write or ReadWrite and S.RAM_UNIT.rights(TA_INSTANCE) is Read or ReadWrite.**
- **FDP_IFF.1.4/Runtime**: The TSF shall explicitly authorise an information flow based on the following rules:

  Rules for information flow from and to S.TA_INSTANCE_SESSION:
  - **Flow of I.RUNTIME_DATA that are parameter or return values is allowed between S.TA_INSTANCE_SESSION and S.COMM_AGENT**
  - **Flow of I.RUNTIME_DATA that are parameter or return values is allowed between S.TA_INSTANCE_SESSION and S.API.**

- **FDP_IFF.1.5/Runtime**: The TSF shall explicitly deny an information flow based on the following rules:
  Any information flow involving a TEE subject unless one of the conditions stated in FDP_IFF.1.1/1.2/1.3/1.4 holds.

7.1.2.3 FDP_ITT.1/Runtime Basic internal transfer protection

FDP_ITT.1.1/Runtime: The TSF shall enforce the **Runtime Data Information Flow Control SFP** to prevent the **disclosure and modification** of user data when it is transmitted between physically- separated parts of the TOE.

7.1.2.4 FDP_RIP.1/Runtime Subset residual information protection

FDP_RIP.1.1/Runtime: The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource** from the following objects: **TEE and TA runtime objects**.

7.1.2.5 FPT_ITT.1/Runtime Basic internal TSF data transfer protection

FPT_ITT.1.1/Runtime: The TSF shall protect TSF data from **disclosure and modification** when it is transmitted between separate parts of the TOE.

7.1.3 Cryptography

7.1.3.1 FCS_CKM.1/ECC Cryptographic key generation – ECC key generation +

FCS_CKM.1.1: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **ECC** and specified cryptographic key sizes **160-bit, 192-bit, 224-bit, 256-bit, 320-bit, 384-bit, and 512-bit in GF(p)** that meet the following: **ECC_KG** according to *Digital Signature Standard (DSS)* (FIPS PUB 186-4) (for the definition of key generation requirements).

Application note

The user of the ECC services is responsible for selecting cryptographically strong elliptic curves for implementing security functionality only. It is strongly recommended to use the predefined or other curves based on approved international standards like the curves defined by NIST in FIPS PUB 186-4 or the Brainpool curves defined in *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation* (RFC 5639).

7.1.3.2 FCS_CKM.1/KD_PRF Cryptographic key generation – Key derivation based on pseudorandom functions +

FCS_CKM.1.1: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **Key Derivation Based on Pseudorandom Functions** and specified cryptographic key sizes **128-bit and 256-bit** that meet the following: **NIST_KD_PRF [AES_CMAC, COUNTER_MODE] according to Recommendation for Key Derivation Using Pseudorandom Functions (NIST SP 800-108)**.

7.1.3.3 FCS_CKM.1/PBKDF Cryptographic key generation – PBKDF +

FCS_CKM.1.1: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **Password-Based Key Derivation Function 2** and specified cryptographic key sizes **128-bit to 4096-bit** that meet the following: **NIST_PBKDF according to Recommendation for Password-Based Key Derivation (NIST SP 800-132)**.

7.1.3.4 FCS_CKM.1/RSAKeyGen Cryptographic key generation – RSA key generation +

FCS_CKM.1.1: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **RSA Key Generation** and specified cryptographic key sizes **1024-bit, 2048-bit, and 4096-bit** that meet the following: **RSA_KG** according to *PKCS #1: RSA Cryptography Standard*, Version 2.2.

7.1.3.5 FCS_CKM.1/SM2 Cryptographic key generation – SM2 generation +

FCS_CKM.1.1: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **SM2 Key Derivation Function** and specified cryptographic key sizes **256-bit** that meet the following: **draft-shen-sm2-ecdsa02**

7.1.3.6 FCS_CKM.4 Cryptographic Key Destruction +

FCS_CKM.4.1: The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **overwrite method** that meets the following: **no standard.**

7.1.3.7 FCS_COP.1/AES Cryptographic operation – Symmetric cipher *

FCS_COP.1.1: The TSF shall perform **AES encryption and decryption** in accordance with a specified cryptographic algorithm **AES in the following modes of operation: electronic code book (ECB[NO_PAD]), cipher block chaining (CBC[NO_PAD]), counter mode (CTR), counter with CBC MAC (CCM[NO_PAD|PKCS5_PAD]), cipher-based message authentication (CMAC), Galois/counter mode (GCM), and XEX tweakable block cipher with cipher text stealing (XTS) modes of operation** and cryptographic key sizes **128-bit and 256-bit** that meet the following:

- ***Specification for the Advanced Encryption Standard (AES)* (FIPS PUB 197) for the core encryption decryption primitives**
- ***Recommendation for Block Cipher Modes of Operation, Methods and Techniques* (NIST SP 800-38A) for the classical ECB, CBC, CTR, and CCM modes of operation**
- ***Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication* (NIST SP 800-38B) for the CMAC mode**
- ***Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC* (NIST SP 800-38D) for the Galois/Counter mode**
- ***Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices* (NIST SP 800-38E) for the XTS mode**

7.1.3.8 FCS_COP.1/DHKE Cryptographic operation – Diffie-Hellman key exchange *

FCS_COP.1.1 – The TSF shall perform **key exchange** in accordance with a specified cryptographic algorithm **Diffie-Hellman key exchange DH** and cryptographic key sizes **RSA keys with 1024-bit to 4096-bit** that meet ***Diffie-Hellman Key Agreement Method* (RFC 2631)**.

7.1.3.9 FCS_COP.1/ECDSA_SIGVER Cryptographic operation – Elliptic curve digital signature algorithm signature generation and verification *

FCS_COP.1.1: The TSF shall perform **ECC signature generation and verification** in accordance with a specified cryptographic algorithm **elliptic curve digital signature algorithm signature generation and verification (ECDSA_SIGVER)** and cryptographic key sizes **160-bit, 192-bit, 224-bit, 256-bit, 320-bit, 384-bit, and 512-bit over GF(p)** that meet ***Digital Signature Standard (DSS)* (FIPS PUB 186-4)**.

7.1.3.10 FCS_COP.1/ECIES Cryptographic operation – ECIES encryption and decryption *

FCS_COP.1.1: The TSF shall perform **ECIES encryption and decryption** in accordance with a specified cryptographic algorithm **ECDSA with curves NIST P-224, P-256, P-384 and P-521** and cryptographic key sizes **128-bit, 192-bit and 256-bit** that meet the following:

- **IEEE Std 1363a**
- **FIPS 186-4, SEC1**
- **RFC-5639**

7.1.3.11 FCS_COP.1/ECKA_DH Cryptographic operation – Elliptic curve Diffie Hellman key agreement *

FCS_COP.1.1: The TSF shall perform **ECC key agreement** in accordance with a specified cryptographic algorithm **elliptic curve Diffie-Hellman key agreement (EC_DH)** and cryptographic key sizes **160-bit, 192-bit, 224-bit, 256-bit, 320-bit, 384-bit, and 512-bit over GF(p)** that meet **Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (NIST SP 800-56A)**.

Application note

The user of the ECC services is responsible for selecting cryptographically strong elliptic curves for implementing security functionality only. It is strongly recommended to use the predefined or other curves based on approved international standards like the curves defined by NIST in Digital Signature Standard (DSS) (FIPS PUB 186-4) or the Brainpool curves defined in Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation (RFC 5639).

7.1.3.12 FCS_COP.1/HMAC Cryptographic operation – Hash-based message authentication *

FCS_COP.1.1: The TSF shall perform **message authentication** in accordance with a specified cryptographic algorithm **keyed-hash message authentication (HMAC[HASH=MD5, SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-256)** and cryptographic key sizes **128-bit and 256-bit** that meet ***The Keyed-Hash Message Authentication Code (HMAC)* (FIPS PUB 198-1)**.

7.1.3.13 FCS_COP.1/MD5 Cryptographic operation – Cryptographic hashing *

FCS_COP.1.1: The TSF shall perform **hashing** in accordance with a specified cryptographic algorithm **MD5** and cryptographic key sizes **none** that meet ***The MD5 Message-Digest Algorithm* (RFC 1321)**.

Application note

The MD5 standard is no longer recommended for security applications because it is considered cryptographically broken and unsuitable for further use. It is listed because the system still provides legacy support of this service.

7.1.3.14 FCS_COP.1/RSA_ENCDEC Cryptographic operation – RSA encryption and decryption *

FCS_COP.1.1: The TSF shall perform **RSA encryption and decryption** in accordance with a specified cryptographic algorithm **RSA_ES_PKCS1_V15 and RSA_ES_OAEP [HASH=SHA1|SHA2-224|SHA2-256|SHA2-384|SHA2-512, MGF=MGF1** and cryptographic key sizes **1024-bit, 2048-bit, and 4096-bit** that meet ***PKCS #1: RSA Cryptography Standard*, Version 2.2**.

7.1.3.15 FCS_COP.1/RSA_SIGVER Cryptographic operation – RSA signature generation and verification *

FCS_COP.1.1: The TSF shall perform **RSA signature generation and verification** in accordance with a specified cryptographic algorithm **RSA_SSA_PKCS1_V15 [HASH=MD5|SHA1|SHA2-224|SHA2-256|SHA2-384|SHA2-512], RSA_SSA_PSS [HASH=SHA1|SHA2-224|SHA2-256|SHA2-384|SHA2-512, MGF=MGF1, and RSA_FIPS_DSA[HASH=SHA1|SHA2-224|SHA2-256** and cryptographic key **1024-bit, 2048-bit, and 4096-bit** that meet ***PKCS #1: RSA Cryptography Standard*, Version 2.2, and *Digital Signature Standard (DSS)* (FIPS PUB 186-4)**.

7.1.3.16 FCS_COP.1/SHA Cryptographic operation – Cryptographic hashing *

FCS_COP.1.1: The TSF shall perform **hashing** in accordance with a specified cryptographic algorithm **SHA1, SHA2-224, SHA2-256, SHA2-384, and SHA2-512** and cryptographic key sizes **none** that meet **Secure Hash Standard (SHS) (FIPS PUB 180-4)**.

Application note

The SHA1 standard is no longer recommended for arbitrary security applications due to its limited security strength.

7.1.3.17 FCS_COP.1/SM3 Cryptographic operation – SM3 Cryptographic hashing *

FCS_COP.1.1: The TSF shall perform **hashing** in accordance with a specified cryptographic algorithm **SM3** and cryptographic key sizes **none** that meet ***SM3 Cryptographic Hash Algorithm***

7.1.3.18 FCS_COP.1/SM4 Cryptographic operation – SM4 Symmetric cipher *

FCS_COP.1.1: The TSF shall perform **SM4 encryption and decryption** in accordance with a specified cryptographic algorithm **SM4 standard mode of operation** and cryptographic key sizes **128-bit** that meet the following: **GB/T 32907-2016**

7.1.3.19 FCS_COP.1/TDES Cryptographic operation – Triple DES encryption decryption *

FCS_COP.1.1: The TSF shall perform **2TDES/3TDES encryption and decryption** in accordance with a specified cryptographic algorithm **2TDES/3TDES in the following modes of operation: electronic code book (ECB), cipher-block-chaining in MAC mode (CBC_MAC)** and cryptographic key sizes **128-bit (112-bit effective) for 2TDES, and 168-bit (156-bit effective) for 3TDES** that meet ***Data Encryption Standard (DES)*** **(FIPS PUB 46-3) and *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher* (NIST SP 800-67)**.

7.1.3.20 FDP_ACC.1/TA_keys Subset access control

FDP_ACC.1.1/TA_keys The TSF shall enforce the **TA Keys Access Control SFP** on

- **Subjects: S.API, S.TA_INSTANCE and any other subject in the TEE**
- **Objects: OB.TA_KEY**
- **Operations: OP.USE_KEY, OP.EXTRACT_KEY.**

7.1.3.21 FDP_ACF.1/TA_keys Security attribute based access control *

- **FDP_ACF.1.1/TA_keys**: The TSF shall enforce the **TA Keys Access Control SFP** to objects based on the following**: OB.TA_KEY.usage, OB.TA_KEY.owner, OB_TA_KEY.isExtractable and S.API.caller**.
- **FDP_ACF.1.2/TA_keys**: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

  OP.USE_KEY is allowed if the following conditions hold:

  - **The TA instance that requested the operation to the API owns the key (S.API.caller = OB.TA_KEY.owner).**

- **The intended usage of the key (OB.TA_KEY.usage) matches the requested operation.**

OP.EXTRACT_KEY is allowed if the following conditions hold:

- **The TA instance that requested the operation to the API owns the key (S.API.caller = OB.TA_KEY.owner).**
- **The operation attempts to extract the public part of OB.TA_KEY or the key is extractable (OB.TA_KEY.isExtractable = True).**

- **FDP_ACF.1.3/TA_keys**: The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none.**
- **FDP_ACF.1.4/TA_keys**: The TSF shall explicitly deny access of subjects to objects based on the following additional rules:
  - **Any access to a user key attempted directly from S.TA_INSTANCE or any other subject of the TEE that is not S.API.**
  - **Any access to a user key attempted from S.API without valid caller (S.API.caller is undefined).**
  - **No further rules deny requests to TA_keys.**

7.1.3.22 FMT_MSA.1/TA_keys Management of security attributes

FMT_MSA.1.1/TA_keys: The TSF shall enforce the **TA Keys Access Control SFP** to restrict the ability to **change_default, query and modify** the security attributes **OB.TA_KEY.usage, OB.TA_KEYS.isExtractable and OB.TA_KEY.owner** to **the following roles:**

- **change_default, query and modify OB.TA_KEY.usage to TA_User role**
- **query OB.TA_KEY.owner to the TSF role.**

7.1.3.23 FMT_MSA.3/TA_keys Static attribute initialisation *

- **FMT_MSA.3.1/TA_keys**: The TSF shall enforce the **TA Keys Access Control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.
- **FMT_MSA.3.2/TA_keys** : The TSF shall allow the **TA_User role, no other authorized roles** to specify alternative initial values to override the default values when an object or information is created.

7.1.4 Initialization, operation, and firmware integrity

7.1.4.1 FAU_ARP.1 Security alarms *

FAU_ARP.1.1: The TSF shall take **[assignment: list of actions]** upon detection of a potential security violation.

Refinement:

The TSF shall take the following actions upon detection of a potential security violation:

- detection of consistency violation of TA data, TA code or TEE data: **prevent use of the corrupted TA data, abort loading of corrupted TA code, and prevent use of the corrupted TEE data**.
- detection of TEE firmware integrity violation: **abort the loading and execution of the corrupted TEE firmware**.
- **Detection of a corruption of the internal TEE state: initiate a fatal error**.

7.1.4.2 FDP_SDI.2 Stored data integrity monitoring and action *

FDP_SDI.2.1: The TSF shall monitor user data stored in containers controlled by the TSF for **[assignment: integrity errors]** on all objects, based on the following attributes: **[assignment: user data attributes]**.

Refinement:

The TSF shall monitor TEE runtime data, TEE persistent data, TA data and keys and TA code stored in containers controlled by the TSF for **authenticity and consistency errors** on all objects, based on the following attributes: **location in secure memory for TEE runtime data, authenticating MAC for TEE persistent data, TA data and keys and code signature for TA code**.

FDP_SDI.2.2: Upon detection of a data integrity error, the TSF shall **[assignment: action to be taken]**.

Refinement:

- Upon detection of authenticity or consistency errors in TEE runtime data or TEE persistent data, the TSF shall **not use the data but fail in a safe manner**.
- Upon detection of TA code authenticity or consistency errors, the TSF shall **abort the execution of the TA instance**
- Upon detection of TA data or TA keys authenticity or consistency errors, the TSF shall
  - **Not give back any compromised data,**
  - **signal the consistency error to the calling TA.**
- **No other actions are taken**.

7.1.4.3 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1: The TSF shall be capable of performing the following management functions:

- Management of TA keys security attributes
- Provision of Trusted Storage security attributes to authorised users.

7.1.4.4 FPT_FLS.1 Failure with preservation of secure state *

FPT_FLS.1.1: The TSF shall preserve a secure state when the following types of failures occur:

- **Device binding failure**
- **Cryptographic operation failure**
- **Invalid CA requests, in particular bad-formed requests**
- **Panic states (as defined in [IAPI], Section 2.2.3)**
- **TA code, TA data or TA keys authenticity or consistency failure**
- **TEE data (in particular TA properties, TEE keys and all security attributes) authenticity or consistency failure**
- **TEE firmware integrity failure**
- **TEE initialization failure**
- **Unexpected commands in the current TEE state**
- **Inconsistent internal TEE state**.

7.1.4.5 FPT_INI.1 TSF initialisation *

- **FPT_INI.1.1**: The TOE initialization function shall verify:
  - The integrity of TEE initialization code and data
  - Authenticity and integrity of TEE firmware
  - Integrity of the storage root of trust
  - Integrity of the TEE identification data
  - Version of the firmware to prevent downgrade to previous versions
  - **No other implementation-dependent verifications prior to establishing the TSF in a secure initial state**.
- **FPT_INI.1.2**: The TOE initialization function shall detect and respond to errors and failures during initialization such that the TOE either successfully completes initialization or is halted.
- **FPT_INI.1.3**: The TOE initialization function shall not be able to arbitrarily interact with the TSF after TOE initialization completes.

7.1.4.6 FPT_TEE.1 Testing of external entities *

- **FPT_TEE.1.1**: The TSF shall run a suite of tests **prior execution and under no other conditions** to check the fulfillment of authenticity of TA code.
- **FPT_TEE.1.2**: If the test fails, the TSF shall **not start the execution of the TA instance**.

7.1.5 TEE Identification

7.1.5.1 FAU_SAR.1 Audit review

- **FAU_SAR.1.1**: The TSF shall provide **all users** with the capability to read **TEE identifier** from the audit records.
- **FAU_SAR.1.2**: The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

7.1.5.2 FAU_STG.1 Protected audit trail storage

- **FAU_STG.1.1**: The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
- **FAU_STG.1.2**: The TSF shall be able to **prevent** unauthorised modifications to the stored audit records in the audit trail.

Application note:

The data required to produce the device unique identifier involves using TEE-diversified information embedded generated off-TEE and integrated into the product during integration. A specifically protected persistent memory is used to store the required information. Details about these security mechanism are described in the security architecture of the TOE and assessed in the evaluation.

7.1.6 Instance time

7.1.6.1 FPT_STM.1/Instance time Reliable time stamps

FPT_STM.1.1/Instance time: The TSF shall be able to provide reliable time stamps.

Refinement

The TSF shall be able to provide time stamps to TA instances such that time stamps are monotonic during the TA instance lifetime.

7.1.7 Random number generator

7.1.7.1 FCS_RNG.1 Random Number Generator *

- FCS_RNG.1.1: The TSF shall provide a **deterministic** random number generator that implements:
  - **(DRG.3.1) If initialized with a random seed drawn from several independent hardware random sources each of which exhibiting an entropy of more than 0.999 per bit which the internal state of the RNG shall have at least 256 bits of entropy.**
  - **(DRG.3.2) The RNG provides forward secrecy.**

- **(DRG.3.3) The RNG provides backward secrecy even if the current internal state is known.**
- **Online total failure tests of the DRNG output**.

- FCS_RNG.1.2: The TSF shall provide random numbers that meet:
  - **(DRG.3.4) The RNG, initialized with a random seed drawn from several independent hardware random sources each of which exhibiting an entropy of more than 0.999 per bit which are subsequently XORed to produce a single bit of the bit sequence used for seeding, generates output for which 2^{34} strings of bit length 128 are mutually different with probability 2^{-16}.**
  - **(DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A.**

Application note

- The TRNG fulfils the entropy requirements required for a proper seeding of the DRNG, and the DRNG exhibits all required properties for providing unpredictable random numbers that pass the standard test procedures. The RNG did not fall into one of the predefined RNG classes because the mandated (online) self-tests are structured differently. However, the evaluation will assess that sufficiently biasing the TRNG is not a suitable attack path for the assumed attack potential.

7.1.8 Trusted storage

7.1.8.1 FDP_ACC.1/Trusted Storage Subset access control

FDP_ACC.1.1/Trusted Storage The TSF shall enforce the **Trusted Storage Access Control SFP** on

- **Subjects: S.API**
- **Objects: OB.TA_STORAGE, OB.SRT**
- **Operations: OP.LOAD, OP.STORE.**

7.1.8.2 FDP_ACF.1/Trusted Storage Security attribute based access control *

- FDP_ACF.1.1/Trusted Storage :The TSF shall enforce the **Trusted Storage Access Control SFP** to objects based on the following: **S.API.caller, OB.TA_STORAGE.owner, OB.TA_STORAGE.inExtMem, OB.TA_STORAGE.TEE_identity and OB.SRT.TEE_identity**.
- FDP_ACF.1.2/Trusted Storage The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

  OP.LOAD of an object from OB.TA_STORAGE is allowed if the following conditions hold:
  - **The operation is performed by S.API**

- **The load request comes from an instance of the owner of the trusted storage space (S.API.caller = OB.TA_STORAGE.owner)**
- **OB.TA_STORAGE is bound to the TEE storage root of trust OB.SRT (OB.TA_STORAGE.TEE_identity = OB.SRT.TEE_identity)**
- **If OB.TA_STORAGE is located in external memory accessible to the REE (OB.TA_STORAGE.inExtMem = True) then the object is authenticated and decrypted before load.**

OP.STORE of an object to OB.TA_STORAGE is allowed if the following conditions hold:

- **The operation is performed by S.API**
- **The store request comes from an instance of the owner of the trusted storage space (S.API.caller = OB.TA_STORAGE.owner)**
- **OB.TA_STORAGE is bound to the TEE storage root of trust OB.SRT OB.TA_STORAGE.TEE_identity = OB.SRT.TEE_identity)**
- **If OB.TA_STORAGE is located in external memory accessible to the REE (OB.TA_STORAGE.inExtMem = True) then the object is signed and encrypted before storage.**

- FDP_ACF.1.3/Trusted Storage: The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none.**
- FDP_ACF.1.4/Trusted Storage: The TSF shall explicitly deny access of subjects to objects based on the following additional rules:
  - **Any access to a trusted storage attempted from S.API without valid caller (S.API.caller = undefined)**
  - **Any access to a trusted storage that was bound to a different TEE (OB.TA_STORAGE.TEE_identity different from OB.SRT.TEE_identity)**
  - **Any access to a trusted storage from a subject different from S.API**
  - **No additional rules**.

7.1.8.3 FDP_ITT.1/Trusted Storage Basic internal transfer protection

FDP_ITT.1.1/Trusted Storage The TSF shall enforce the **Trusted Storage Access Control SFP** to prevent the **disclosure and modification** of user data when it is transmitted between physically-separated parts of the TOE.

7.1.8.4 FDP_ROL.1/Trusted Storage Basic rollback *

- FDP_ROL.1.1/Trusted Storage: The TSF shall enforce the **Trusted Storage Access Control SFP** to permit the rollback of the **unsuccessful or interrupted OP.STORE operation** on **the storage**.
- FDP_ROL.1.2/Trusted Storage: The TSF shall permit operations to be rolled back within the **single update operation on an object using the GP API or the Secure File System**.

Application note

This SFR enforces atomicity of any write operation [IAPI].

The system offers additional memory write operations that do not enforce atomicity for performance reasons, but when using theses services the using application is made aware that it cannot rely on this property.

7.1.8.5 FMT_MSA.1/Trusted Storage Management of security attributes

FMT_MSA.1.1/Trusted Storage: The TSF shall enforce the **Trusted Storage Access Control SFP** to restrict the ability to **query** the security attributes **OB.TA_STORAGE.owner, OB.TA_STORAGE.inExtMem, OB.TA_STORAGE.TEE_identity and OB.SRT.TEE_identity** to **TA_User role**.

7.1.8.6 FMT_MSA.3/Trusted Storage Static attribute initialisation

- FMT_MSA.3.1/Trusted Storage: The TSF shall enforce the **Trusted Storage Access Control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.
- FMT_MSA.3.2/Trusted Storage: The TSF shall allow the **TA_User** to specify alternative initial values to override the default values when an object or information is created.

7.1.9 Security requirements rationale

See Section 7.3, *Security Requirements Rationale* in *GlobalPlatform Device Committee TEE Protection Profile, Version 1.2.1* (GPD_SPE_021) for more information.

The only additional SFRs specified in this security target are detailed iterations for all supported crypto operations which are in the scope of the assessment (FCS_COP.1/xxx and FCS_CKM.1/xxx) requirements. Furthermore, a dedicated FCS_CKM.4 SFR captures the related key destruction method.

The security rationale for these SFRs is captured in the TEE PP by the rationale for the single FCS_COP.1 requirements which acts as a place-holder for the cryptographic primitives implemented in the TOE. Therefore, the TEE PP also defines properly the SFR rationale for the detailed SFRs.

7.1.9.1 Security requirements dependencies

See Section 7.3.3, *Dependencies* in *GlobalPlatform Device Committee TEE Protection Profile, Version 1.2.1* (GPD_SPE_021) for more information on table defining the dependencies for the SFRs in the PP and how they are fulfilled.

For some SFRs, the dependencies are not satisfied, and the PP provides reasoning for discarding these dependencies. The TSF provides the dependencies for some SFRs, specifically FCS_CKM.1/* and FCS_CKM.4. The table presented here shows how the open dependencies are fulfilled.

Requirements	CC Dependencies	Satisfied Dependencies
FCS_COP.1/Debug	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/*, FCS_CKM.4
FCS_COP.1/*	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/*, FCS_CKM.4
FAU_ARP.1	(FAU_SAA.1)	Discarded in PP
FAU_SAR.1	(FAU_GEN.1)	Discarded in PP
FAU_STG.1	(FAU_GEN.1)	Discarded in PP
FCS_CKM.1 (+)	[FCS_CKM.2 or FCS_COP.1] and FCS_CKM.4	FCS_COP.1, FCS_CKM.4
FCS_CKM.4 (+)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1)	FCS_CKM.1

7.2 TEE time and rollback PP-Module

7.2.1 Rollback protection

7.2.1.1 FDP_SDI.2/Rollback – Stored data integrity monitoring and action *

FDP_SDI.2.1/Rollback: The TSF shall monitor user data stored in containers controlled by the TSF for **[assignment: integrity errors]** on all objects, based on the following attributes: **[assignment: user data attributes]**.

Refinement

The TSF shall monitor **TEE rollback detection data, TEE runtime data, TEE persistent data, TA data and keys, and TA code** stored in containers controlled by the TSF for **integrity errors** on all objects, based on the following attributes:

- **TEE rollback detection data attributes: version information of the previously loaded Qualcomm TEE and GPLib images.**
- **TEE runtime data and TEE persistent data attributes: the number of executed write operations.**
- **TA data and keys attributes: the number of executed write operations.**
- **TA code attributes: version information of the previously loaded TA images.**

FDP_SDI.2.2/Rollback: Upon detection of a data integrity error, the TSF shall **[assignment: action to be taken]**.

Refinement

- Upon detection of integrity errors in TEE rollback detection data, TEE runtime data, or TEE persistent data, the TSF shall **behave in a manner that does not depend on the compromised data.**

- Upon detection of TA code integrity errors, the TSF shall **abort the execution of the TA instance.**
- Upon detection of TA data or TA keys integrity errors, the TSF shall:
  - **Not provide any compromised data.**
  - **Behave in a manner that does not depend on the compromised data.**
- **No other actions.**

7.2.1.2 FPT_FLS.1/Rollback – Failure with preservation of secure state

FPT_FLS.1.1/Rollback: The TSF shall preserve a secure state when the following types of failures occur:

- **TA code and data integrity failure.**
- **TEE persistent data integrity failure.**

Application note:

This requirement is a complement to FPT_FLS.1. See Section *8.1.4, Initialization, operation, and firmware integrity.*

7.2.2 TA persistent time

7.2.2.1 FPT_STM.1/Persistent time – Reliable timestamps

FPT_STM.1.1/Persistent time: The TSF shall be able to provide reliable timestamps.

Refinement

The TSF shall be able to provide timestamps to TA instances such that:

- Timestamps are persistent over TEE reset
- Timestamps are monotonic between two time setting operations performed by any instance of the TA

The TSF shall invalidate any persistent time that does not meet the monotonicity property.

7.2.2.2 FMT_MTD.1/Persistent time – Management of TSF data

FMT_MTD.1.1/Persistent time: The TSF shall restrict the ability to **perform a time setting operation on** the **TA persistent time** to **any instance of the TA.**

7.2.2.3 FMT_SMF.1/Persistent time – Specification of management functions

FMT_SMF.1.1/Persistent time: The TSF shall be capable of performing the following management functions: **time setting operation for TA persistent time.**

7.2.3 Security requirements rationale

For the security requirements rationale for the SFRs of the TEE Time and Rollback-Module refer to the TEE PP.

7.3 Security requirements - TEE debug PP-Module

7.3.1 Debug requirements

7.3.1.1 FDP_ACC.1/Debug – Subset access control

FDP_ACC.1.1/Debug: The TSF shall enforce the **Debug Access Control SFP** on:

- **Subjects: S.DEBUG**
- **Objects: all objects**
- **Operations: OP.ACTIVATE, OP.DEBUG**

7.3.1.2 FDP_ACF.1/Debug – Security attribute based access control *

FDP_ACF.1.1/Debug: The TSF shall enforce the **Debug Access Control SFP** to objects based on the following:

- **S.DEBUG.enabled, S.DEBUG.authenticated**
- **No further subjects and objects**

FDP_ACF.1.2/Debug: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- **OP.AUTHENTICATE is allowed if the following conditions hold:**
  - **The operation is performed by S.DEBUG**
  - **The debug interface is enabled (S.DEBUG.enabled = True)**
- **OP.DEBUG on all objects is allowed if the following conditions hold:**
  - **The operation is performed by S.DEBUG**
  - **The debug interface is enabled (S.DEBUG.enabled = True)**
  - **The TEE Debug Administrator is authenticated (S.DEBUG.authenticated = True)**
- **No further rules**

FDP_ACF.1.3/Debug: The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **None**.

FDP_ACF.1.4/Debug: The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **None**.

7.3.1.3 FCS_COP.1/Debug – Cryptographic operation *

FCS_COP.1.1/Debug – The TSF shall perform **authentication of the TEE Debug Administrator or the actor acting on his or her behalf** in accordance with a specified cryptographic algorithm **RSA_SSA_PKCS1_V15[HASH=MD5|SHA1|SHA2-224|SHA2-256| SHA2-384|SHA2-512], RSA_SSA_PSS[HASH=SHA1|SHA2-224|SHA2-256|SHA2_384| SHA2-512, MGF=MGF1], or RSA_FIPS_DSA[HASH=SHA1|SHA2-224|SHA2-256]** and cryptographic key sizes **1024-bit, 2048-bit, and 4096-bit** that meet ***PKCS #1: RSA Cryptography Standard*, Version 2.2, and *Digital Signature Standard (DSS)* (FIPS PUB 186-4)**

Application note

The cryptographic protection of the debug features is linked to the above-mentioned security services, which also serve a different purpose in the system.

7.3.1.4 FMT_SMR.1/Debug – Security roles

FMT_SMR.1.1/Debug: The TSF shall maintain the roles **TEE Debug Administrator.**

FMT_SMR.1.2/Debug: The TSF shall be able to associate users with roles.

7.3.1.5 FIA_UID.2/Debug – User identification before any action

FIA_UID.2.1/Debug [editorially refined]: The TSF shall require each user to be successfully identified before allowing any other TSF-mediated **debug** actions on behalf of that user.

7.3.1.6 FIA_ATD.1/Debug – User attribute definition

FIA_ATD.1.1/Debug: The TSF shall maintain the following list of security attributes belonging to individual users: **S.DEBUG.enabled, S.DEBUG.authenticated.**

7.3.1.7 FIA_USB.1/Debug – User-subject binding *

- FIA_USB.1.1/Debug: The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **S.DEBUG.enabled, S.DEBUG.authenticated.**
- FIA_USB.1.2/Debug: The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **S.DEBUG.authenticated is False**.
- FIA_USB.1.3/Debug: The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:
  - **S.DEBUG.authenticated is set to True after TEE Debug Administrator successful authentication**

- **S.DEBUG.authenticated is set to False when the authentication is lost, for instance after power-off** (see Section *7.3.1.9, FIA_UAU.6/Debug – Re-authenticating*).
- **No further rules for changing the security attributes**.

7.3.1.8 FIA_UAU.2/Debug – User authentication before any action

FIA_UAU.2.1/Debug [editorially refined]: The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated **debug** actions on behalf of that user.

7.3.1.9 FIA_UAU.6/Debug – Re-authenticating *

FIA_UAU.6.1/Debug: The TSF shall re-authenticate the user under the conditions:

- **After TEE power-off**
- **No additional conditions**

7.3.2 Security requirements rationale

For the security requirements rationale for the TEE Debug PP-Module refer to the TEE PP.

7.4 Security requirements – eSE trusted path

7.4.1 eSE requirements

7.4.1.1 FCS_CKM.2/eSE Cryptographic key distribution +

FCS_CKM.2.1: The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method certificate-based key import and export that meets the following: SCP11a as specified in Card Specification v2.2 - Amendment F.

Application notes

The communications channel between the eSE and the TSF is not exclusively assigned to the eSE trusted path functionality. Alternative secure channel mechanisms can be implemented by the OEM that can provide similar security assurance, but these are not considered as part of the certification scope.

FCS_CKM.2/eSE Cryptographic key distribution only applies if the eSE Trusted Path module is present in the TOE.

7.4.1.2 FTP_TRP.1/eSE Trusted path +

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1: The TSF shall provide a communication path between itself and local, remote users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification, disclosure.

FTP_TRP.1.2: The TSF shall permit local users to initiate communication via the trusted path.

FTP_TRP.1.3: The TSF shall require the use of the trusted path for establishing and enforcing secure communication between TAs and an embedded Secure Element.

Application note

The local user refers to a TA operating in the TEE, while the remote user refers to the external embedded secure element. The TSF is implemented by the TEE itself which provides the required channel management and encryption services in a transparent way to the TAs. The communication data is not only protected against disclosure towards the REE but also against disclosure towards other TAs also communicating with the eSE.

The communications channel between the eSE and the TSF is not exclusively assigned to the eSE trusted path functionality. Alternative secure channel mechanisms can be implemented by the OEM that can provide similar security assurance, but these are not considered as part of the certification scope.

FCS_TRP.1/eSE Trusted Path only applies if the eSE Trusted Path module is present in the TOE.

7.4.2 Security requirements rationale

The security requirement FCS_CKM.2/eSE fulfills the objective O.TRUSTED_CHANNEL_TO_eSE by providing a means to confidentially exchange a communications key between the embedded SE and the TEE.

FTP_TRP.1/Trusted Path fulfills O.NO_INTERFERENCE_ON_eSE_PATH by providing a industry-standard encrypted communications channel that is end-to-end between the eSE and the TEE.

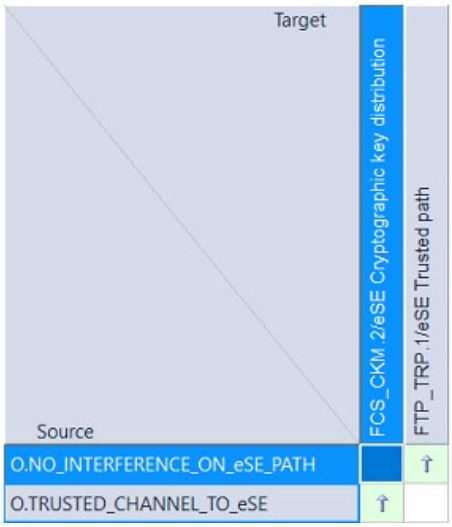

Figure 7-1 Security Requirements Rationale – eSE

7.4.2.1 SFR dependencies

Requirements	CC Dependencies	Satisfied by
FCS_CKM.2/eSE	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 (Base PP) FCS_CKM.4 (Base PP)
FTP_TRP.1/eSE	No dependencies	

7.5 Security requirements – Device Attestation

7.5.1 Device Attestation requirements

7.5.1.1 FCO_NRO.2 Enforced proof of origin +

Hierarchical to: FCO_NRO.1 Selective proof of origin

Dependencies: FIA_UID.1 Timing of identification

- **FCO_NRO.2.1:** The TSF shall **enforce the generation of** evidence of origin for transmitted attestation information at **all times.**
- **FCO_NRO.2.2:** The TSF shall be able to relate the key material for signature of the originator of the information, and the *signature* of the information to which the evidence applies.
- **FCO_NRO.2.3:** The TSF shall provide a capability to verify the evidence of origin of information to recipient given limitations of the digital signature.

Application Note

FCO_NRO.2 Enforced proof of origin only applies if the Device Attestation module is present in the TOE.

7.5.1.2 FPT_TST.1/Attestation TSF testing

FPT_TST.1 TSF testing +

Hierarchical to: No other components.

Dependencies: No dependencies.

- **FPT_TST.1.1:** The TSF shall run a suite of self tests at the conditions (during during initial start-up) and (at the request of the authorized user) to demonstrate the correct operation of the TSF.
- **FPT_TST.1.2:** The TSF shall provide authorised users with the capability to verify the integrity of parts of TSF data.
- **FPT_TST.1.3:** The TSF shall provide authorised users with the capability to verify the integrity of the TSF.

Application Note

The self-test at initial startup consists of a successful secure initialization of the TOE. The request for attestation data triggers self-tests for which the results are part of the attestation data.

FPT_TST.1/Attestation TSF testing only applies if the Device Attestation module is present in the TOE.

7.5.2 Security requirements rationale

The SFRs defined for device attestation have a direct relationship to fulfilling the security objectives of the TOE, where FCO_NRO.2 contributes to the objectives by designating a device-unique key for use in authentication and confidentiality of attestation data.

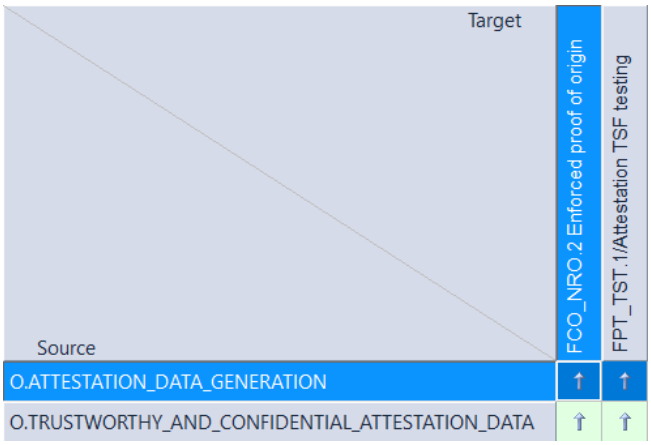

Figure 7-2 Security Requirements Rationale – Device Attestation

7.5.2.1 SFR dependencies

Requirements	CC Dependencies	Satisfied by
FCO_NRO.2	FIA_UID.1	FIA_UID.2 (Base PP)
FPT_TST.1	No dependencies	

FIA_UID.2 is hierarchical to FIA_UID.1, satisfying the dependency for FCO_NRO.2.

8 TOE summary specification

8.1 TOE summary specification - TEE base-PP

8.1.1.1 CA/TA Identification and Session Management

The TOE implements the execution processing within the TEE in a way that a proper binding to the CA-TA communication session is maintained and that the originating TA of internal calls is tracked, in order to support:

- Proper communication.
- To control access to resources.

This security service is provided by the Qualcomm TEE which implements:

- The task scheduling, TA loading and the proper call backs into the TAs as response to communication requests from the client applications.
- TA execution infrastructure, including session management.
- SMC interface that allows to trigger the communication requests from client applications.

Observe that the implementation of the GP client application API in the REE is not in the scope of the evaluation because it resided in the potentially compromised REE.

8.1.2 Confidentiality, integrity, and isolation (of runtime data in RAM and in transfer)

The TOE enforces the main execution separation mechanisms, that is:

- REE - TEE separation
- TEE OS - TA separation
- TA - TA separation

The separation is enforced by Qualcomm TEE in close collaboration with the ARM TrustZone aware hardware components, like the CPU, memory controllers, as well as memory management units etc.

In addition to this, the TOE enforces the confidentiality and integrity of data when transferred between different memories and processing components and when residing in RAM.

8.1.3 Cryptography

The TOE provide a wide range of cryptographic support functionality as listed in the FCS_COP requirements to support security implementation in TAs and internal security mechanisms.

In detail, the following crypto algorithms are evaluated:

- Key Derivation: NIST_KD_PRF (AES_CMAC, COUNTER_MODE)
- RSA: Key Generation (from 1024bit to 4096bit), RSA encryption and decryption according to PKCS1_V15 and OAEP, RSA signature generation accoding to PKCS1_V15 and FIPS_DSA, DHKE,
- ECC: Key Generation (from160bit to 512 bit in GF(p), signature generation according to the DSS standard, ECKA_DH
- Keyed/hashed message authentication (HMAC)
- AES (128 and 256bit in ECB, CBC, CTR, CBC_MAC,CMAC, GCM, XEX, and XTS mode of operation
- 2/3TDES in ECB and CBC_MAC mode of operation
- Hashing: SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, MD5

For all of these crypto mechanisms the TOE enforces strict access controls on TA keys.

The crypto algorithms are provided via the standard GP API and some of them also via a Qualcomm TEE proprietary API. The underlying hardware provides HW accelerators for the symmetric crypto operations which are also in the scope of the evaluation.

8.1.4 Initialization, operation, and firmware integrity

The TOE enforces secure initialization. OEMs can see *SM8250/SM8250P Secure Boot Enablement User Guide* (80-PK882-9) for more details.

Normal TAs are securely loaded from external memory during normal operation of the TOE when communication to them is initiated from the REE / the client applications.

During the initialization and runtime loading the TOE enforces the integrity of the firmware.Additionally, the TOE enforces that it always maintains a secure state by implementing TA isolation and sandboxing, supported by hardware enforced memory isolation and the Arm CPU feature Data Execution Prevention. The Qualcomm TEE further implements ASLR and stack canaries, which makes it harder to mount attacks on the memory layout and call stack structures. The detection mechanisms result in a fatal error, which causes a hard reset of the TOE. The hard reset ensures the TOE will never enter a state where it cannot guarantee the security function.

8.1.5 Random number generator

The TOE provides a SP800-90A compliant random number generator which meets the DRNG.3 requirements of [AIS31].

8.1.6 TEE identification

The TOE implements a TEE identification mechnanism according to the GP specifications which is in the scope of the evaluation. The TOE enforces that the TEE identifier is unique per device.

8.1.7 TEE instance time

The TOE provides a TEE instance time which acts as a monotonous counter.

8.1.8 Trusted storage

The TEE implements a trusted storage which allows for securely loading data form external memories while enforcing its integrity and confidentiality. This mechanisms is backed in hardware by a replay protected memory block (RPMB), which is also in the scope of the evaluation.

8.2 TOE summary specification - TEE Time and Rollback

8.2.1 Rollback protection

The TOE implements full rollback protection, so it enforces that the TOE cannot be downgraded to previous versions.

For the Qualcomm TEE this rollback protection is enforced directly in the secure boot flow.

All images loaded through Qualcomm TEE are also protected against rollback.

8.2.2 TA persistent time

In addition to the TEE instance time the TOE also implements a TA persistent time in form of a monotonous counter as specified by GP.

8.3 TOE summary specification - TEE Debug

The TOE implements a strong protection of all debugging features available on the TOE. The protections consist of strong authentication and control over the system behavior towards:

- Memory/crash dumping,
- Kernel, boot and JTAG logging,
- Debug enable and modes.

OEM can see *Debug Policy Version 2 User Guide* (80-NV396-72) for more details.

8.4 TOE summary specification - eSE Trusted Path

The TOE provides a mechanism to establish a trusted communications path to an embedded secure element. The embedded secure element is a physically external component. The communications path provides confidentiality, authenticity and integrity by means of the GlobalPlatform SCP11 protocol, relying on the base TEE security features for cryptographic key storage and algorithm implementations.

eSE Trusted Path is optional functionality that can be removed from the TOE with no impact to other security features, services or mechanisms of the TOE. By removing the functionality the security problem definition (*4.2, Security problem definition – eSE*), objectives (*6.2, Security objectives – eSE*) and requirements (*7.4, Security requirements – eSE trusted path*) related to the eSE Trusted Path functionality are no longer applicable to the TOE.

8.5 TOE summary specification - Device Attestation

The TOE implements a device attestation feature based on a combination of user provided information, tokens from the hardware environment and software integrity tests, fulfilling the SFRs FCO_NRO.2.1, FCO_NRO.2.2 and FPT_TST.1/Attestation. Device Attestation results are confidentiality protected by a signed public key of the service provider seeking the device attestation. The user provided information may include a nonce, a timestamp or unique number to prevent replay of attestation results. The attestation results are signed by a device key that allows the service provider to verify the authenticity of the results, fulfilling FCO_NRO.2.3..

Device Attestation is optional functionality and may be removed from the TOE with no impact to other security features, services or mechanisms of the TOE. When the functionality is removed from the TOE the security problem definition (*4.3, Security problem definition – Device Attestation*), objectives (*6.3, Security objectives – Device Attestation*) and requirements (*7.5, Security requirements – Device Attestation*) related to the device attestation functionality are no longer applicable to the TOE.

8.6 TOE summary specification rationale

8.6.1 TOE summary specification rationale - Base TEE Features

The TOE summary specification for the security features and security services which are related to the requirements of the TEE Base-PP and the TEE Time and Rollback as well as the TEE Debug module are structured in the same way than the SFR sub-grouping in the PP.

Therefore, the mapping between these security features and security services and the SFRs taken from the PP is straight-forward.

8.6.2 TOE summary specification rationale - eSE Trusted Path

The summary specification for the additional security feature eSE Trusted Path are directly related to the requirements grouped in the security requirements for eSE Trusted Path.

8.6.3 TOE summary specification rationale - Device Attestation

The summary specification for the additional security feature for Device Attestation are directly related to the requirements grouped in the security requirements for Device Attestation.

A References

A.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Qualcomm Trusted Execution Environment (TEE) Version 5.0 Reference Manual	80-NH537-4
Secure Coding Guidelines for TrustZone QTEE Applications	80-NK069-1
Sectools: SecImage Tool (Version 5.20 and Later) User Guide	80-NM248-8
Sectools: FuseBlower User Guide	80-NM248-3
Sectools: KeyProvision Tool User Guide	80-NM248-5
Debug Policy Version 2 User Guide	80-NV396-72
SM8250 Secure Boot Enablement	80-PK882-9
Provisioning Encryption Tool User Guide	80-P1824-1
SM8250 QFPROM Programming Reference Guide	80-PL546-97
SM8250 Security Overview	80-PK882-10
Security Quick Start	80-PF777-103
TEE-based Mobile Payment Security Guidelines for OEMs	80-NR875-15
Qualcomm Trusted Execution Environment v5.8 Functional Description for TEE Protection Profile Compliance Technical Overview	80-NR875-19
Standards	
Data Encryption Standard (DES)	FIPS PUB 46-3
Secure Hash Standard (SHS)	FIPS PUB 180-4
Digital Signature Standard (DSS)	FIPS PUB 186-4
Specification for the ADVANCED ENCRYPTION STANDARD (AES)	FIPS PUB 197
The Keyed-Hash Message Authentication Code (HMAC)	FIPS PUB 198-1
GlobalPlatformDevice Technology TEE Internal API Specification, Version 1.1	GPD_SPE_010
GlobalPlatform Device Committee TEE Protection Profile V1.2.1	GPD_SPE_021
Recommendation for Block Cipher Modes of Operation, Methods and Techniques	NIST SP 800-38A
Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication	NIST SP 800-38B
Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC	NIST SP 800-38D
Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices	NIST SP 800-38E
Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography	NIST SP 800-56A

Title	Number
Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher	NIST SP 800-67
Recommendation for Key Derivation Using Pseudorandom Functions	NIST SP 800-108
The MD5 Message-Digest Algorithm	RFC 1321
Diffie-Hellman Key Agreement Method	RFC 2631
Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation	RFC 5639
PKCS #1: RSA Cryptography Standard, Version 2.2	RSA Laboratories
Resources	
ISO 9001:2008 – Certificate Number: 110371.001; valid as of Jul 25, 2014	DEKRA Certification Group

A.2 Acronyms and terms

Acronym or term	Definition
AES	Advanced Encryption Standard
API	Application programming interfaces
BSP	Board support package
CA	Client application
CC	Claims conformance
DRAM	Dynamic random-access memory
DRM	Digital rights management
DRNG	Deterministic random number generator
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
eSE	Embedded secure element
GCM	Galois/Counter Mode
IoT	Internet of things
LOQ	Letter of Qualification
PP	Protection profile
PoP	Package-on-package
Qualcomm HEE	Qualcomm Hypervisor Execution Environment
Qualcomm TEE	Qualcomm Trusted Execution Environment
REE	Rich Execution Environment
RoT	Toot of trust
RPMB	Replay protected memory block
SFP	Security function policy
SFS	Secure file system
SFR	Security functional requirement
SMC	Secure monitor call
SMMU	System memory management units

Acronym or term	Definition
SoC	System-on-chip
TA	Trusted application
TEE	Trusted Execution Environment
TOE	Target of evaluation
TRNG	True random number generation
TSF	TOE security function
UEFI	Unified extensible firmware interface