Huawei HSM 2.0 Management Firmware

Security Target

HUAWEI

Huawei Technologies Co., Ltd

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People’s Republic of China

Website: http://www.huawei.com

Change History

Date	Version	Description
2022-05-10	V1.0	Draft ST
2022-06-10	V1.1	Add HDIM, secure Upgrade
2022-07-18	V1.2	Specify the size of the crypto
2022-08-23	V1.3	Process evaluator comments
2022-11-08	V1.4	Process evaluator comments, add CPU and so on
2023-01-16	V1.5	1. Add environment-protection assumptions: hardware environment provides external storage access control protection mechanisms. 2. Add environment-protection assumptions: SOC which the TOE integrated into is used on network devices such as firewalls and routers,the working environment is a physically secure environment. 3. Add description for FPT_TEE.1/Upgrade in section 7.5 SF.Upgrade. 4. Update the description “PKCS#1” and “verification” in FCS_COP.1.1. 5. Update the external memory description, delete the “secure” description and add the SFC description in section 3.4 and section 4.2 6. In FAU_GEN.1.1, delete the description “Start-up and shutdown of the audit functions; b) All auditable events for the not specified level of audit; and”
2023-02-16	V1.6	1. Update environment-protection assumptions, remove the detail SFC spec in the ST and add in Hardware-Support that “the BaseBIOS shall configure the SFC to ensure the HSM’s Flash area can only be accessed by the HSM hardware.” 2. Change to italic font “PKCS#1” and “and verification” in FCS_COP.1.1. 3. Update tracing description, remove the FMT_SMR.1 in O.Cryptographic-Service, O.Key-Management-Service and O.Secure-Storage in 6.2.1.2 Tracing. 4. Add a section for SF.Audit 5. Update the section 3.1 Assets, remove “generated within the HSM” and update “Measurement value of software running

		outside of the TOE”. 6． Update the P.HUDI description in section 3.3. 7． In FAU_GEN.1.1, add the description “Start-up and shutdown of the audit functions; b) All auditable events for the not specified level of audit; and”
2023-02-20	V1.7	1． Update the RSA signature generation and verify into “*digest encoding/decoding*”, encryption/decryption into “message *encoding/decoding*”; 2． Add section 5.1 for Security audit data generation (FAU_SAG); 3． Replace FAU_GEN.1 with FAU_SAG.1, and update the 6.1.2.9 Audit description according to the section 5.1. 4． Update the FMT_MSA.3 Application note description. 5． Update FDP_RIP.1.1 description to remove “all keys”
2023-03-02	V1.8	1． Update the description of Statement of Security Requirements in section 6.1. 2． Update the “HashMaster” description in A.Hardware-Support and OE.Hardware-Support. 3． Update the version number of Guidance document to V1.5.
2023-05-15	V1.9	1． Add SFC/PMP/Hashmaster description in Abbreviations. 2． Update the host CPU description in section1.2.3. 3． Update the SF.HRA description. 4． Update hardware assumptions in section 3.4. 5. Update the version number of Guidance document.
2023-08-24	V2.0	1、update the TOE name to HSM Management Firmware

Content

Huawei HSM 2.0 Management firmware Security Target ... 1
1 Introduction ... 9
1.1 ST Reference and TOE Reference ... 9
1.1.1 ST Reference ... 9
1.1.2 TOE Reference ... 9
1.2 TOE Overview ... 9
1.2.1 TOE Type ... 9
1.2.2 TOE Usage and Major Security Features ... 10
1.2.3 Required non-TOE hardware/software/firmware ... 10
1.3 TOE Description ... 11
1.3.1 Physical Scope of TOE ... 11
1.3.2 Logical Scope of TOE ... 12
1.3.3 TOE Life Cycle ... 13
2 Conformance Claims ... 14
2.1 CC Conformance Claim ... 14
2.2 PP Conformance Claim ... 14
2.3 Conformance Claim Rationale ... 14
3 Security Problem Definition ... 15
3.1 Assets ... 15
3.2 Threats ... 15
3.3 Organizational security policies ... 16
3.4 Assumptions ... 16
4 Security Objectives ... 19
4.1 Security Objectives for the TOE ... 19
4.2 Security Objectives for the Operational Environment ... 20
4.3 Security Objective Rationale ... 21
4.3.1 Overview ... 22
4.3.2 Threats ... 22
4.3.3 Organizational Security Policies ... 23
4.3.4 Assumptions ... 25
5 Extended Components Definition ... 26
5.1 Security audit data generation (FAU_SAG) ... 26

6 Security Requirements 27
6.1 Statement of Security Requirements 27
6.1.1 Notation 27
6.1.2 Security Functional Requirements 27
6.1.3 Security Assurance Requirements 41
6.2 Security Requirements Rationale 41
6.2.1 Security Functional Requirements 41
6.2.2 Security Assurance Requirements 47
7 TOE summary specification 49
7.1 SF.Crypto 49
7.2 SF.KM 49
7.3 SF.HDIM 49
7.4 SF.HUDI 49
7.5 SF.Upgrade 50
7.6 SF.CCM 50
7.7 SF.Storage 50
7.8 SF.HRA 50
7.9 SF.Audit 50
8 Bibliography 51
8.1 Evaluation Documents 51
8.2 Developer Documents 51
8.3 Other Documents 51

List of tables

Table 1 Components of the TOE scope 12

Table 2 Actors in the TOE Life Cycle 13

Table 3 Tracing from Security Objectives to Threats 22

Table 4 Tracing from Security Objectives to OSPs 24

Table 5 Tracing from Security Objectives to Assumptions 25

Table 6 Security Functional Requirements dependencies....41

Table 7 Tracing from Security Functional Requirements to Security Objectives 44

Table 8 Security Assurance Requirements dependencies for augmentations 47

Table of Figures

Figure 1 HSM and the TOE 10

Figure 2 TOE and Operational Environment 10

Abbreviations

Abbreviation	Full Name	Description
HSM	Hardware security module	Hardware security module is a subsystem of SOC that provide security functions, include the HSM module in the SoC and HSM firmware
TEE	Trusted Execution Environment	
BSBC	Boot-rom Secure Boot Code	The chiprom boot code.
BaseBIOS	Base Basic Input Output System	
AP	Application Process	Application Process in host CPU
AP BIOS	Application Process Basic Input Output System	Basic Input Output System running in host CPU
HSM-Agent	Hardware Security Module Agent	HSM driver which run on TEE OS outside HSM
CCM	Communication Core Module	
ST	Security Target	
TOE	Target Of Evaluation	
eFUSE	Electrically Programmable Fuse	
HUDI	Huawei Unique device Identity	
HDIM	Huawei Dynamic integrity measurement	
HRA	Huawei Remote Attestation	
PMP	Physical Memory Protection	Provides memory protection function in RISCV CPUs
SFC	SPI Flash Controller	
Hashmaster	Hash hardware engine with AXI master	Provides access to external memory and calculates its hash value

1 Introduction

This introduction chapter contains the following sections:

1.1 ST Reference and TOE Reference

1.2 TOE Overview

1.3 TOE Description

1.1 ST Reference and TOE Reference

1.1.1 ST Reference

Title:	Huawei HSM 2.0 management firmware security Target
Author:	Huawei Technologies Co., Ltd.
Version:	2.0
CC Version:	Version 3.1, Revision 5
Assurance Level:	EAL4+
ITSEF:	Brightsight
Certification Body:	Netherlands Scheme for Certification in the Area of IT Security (NSCIB)

1.1.2 TOE Reference

Name:	HSM 2.0 management firmware
Developers:	Huawei Technologies Co., Ltd.
Version:	B006

1.2 TOE Overview

1.2.1 TOE Type

The TOE is management firmware that runs on a dedicated hardware security module (called HSM) in a SoC after the booting of the system. It provides secure services to the host CPU in the SoC. HSM is a secure module in SoC, that is isolated from the remaining SOC components based on physical and/or logical isolation mechanisms, the SOC is used on network devices such as firewalls and routers,the working environment is a physically secure environment, i.e., no unauthorized person has physical access to the device. The HSM relies on external memory to store content (data, code). The whole HSM solution consist of an HSM module in the SoC, the HSM boot firmware (including BSBC and BaseBIOS), and the HSM management firmware as shown in Figure 1, the TOE in this document refers to HSM management firmware , which consist of

1) HSM management firmware

2) HSM management firmware guidance documents

The HSM management firmware is running on the HSM platform. The HSM management firmware guidance documentation provides the necessary information for secure usage of the TOE by customers and users.

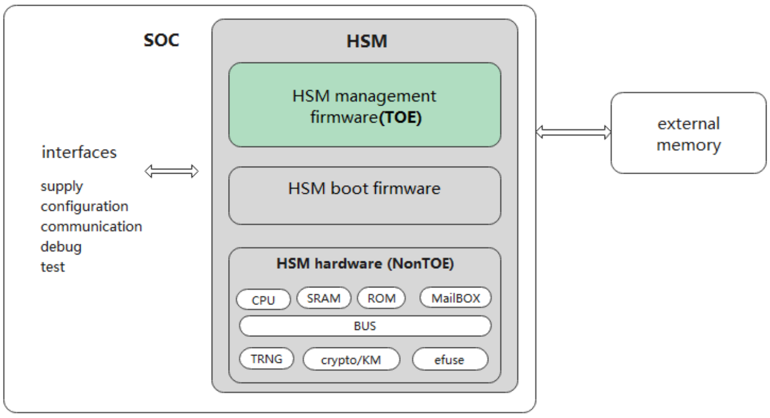


Figure 1 HSM and the TOE

1.2.2 TOE Usage and Major Security Features

The TOE is used by integrating it with the two layers of boot firmware running on a compatible hardware platform.

As part of the management secure service of this hardware and software stack, the TOE will be used to provide secure services. In order to support this, the TOE provides the following security features:

- SF.Crypto Cryptographic Service
- SF.KM Key Management Service
- SF.HDIM Dynamic Measurement
- SF.HUDI Device Unique Identity
- SF.Upgrade Secure Upgrade
- SF.CCM Communication Core Module
- SF.HRA remote Attesstation
- SF.Storage Secure storage
- SF.Audit Audit logging

1.2.3 Required non-TOE hardware/software/firmware

The TOE is located in HSM (a hardware security module inside an SoC). The TOE is securely loaded by a chain of bootloaders and runs in the SRAM of the HSM hardware. Initially, the encrypted TOE is stored in the Flash, which is an external component that is not part of the same hardware that contains the

HSM. The second layer bootloader decrypts and authenticates the TOE and stores it in SRAM, where it will run.

The TOE relies on HSM hardware, HSM boot Firmware, and external memory which are not part of the HSM, but they are still part of the same hardware that comprises the operational environment of the TOE. Figure 2 shows the TOE in its operational environment. The following summarises the role of the various components:

- **HSM hardware:** including dedicated SRAM, BootROM, TRNG, Crypto and KM, Hardware line used for key transmit protection between HSM and HostCPU, efuse(OTP), CPU with PMP protection, deciated Mailbox that used to communicate with TEE, hashmaster that used to retrieve the content of the external memory and calculate the hash value of the content for HDIM service, SFC that used to store the HSM content into External NVM.
- **External memory**: Including External NVM (via SPI Flash Controller for storing the HSM content, i.e. persistent data, code) and DDR RAM (via Hashmaster).
- **HSM boot firmware**: verifies the TOE before running on the HSM platform.
- **Host CPU**: the remaining component of the SOC, which includes trusted AP BIOS, TEE OS and HSM-Agent, to provide the interface for using HSM secure services.

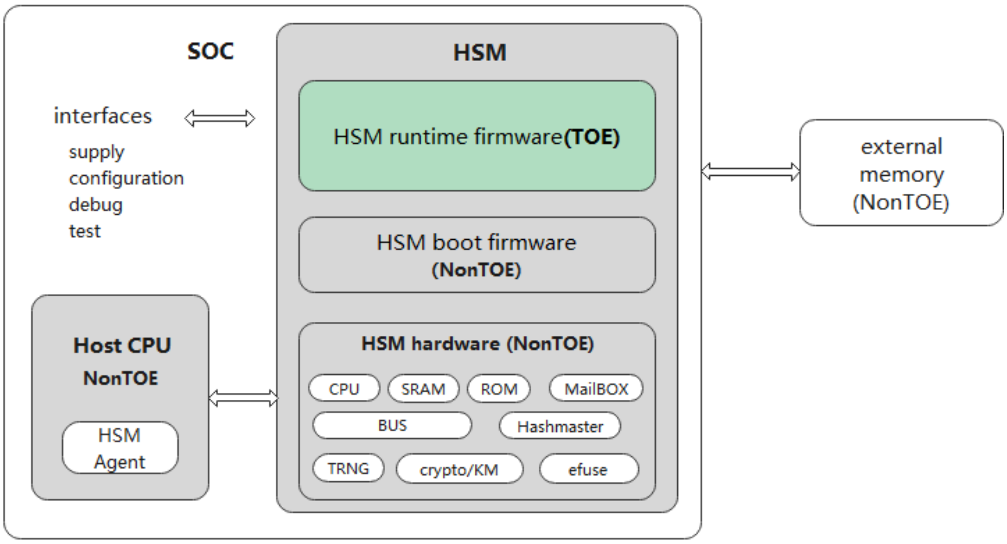


Figure 2 TOE and Operational Environment

1.3 TOE Description

1.3.1 Physical Scope of TOE

Table 1 lists the TOE components, which includes the component shown in Figure 2, as well as the guidance:

Table 1 Components of the TOE scope

Type	Component	Version	Format	Delivery method
Software	Huawei HSM 2.0 management firmware	B006	Binary(.bin)	Encrypted file download
Guidance document	Huawei HSM 2.0 management firmware Guidance Document	V1.7	Document (.pdf)	GPG encrypted file download
	Huawei HSM 2.0 management firmware service command specification	V1.6	Document (.pdf)	GPG encrypted file download

1.3.2 Logical Scope of TOE

The logical scope of the TOE consists of the following security features:

- SF.Crypto, which implements digest/message encoding/decoding and access control to cryptographic services.
- SF.KM, which implements key management.
- SF.HDIM, which implements a mechanism for dynamic measurement of the external memory contents
- SF.HUDI, which implements process to communicate device identity information stored in the hardware environment to the requesting user
- SF.Upgrade, which implements a secure upgrade process to update the second-stage bootloader image or the TOE image.
- SF.CCM, which implements the communication with the user for securely accessing the HSM service and access control based on role-based authentication.
- SF.HRA, HRA remote attestation service performs device measurement during system boot up and run-time which enables the remote attestation server to obtain the status of the devices. HRA Remote attestation involved three different phases: evidence creation phase, evidence transmission phase and evidence evaluation phase. During evidence creation phase, the TOE performs device measurement and provides the measurement results to the remote attestation server for measurement provisioning and device status obtaining. During evidence transmission phase, the TOE performs signature protection for the measurement results (The evidence transmission is implemented by the AP side). The evidence evaluation is implemented by the remote management server .SF.Storage, which implements an access control mechanism on non-persistent objects.
- SF.Audit, which implements an audit mechanism.

1.3.3 TOE Life Cycle

The life cycle of the TOE is divided into the following three phases, but only phase 1 to phase 3 are related to the TOE, will be audited in the TOE, others will be audited in the composite product certification.

- Phase 1 corresponds to the design of the TOE
- Phase 2 corresponds to the development and the testing of the TOE
- Phase 3 corresponds to the delivery of the TOE
- Phase 4 coresponds to the loading of the TOE into a target hardware platform,like HSM .
- Phase 5 The whole HSM(hardware and firmware) is installed on the manufacture platform.
- Phase 6 the whole HSM is prepared for use,deliverty to the end user.
- Phase 7: operation and usage.

Table 2 shows the whole life cycle of the TOE.

Table 2 Actors in the TOE Life Cycle

Phase	Actors
1：the design of the TOE	The TOE developer：design the architecture of the TOE
2：the develop and the test of the TOE	The TOE developer: develop and test of the TOE
3：the delivery of the TOE	The TOE developer: write the usage guidelines documents，encrypt and package the TOE
4: loading of the TOE	Loading the encrypted binary of the TOE to a qualified HSM hardware with external flash.
5: install of the HSM	Install the HSM(hardware and firmware) to the hardware board ,and platform specific keys and certificates are set
6:to the end user of the HSM	HSM is prepared for operational usage, used in the end user environment

2 Conformance Claims

This chapter is divided into the following sections:

2.1 CC Conformance Claim

2.2 PP Conformance Claim

2.3 Conformance Claim Rationale

2.1 CC Conformance Claim

This Security Target conforms to CC Part 2 extended and Part 3 conformant, with a claimed Evaluation Assurance Level of EAL 4, augmented by ALC_DVS.2 and AVA_VAN.5.

This Security Target claims conformance to the following specifications：

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Function Requirements; Version 3.1, Revision 5, April 2017 (Extended)
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 5, April 2017 (Conformant)
- Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5, April 2017

2.2 PP Conformance Claim

This Security Target does not claim conformance to any PP.

2.3 Conformance Claim Rationale

This is not applicable to this Security Target, because it does not claim conformance to any PP.

3 Security Problem Definition

This chapter lists the assets, threats, organizational security policies and assumptions (as defined in CC Part 1) that are relevant for the TOE of this Security Target.

3.1 Assets

The assets of the TOE are divided into the following:

- Cryptographic keys, which can be
  - stored in OTP,
  - imported to HSM, or
  - derived within the HSM.
- Role authentication credentials.
- PCR(platform configuration register) intended to record the measurement digests to be used for HRA.
- Device identity.
- Second-stage bootloader and TOE software image.
- Measurement value of software running outside of the TOE.
- Secure services provided by the TOE.

3.2 Threats

The threats of the TOE are divided into the following:

T.Key-Confidentiality

An unauthorized external entity discloses the user cryptographic keys (or keys that are derived from such keys) through logical means, including but not limited to disclose the keys while stored in external NVM.

T.Key-Integrity

An unauthorized external entity manipulate the user cryptographic keys (or keys that are derived from such keys) through logical means, including but not limited to manipulate the keys while stored in external NVM and.

T.Credential-Confidentiality

An unauthorized external entity discloses the Role authentication credential through logical means, including but not limited to disclose the Role authentication credential while stored in external NVM.

T.Credential-Integrity

An unauthorized external entity manipulate the Role authentication credential through logical means, including but not limited to manipulate Role authentication credential while stored in external NVM.

T.Image-Confidentiality

An unauthorized external entity discloses the second-stage bootloader or TOE upgrade image through logical means, including but not limited to abuse of debug functionality. The entity then uses the knowledge of the implementation to mount attacks on all user assets.

T.Image-Integrity

An unauthorized external entity upgrades unauthentic second-stage bootloader or TOE image through logical means, including but not limited to changing the image while stored in external NVM .The entity then uses the unauthentic software to mount attacks on all user assets.

T.Service-Abuse

An unauthorized external entity access to security services provided by HSM.

3.3 Organizational security policies

P.HDIM

The software stored in memory outside of the TOE at various levels shall be measured and the measurement value shall be compared to a reference value.

P.HUDI

The TOE shall enable the user to export device identity information, which includes individual products and their different sub-components (chips, boards, devices). Furthermore, the identities of sub-components can be composed into the identities of the components containing these sub-components.

P.HRA

The TOE provide response of device measurement status request, reporting the boot and run-time measurement results to the AP of the SoC and then the off-chip remote attestation serve via a secure way.

P.Audit

Security-relevant actions taken by the TOE shall be auditable by authorized external users.

P.Crypto-Service

The TOE provides secure hardware-based cryptographic services to the user.

3.4 Assumptions

This section states the assumptions that hold on the TOE operational environment. These assumptions have to be satisfied.

A.Environment-Protection

It is assumed that the SOC which the TOE integrated into is used on network devices such as firewalls and routers,the working environment is a physically secure environment, i.e., no unauthorized person has physical access to the device.

It is assumed that the TOE is integrated into a CC EAL4+ ALC_DVS.2 and AVA_VAN.5 certified HSM hardware security module and secure boot firmware in a SoC. The HSM hardware security module (SoC) shall protect the TOE against:

- Physical attacks, such as probing or physical manipulation.
- Malfunction attacks, where environmental stress is applied to cause a malfunction.
- Side-channel attacks, where sensitive information leaks as a result of leakage.

It is assumed that HSM-agent, TEE OS and AP BIOS outside of the TOE are trusted.

A.Hardware-Support

It is assumed that the TOE is integrated into a CC EAL4+ ALC_DVS.2 and AVA_VAN.5 certified HSM hardware security module and secure boot firmware in a SoC. The HSM hardware security module that provides (at a minimum) the following functionality:

- One-Time Programmable (OTP) memory. The Hardware Key shall be included and it shall only be accessible by the by the HSM hardware.
- External NVM and SFC are used to store the HSM content (persistent data,code), the BaseBIOS shall configure the SFC to ensure the HSM’s External NVM Flash area can only be accessed by the HSM.
- CPU and internal SRAM with PMP protection.
- HashMaster that is used to retrieve the content of the external memory and calculate the hash value of the content for HDIM service or exchange the data for encryption/decryption or load the upgrade image.
- Dedicated Mailbox that is used to communicate with TOE.
- A bootloader that securely loads the TOE image.
- Cryptographic functionality, which includes
    - AES CTR/CBC/ECB as defined by [SP800-38A] and GCM as defined by [SP800-38D].
    - SHA-256 and SHA-512 as defined by [FIPS180-4],
    - HMAC-SHA-256 and HMAC-SHA-512 as defined by [FIPS198-1],
    - PBKDF2 as defined by [SP800-132].
    - True Random Number Generator as defined by FIPS140-2/SP800-90A/SP800-90B/AIS31 PTG.2/3.
    - Large number operation (multiplication, as well as modular addition, modular subtraction, modular multiplication, modular inverse, and modular exponentiation, with bit size 2048/3072/4096).
    - RSA/ECC key generation(RSA key size with 2048/3072/4096 bits, ECC base on NIST or

brainpool curves , key size with 256, 384, 512[brainpool]/521[NIST]).

- Elliptic curve operations (point multiplication and point addition for prime order curves of bit sizes 256, 384, 512, and 521).

A.Process-Security

It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). Specifically, the second-stage bootloader and HSM firmware images to be upgraded are assumed to be stored in the external NVM in a way that preserves their confidentiality and supports the authentication of the images by TOE, as well as the automated recovery.

A.Process-User-Data

All data that is considered user data from the point of view of this TOE needs to be treated according to its security needs by the user when handled outside of the TOE. The role authentication credentials on the TOE user side are assumed to be handled securely by TOE user. Additionally, user keys are assumed to be handled securely when outside of the control of the TOE.

4 Security Objectives

This chapter contains the following sections:

4.1 Security Objectives for the TOE

4.2 Security Objectives for the Operational Environment

4.3 Security Objective Rationale

4.1 Security Objectives for the TOE

The TOE shall provide the following security objectives:

O.Cryptographic-Service

The TOE shall provide access to cryptographic services (some of which are provided by the hardware environment as described in the following section) and restrict the access to cryptographic services based on user roles.

O.Key-Management-Service

The TOE shall restrict the import and export of cryptographic keys based on pre-defined and user-defined policies.

O.HDIM

The TOE shall perform dynamic measurements on the external SoC memory and compare it to reference values.

O.HUDI

The TOE shall enable the user to export device identity information, which includes individual products and their different sub-components (chips, boards, devices). Furthermore, the identities of sub-components can be composed into the identities of the components containing these sub-components.

O.HRA

The TOE shall provide the software register PCR and control the access and operations of PCR. Enable the user to store the sensitive information (such as measurement results) of the TOE.

O.Upgrade

The TOE shall authenticate the software image to be upgraded (with the support of the hardware platform as described in the following section). Before the upgrade process, the TOE will verify that recovery using a back-up image is possible. In case any failure occurs during the authentication of the image to be upgraded, the TOE shall cancel the upgrade. All sensitive data will be cleared after the operation.

O.Communication

The TOE shall ensure that incoming and outgoing communication data satisfies pre-defined information flow control rules to help prevent unauthorized modification and disclosure. Additionally, the TOE shall implement role-based service access control to prevent unauthorized service access.

O.Secure-Storage

The TOE shall control the access to non-persistent objects in memory used by the TSF.

O.Audit

The TOE shall create audit logs for specific security events including service requests, exceptions, user role accesses, user data accesses, and tests.

4.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment are specified according to the CC.

OE.Environmental-Protection

The SOC which the TOE integrated into is used on network devices such as firewalls and routers. Its working environment shall be a physically secure environment, i.e., no unauthorized person has physical access to the device.

The hardware environment that the TOE is integrated into a CC EAL4+ ALC_DVS.2 and AVA_VAN.5 certified HSM hardware security module and secure boot firmware in a SoC. The HSM hardware security module shall protect the TOE against:

The hardware environment that the TOE is integrated into shall protect against the following attacks:

- Physical attacks, such as probing or physical manipulation
- Malfunctions, caused by deliberate environmental stress applied by an attacker
- Side-channel attacks, where sensitive information leaks through operational parameters

HSM-Agent, TEE OS and AP BIOS outside of the TOE shall be trusted.

OE.Hardware-Support

The hardware and firmware environment that the TOE is integrated into shall provide (at a minimum) the following functionality:

- One-Time Programmable (OTP) memory.
- External NVM and SFC are used to store the HSM content (persistent data,code), the BaseBIOS shall configure the SFC to ensure the HSM’s External NVM Flash area can only be accessed by the HSM.
- CPU with PMP protection.
- HashMaster that uses to retrieve the content of the external memory and calculate the hash value of the content for HDIM service.
- Mailbox that used to communicate with TOE.
- A bootloader that securely loads the TOE image.

- Cryptographic functionality, which includes
    - AES CTR/CBC/ECB as defined by [SP800-38A] and GCM as defined by [SP800-38D].
    - SHA-256 and SHA-512 as defined by [FIPS180-4],
    - HMAC-SHA-256 and HMAC-SHA-512 as defined by [FIPS198-1],
    - PBKDF2 as defined by [SP800-132].
    - True Random Number Generator as defined by FIPS140-2/SP800-90A/SP800-90B/AIS31 PTG.2/3.
    - Large number operation (multiplication, as well as modular addition, modular subtraction, modular multiplication, modular inverse, and modular exponentiation, with bit size 2048/3072/4096).
    - RSA/ECC key generation(RSA key size with 2048/3072/4096 bits, ECC base on NIST or brainpool curves , key size with 256, 384, 512[brainpool]/521[NIST]).
    - Elliptic curve operations (point multiplication and point addition for prime order curves of bit sizes 256, 384, 512, and 521).

OE.Process-Security

Security procedures shall be used after TOE Delivery up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).

Specifically, the bootloader and HSM firmware image to be upgraded shall be encrypted using AES-256-GCM (as defined by [SP800-38D]) and shall be packaged together with a digital signature according to RSASSA-PSS (as defined by [PKCS#1]) with key size 4096 bits and hash function SHA-256 or SHA-512 (as defined by [FIPS180-4]), or according to ECDSA (as defined by [FIPS186-4]) based on the NIST P256/P384/P521 or brainpool256r1/ brainpool384r1/brainpool512r1 curves with SHA-256/ SHA-384/SHA-512 (as defined by [FIPS180-4]). The digital signature(s) shall be made with a key corresponding to a value stored in OTP. Additionally, the redundant authentication configuration and a counter corresponding to the image version shall be stored in OTP, and a recovery copy of the user software to be loaded shall be stored in the external NVM to enable the automated recovery function. Finally, all associated keys shall be handled securely when outside of the control of the TOE.

OE.Process-User-Data

All data that is considered user data from the point of view of this TOE shall be treated according to its security needs by the user when handled outside of the TOE. Specifically, the role authentication credentials on the TOE user side are assumed to be handled securely by the TOE user when outside of the control of the TOE. Additionally, user keys shall be handled securely when outside of the control of the TOE.

4.3 Security Objective Rationale

4.3.1 Overview

4.3.2 Threats

Table 3 shows an overview of the tracing from objectives to threats. Subsequently the tracing is explained in more detail.

Table 3 Tracing from Security Objectives to Threats

Security Objectives Security Problem Definition	O.Cryptographic-Service	O.Key-Management-Service	O.Upgrade	O.Communication	O.Storage	O.Audit	OE.Environmental-Protection	OE.Hardware-Support	OE.Process-Security	OE.Process-User-Data
T.Key-Confidentiality	√	√		√	√		√	√		√
T.Key-Integrity	√	√		√	√		√	√		√
T.Credential-Confidentiality				√	√		√	√		√
T.Credential-Integrity				√	√		√	√		√
T.Image-Confidentiality			√	√	√		√	√	√	
T.Image-Integrity	√		√	√	√		√	√	√	
T.Service-Abuse				√	√		√	√		

In general, all threats are countered in part by **O.Communication**, **O.Storage**. **O.Communication** implements the role-based service access control that prevents unauthorized entities from accessing assets through services. **O.Storage** implements access control on non-persistent objects in memory stored by the TSF, which ensures that all services operate correctly without interference on non-persistent objects at runtime. **OE.Environmental-Protection** contributes to the enforcement of all threats, as it protects all associated services and features from physical attacks, malfunctions, and side-channel attacks. In addition, it ensures that HSM-Agent, TEE OS and AP BIOS are trusted.

The threats **T.Key-Confidentiality** and **T.Key-Integrity** are further countered by **O.Cryptographic-Service (**restrict the access to cryptographic services and keys), **O.Key-Management-Service** (restrict the import and export of keys), **OE.Hardware-Support** (relying on the OTP memory and cryptographic functionality provided by hardware), and **OE.Process-User-Data** (processing of the keys when not in the control of the TOE).

The threats **T.Credential-Confidentiality** and **T.Credential-Integrity** are further countered by **OE.Hardware-Support** (as the credential authentication procedure relies on the cryptographic functionality provided by hardware) and **OE.Process-User-Data** (processing of the credentials outside of the TOE).

The threat **T.Image-Confidentiality** is further countered by **O.Upgrade** (which ensures that the update images are authenticated and decrypted, that all failures are handled correctly, and that all sensitive data are cleared after the operation) supported by **OE.Hardware-Support** (as the cryptographic functionality required for image decryption is implemented in the hardware, and the required verification keys are stored in OTP or derived from keys stored in OTP). Finally, it is countered by **OE.Process-Security** (because this covers the handling of the upgrade images outside of the TOE and ensures the availability of recovery images).

The threat **T.Image-Integrity** is countered by the same objectives as **T.Image-Confidentiality** described above and additionally by **O.Cryptographic-Service**, as it provides signature decoding for one of the allowed signature decoding methods.

The threat **T.Service-Abuse** is further countered by **OE.Hardware-Support** (which supports the user role authentication with cryptographic functionality).

4.3.3 Organizational Security Policies

Table 4 shows an overview of the tracing from objectives to OSPs. Subsequently the tracing is explained in more detail.

Table 4 Tracing from Security Objectives to OSPs

Security Objectives / Security Problem Definition	O.Cryptographic-Service	O.HDIM	O.HUDI	O.HRA	O.Audit	OE.Environmental-Protection	OE.Hardware-Support
P.Crypto-Service	√					√	√
P.HDIM		√				√	√
P.HUDI			√			√	√
P.Audit					√	√	
P.HRA				√		√	√

OE.Environmental-Protection contributes to the enforcement of all OSPs, as it protects all associated services and features from physical attacks, malfunctions, and side-channel attacks.

P.Crypto-Service is further enforced by **O.Cryptographic-Service** (which controls the access to the cryptographic services), supported by **OE.Hardware-Support** (which provides the cryptographic functionality used in the services).

P.HDIM is further enforced by **O.HDIM** (which provides the measurement feature to the user), supported by **OE.Hardware-Support** (which provides the cryptographic functionality used in the measurements).

P.HRA is further enforced by **O.HRA** (which provide the software register PCR and control the access and operations of PCR. Enable the user to store the sensitive information (such as measurement results) of the TOE), supported by **OE.Hardware-Support**(which provides the cryptographic functionality used in the measurements).

P.HUDI is further enforced by **O.HUDI** (which controls the flow of the identity data transmitted between the hardware and the user), supported by **OE.Hardware-Support** (which provides the OTP that stores chip identity information, and the cryptographic functionality used for composing the identities).

P.Audit is further enforced by **O.Audit** (which provides the audit services).

4.3.4 Assumptions

Table 5 shows an overview of the tracing from objectives to assumptions. Subsequently the tracing is explained in more detail.

Table 5 Tracing from Security Objectives to Assumptions

Security Objectives / **Security Problem Definition**	**OE.Environmental-Protection**	**OE.Hardware-Support**	**OE.Process-Security**	**OE.Process-User-Data**
A.Environmental-Protection	√			
A.Hardware-Support		√		
A.Process-Security			√	
A.Process-User-Data				√

The assumption **A.Environmental-Protection** is directly met by its corresponding objective for the environment **OE.Environmental-Protection**.

The assumption **A.Hardware-Support** is directly met by its corresponding objective for the environment **OE.Hardware-Support**.

The assumption **A.Process-Security** is directly met by its corresponding objective for the environment **OE.Process-Security**.

The assumption **A.Process-User-Data** is directly met by its corresponding objective for the environment **OE.Process-User-Data**.

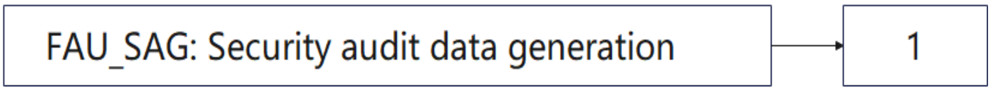
5 Extended Components Definition

5.1 Security audit data generation (FAU_SAG)

Family Behaviour

This family defines requirements for recording the occurrence of security relevant events that take place under TSF control. This family identifies the level of auditing, enumerates the types of events that shall be auditable by the TSF, and identifies the minimum set of audit-related information that should be provided within various audit record types.

Component levelling



FAU_SAG.1 Audit data generation defines the level of auditable events, and specifies the list of data that shall be recorded in each record.

Management: **FAU_SAG.1**

There are no management activities foreseen.

Audit: FAU_SAG.1

There are no auditable events foreseen.

FAU_SAG.1 Audit data generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FAU_SAG.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) All auditable events for the [selection, choose one of: minimum, basic, detailed, not specified] level of audit; and

b) [assignment: other specifically defined auditable events].

FAU_SAG.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: other audit relevant information].

6 Security Requirements

This part of the Security Target defines the detailed security requirements that shall be satisfied by the TOE. The statement of TOE security requirements shall define the functional and assurance security requirements that the TOE needs to satisfy in order to meet the security objectives for the TOE. This chapter contains the following sections:

6.1 Statement of Security Requirements

6.2 Security Requirements Rationale

6.1 Statement of Security Requirements

6.1.1 Notation

The CC allows several operations to be performed on security requirements (on the component level). Refinement, selection, assignment and iteration are defined in paragraph 8.1 of Part 1 of the CC.

The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Performed assignments are denoted by *italic text*.

The selection operation is used to select one or more options provided by the CC in stating a requirement. Performed selections are denoted by *underlined italic text*.

The iteration operation is used to distinguish components that are used multiple times in different context. Performed iterations are denoted by a forward slash (/) followed by a label after the component identifier.

The refinement operation is used to add details to requirements. Performed refinements are denoted by **bold text** when text is added and ~~**bold strikethrough text**~~ when text is removed.

6.1.2 Security Functional Requirements

6.1.2.1 Cryptographic Service

FCS_COP.1/RSASignPadding Cryptographic operation (RSA digest encoding/decoding)

FCS_COP.1.1/RSASignPadding The TSF shall perform *digest encoding/decoding* in accordance with a specified cryptographic algorithm *EMSA-PSS-ENCODE and EMSA-PSS-VERIFY* and cryptographic key sizes *2048, 3072, and 4096 bit* that meet the following: *[PKCS#1]*.

FCS_COP.1/RSAEdPadding Cryptographic operation (RSA message encoding/decoding)

FCS_COP.1.1/RSAEdPadding The TSF shall perform *message encoding/decoding* in accordance with a specified cryptographic algorithm *EME-OAEP encoding and decoding* and cryptographic key sizes *2048, 3072, and 4096 bit* that meet the following: [*PKCS#1*]

FDP_ACC.1/Crypto Subset access control

FDP_ACC.1.1/Crypto The TSF shall enforce the *Cryptographic Service SFP* on

Subjects:

- *User*

Objects:

- *Cryptographic key*

Operations:

- *Cryptographic operation.*
- *Delete key.*

FDP_ACF.1/Crypto Security attribute based access control

FDP_ACF.1.1/Crypto The TSF shall enforce the *Cryptographic Service SFP* to objects based on the following:

- *User: role*
- *Cryptographic key: usage, validity, permissions, owner*

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

- *A user is allowed to perform a cryptographic operation using a cryptographic key if and only if*
  - ○ *the cryptographic key permissions include the user role,*
  - ○ *the cryptographic key usage matches the cryptographic operation, and*
  - ○ *the cryptographic key is valid.*
- *A user is allowed to delete a cryptographic key if the key permissions is allowed to be deleted and the cryptographic key owner matches the user role .*

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

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

FMT_MSA.3/Crypto Static attribute initialization

FMT_MSA.3.1/Crypto The TSF shall enforce the *Cryptographic Service SFP* to provide *fixed and creator-defined* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Crypto The TSF shall allow the *none* to specify alternative initial values to override the default values when an object or information is created.

Application note: The default values are either fixed or defined by the creator for the security attributes:

- The permissions and usage are defined by the creator
- Upon creation, a key is set to valid, and its owner is set to the creator

FMT_MSA.1/Crypto Management of security attributes

FMT_MSA.1.1/Crypto The TSF shall enforce the *Cryptographic Service SFP* to restrict the ability to modify the security attributes *validity* to *the roles authorised according to the permissions*.

FMT_SMF.1/Crypto

FMT_SMF.1.1/Crypto The TSF shall be capable of performing the following management functions: *invalidating (deleting) a cryptographic key*.

FPT_TEE.1/Crypto Testing of external entities

FPT_TEE.1.1/Crypto The TSF shall run a suite of tests *during initial start-up, periodically during normal operation* to check the fulfillment of *correct functioning of cryptographic operation implemented in hardware*

FPT_TEE.1.2/Crypto If the test fails, the TSF shall *generate an alarm, update the health status, and not provide the cryptographic operation*.

6.1.2.2 Key management Service

FDP_IFC.1/KM Subset information flow control

FDP_IFC.1.1/KM The TSF shall enforce the *Key Management SFP* on

Subjects:

- *User*

Information:

- *Cryptographic key*

Operations:

- *Import*
- *Export*

FDP_IFF.1/KM Simple security attributes

FDP_IFF.1.1/KM The TSF shall enforce the *Key Management SFP* based on the following types of subject and information security attributes:

- *User: role*
- *Cryptographic key: permissions, encryption status*

FDP_IFF.1.2/KM The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- *Import: Cryptographic keys can be imported by a user if their role matches the required permissions of the cryptographic key*
- *Export: Cryptographic keys can be exported by a user if their role matches the required permissions of the cryptographic key, if the permissions of the cryptographic key allow export, and if the permissions of the cryptographic key match the encryption status of the key.*

FDP_IFF.1.3/KM The TSF shall enforce the *following additional rules: none.*

FDP_IFF.1.4/KM The TSF shall explicitly authorise an information flow based on the following rules: *none*.

FDP_IFF.1.5/KM The TSF shall explicitly deny an information flow based on the following rules: *none*.

FMT_MSA.3/KM Static attribute initialisation

FMT_MSA.3.1/KM The TSF shall enforce the *Key Management Service SFP* to provide *fixed and creator-defined* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/KM The TSF shall allow the *none* to specify alternative initial values to override the default values when an object or information is created.

Application note: The default values are either fixed or defined by the creator for the security attributes:

- The key permissions are defined by the creator.
- When this creation happens as a result of an import operation, no one can specify the encryption status, the keys always encrypted.

FDP_ITC.1/KM Import of user data without security attributes

FDP_ITC.1.1/KM The TSF shall enforce the *Key Management SFP* when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2/KM The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3/KM The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *none*.

FDP_ETC.1/KM Export of user data without security attributes

FDP_ETC.1.1/KM The TSF shall enforce the *Key Management SFP* when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.1.2/KM The TSF shall export the user data without the user data’s associated security attributes.

6.1.2.3 Dynamic Measurement

FPT_TEE.1/HDIM Testing of external entities

FPT_TEE.1.1/HDIM The TSF shall run a suite of tests *periodically during normal operation, at the request of an authorize user* to check the fulfillment of *dynamic measurement data from the external SoC memory matches the reference data.*

FPT_TEE.1.2/HDIM If the test fails, the TSF shall *generate an alarm*. **If the test succeeds, the TSF shall provide evidence to the requesting user.**

Application note: This SFR has been refined to include a TSF action when the test succeeds. This refinement makes the SFR more strict, as this action is additional to the required action upon failure, and it is clearly related to the component as actions based on the test result relate to the test. This refinement is a better fit for this TOE than the addition of a FAU_SAG.1 claim (which would generate audit data based on the success or failure of the test) as the TSF merely provides evidence of the successful result as a response and not an auditable log including time stamps, etc.

6.1.2.4 Hardware Unique Device Identity

FDP_IFC.1/HUDI Subset information flow control

FDP_IFC.1.1/HUDI The TSF shall enforce the *Device Identity SFP* on

Subjects:

- *User*
- *Hardware*
- *TSF*

Information:

- *Component Identity*

Operations:

- *Identify*

Application note: Component identity can correspond to chip identity, board identity, or device identity.

FDP_IFF.1/HUDI Simple security attributes

FDP_IFF.1.1/HUDI The TSF shall enforce the *Device Identity SFP* based on the following types of subject and information security attributes:

- *All subjects: Identity*
- *Component Identity: Authentication Status*

FDP_IFF.1.2/HUDI The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- *Identify:*
  - *The information Component Identity is allowed to flow from the Hardware to the TSF.*
  - *The information Component Identity is allowed to flow from the TSF to the User, if and only if its Authentication Status is authenticated.*

FDP_IFF.1.3/HUDI The TSF shall enforce the *following additional rules:*

- *If the Hardware response indicates that the Component Identity authentication data generation is successful during an Identify operation, the Authentication Status is set to authenticated.*

FDP_IFF.1.4/HUDI The TSF shall explicitly authorise an information flow based on the following rules: *none*.

FDP_IFF.1.5/HUDI The TSF shall explicitly deny an information flow based on the following rules: *none*.

FDP_ITC.1/HUDI Import of user data without security attributes

FDP_ITC.1.1/HUDI The TSF shall enforce the *Device Identity SFP* when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2/HUDI The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3/HUDI The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *none*.

FDP_ETC.1/HUDI Export of user data without security attributes

FDP_ETC.1.1/HUDI The TSF shall enforce the *Device Identity SFP* when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.1.2/HUDI The TSF shall export the user data without the user data’s associated security attributes.

6.1.2.5 Device Upgrade

FPT_RCV.4 Function recovery

FPT_RCV.4.1 The TSF shall ensure that *device upgrade and failure to authenticate the upgrade image* have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

FPT_TEE.1/Upgrade

FPT_TEE.1.1/Upgrade The TSF shall run a suite of tests *before the upgrade operation* to check the fulfillment of *integrity of images stored in primary and backup areas in the external flash memory.*

FPT_TEE.1.2/Upgrade If the test fails, the TSF shall *not execute the upgrade operation*.

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:

- *Failure of cryptographic self-test*
- *The image certificate key has been revoked*
- *Failure to authenticate the upgrade image*
- *The image version is lower than the OTP counter value*
- *An exception occurs*

FPT_RCV.2 Automated recovery

FPT_RCV.2.1 When automated recovery from *none* is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.2.2 For *any failure when authenticating the loaded software*, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

Application note: The secure state differs depending on the failure. If the first failure is an authentication failure or an image version failure, the TSF shall try to authenticate the recovery copy. If this recovery authentication is successful, then the TSF is in a fully operational secure state. If this recovery authentication also fails, or if the first failure is caused by an exception or because the cryptographic self-test failed, the secure state consists of a reset.

FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1 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: *sensitive data stored in SRAM.*

6.1.2.6 Communication Core Module

FDP_IFC.1/Comm Subset information flow control

FDP_IFC.1.1/Comm The TSF shall enforce the *Communication SFP* on

Subjects:

- *User*

- *Hardware*
- *TSF*

Information:

- *Command ID & Body*
- *Command Payload*
- *Response*

Operations:

- *Receive*
- *Transmit*

FDP_IFF.1/Comm Simple security attributes

FDP_IFF.1.1/Comm The TSF shall enforce the *Communication SFP* based on the following types of subject and information security attributes:

- *For all subjects: Identity*
- *Command ID & Body: Integrity Data*
- *Command Payload: Integrity Data, Encryption Status*
- *Response: Integrity Data, Encryption Status*

FDP_IFF.1.2/Comm The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [

- *Receive:*
  - *The information Command ID & Body and Command Payload is allowed to flow from the User to the Hardware.*
  - *The information Command ID & Body is allowed to flow from the Hardware to the TSF, if and only if the Integrity Data is verified.*
  - *The information Command Payload is allowed to flow from the Hardware to the TSF, if and only if the Integrity Data is verified, and the Encryption Status is decrypted.*
- *Transmit:*
  - *The information Response is allowed to flow from the TSF to the Hardware*
  - *The information Response is allowed to flow from the Hardware to the User, if and only if the Integrity Data is present, and the Encryption Status is encrypted.*

FDP_IFF.1.3/Comm The TSF shall enforce the *following additional rules:*

- *If the Hardware response indicates that the Command ID & Body integrity verification is successful during a Receive operation, the Integrity Data is set to verified.*
- *If the Hardware response indicates that the Command Payload integrity verification is successful during a Receive operation, the Integrity Data is set to verified.*
- *If the Hardware response indicates that the Command Payload decryption is successful during a Receive operation, the Encryption Status is set to decrypted.*
- *If the Hardware response indicates that the Response integrity data generation is successful during a Transmit operation, the Integrity Data is set to present.*
- *If the Hardware response indicates that the Response encryption is successful during a Transmit operation, the Encryption Status is set to decrypted.*

FDP_IFF.1.4/Comm The TSF shall explicitly authorise an information flow based on the following rules: *none*.

FDP_IFF.1.5/Comm The TSF shall explicitly deny an information flow based on the following rules: *none*.

FMT_MSA.3/Comm Static attribute initialisation

FMT_MSA.3.1/Comm The TSF shall enforce the *Communication SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Comm The TSF shall allow the *none* to specify alternative initial values to override the default values when an object or information is created.

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles

- *Manufacturer*
- *User Administrator*
- *User Operator*
- *Public User*

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

FIA_UID.1 Timing of identification

FIA_UID.1.1 The TSF shall allow

- *Querying Health Status*
- *Trusted Boot Metric Extensions*
- *HDIM Measurement Target Configuration only once after boot.*
- *Initiating Role Initialization*

on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application note: The health status comprises whether any hardware exceptions have been raised and whether any external tests of cryptographic functionality have failed.

FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow

- *Querying Health Status*
- *Trusted Boot Metric Extensions*
- *HDIM Measurement Target Configuration*
- *Initiating Role Initialization*

on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note: The TOE enforces the timing of authentication, but the authentication mechanism is implemented with the support of the non-TOE hardware as described in Section 4.3.2.

FDP_ACC.1/Service Subset access control

FDP_ACC.1.1/Service The TSF shall enforce the *Service SFP* on

Subjects:

- *User*

Objects:

- *Service*

Operations:

- *Request*

FDP_ACF.1/Service Security attribute based access control

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

- *User: Role*
- *Service: Access Control List (ACL).*

FDP_ACF.1.2/Service The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *A User is allowed to Request a Service if and only if the Service ACL allows access to the User Role.*

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

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

FMT_MSA.3/Service Static attribute initialisation

FMT_MSA.3.1/Service The TSF shall enforce the *Service SFP* to provide *fixed* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Service The TSF shall allow the *none* to specify alternative initial values to override the default values when an object or information is created.

6.1.2.7 Secure Storage

FDP_ACC.1/Storage Subset access control

FDP_ACC.1.1/Storage The TSF shall enforce the *Secure Storage SFP* on

Subjects:

- *User*

Objects:

- *Non-persistent object*

Operations:

- *Read*
- *Update*
- *Delete*

FDP_ACF.1/Storage Security attribute based access control

FDP_ACF.1.1/Storage The TSF shall enforce the *Secure Storage SFP* to objects based on the following:

- *User: role*
- *Non-persistent object: owner, permissions*

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

- *A user is allowed to perform any operation on an object if and only if the object permissions state that the user role is allowed to perform this operation.*

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

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

FMT_MSA.3/Storage Static attribute initialisation

FMT_MSA.3.1/Storage The TSF shall enforce the *Secure Storage SFP* to provide *creator-defined* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Storage The TSF shall allow the *none* to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1/Storage Management of security attributes

FMT_MSA.1.1/Storage The TSF shall enforce the *Secure Storage SFP* to restrict the ability to *modify* the security attributes *permissions* to *nobody*.

6.1.2.8 Remote Attestation

FDP_ACC.1/Attestation Subset access control

FDP_ACC.1.1/Attestation The TSF shall enforce the *Remote Attestation SFP* on

Subjects:

- *User*

Objects:

- *PCR*
- *DMR(DIM Measurement Value)*
- *DIM report*

Operations:

- *PCR initialize*
- *PCR reset*
- *PCR extend*

- *PCR read*
- *PCR digest calculate*
- *PCR attestation issuance*
- *DMR initialize*
- *DMR extend*
- *DMR attestation issuance*
- *DIM report read*

FDP_ACF.1/ Attestation Security attribute based access control (Attestation)

FDP_ACF.1.1/Attestation The TSF shall enforce the *Remote Attestation SFP* based on the following types of subject and information security attributes:

- *User: role*

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

- *If the role is a user role then all the operations are allowed.*
- *If the role is a user administrator or manufacture or public user then all the operations are not allowed.*

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

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

FMT_MSA.3/ Attestation Management of security attributes (Attestation)

FMT_ MSA.3.1/Attestation The TSF shall enforce the *Remote Attestation SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_ MSA.3.2/Attestation The TSF shall allow the *nobody* to specify alternative initial values to override the default values when an object or information is created.

6.1.2.9 Audit

FAU_SAG.1 Audit data generation

FAU_SAG.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) All auditable events for the *not specified* level of audit; and

b) *The following specifically defined auditable events*

- *Role initialization*
- *Role access authentication success*
- *Role access exit*

- *Role access authentication exception*
- *Adding/deleting/exporting/updating sensitive user data*
- *User data usage authentication exception*
- *Security attribute change*
- *Self-tests failed.*

FAU_SAG.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *no other audit relevant information*.

Application note: The date and time of the event comprise a non-persistent counter, counting from reset.

6.1.3 Security Assurance Requirements

The claimed Security Assurance Requirements are given by the package claim in Section 2.1, and the corresponding SARs are included here by reference to Part 3.

6.2 Security Requirements Rationale

6.2.1 Security Functional Requirements

6.2.1.1 Dependencies

Table 4 shows the dependencies of the Security Functional requirements, whether they are met, and, if not, the justification.

Table 6 Security Functional Requirements dependencies

SFR	Dependencies	Met by	Justification
FCS_COP.1/RSASignPadding	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	-	The cryptographic operation comprises encoding and decoding which does not require a cryptographic key. Therefore these dependencies do not apply.
	FCS_CKM.4	-	
FCS_COP.1/RSAEdPadding	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	-	The cryptographic operation comprises encoding and decoding which does not require a cryptographic key. Therefore these dependencies do not apply.
	FCS_CKM.4	-	

FDP_ACC.1/Crypto	FDP_ACF.1	FDP_ACF.1/Crypto	The dependency is met.
FDP_ACF.1/Crypto	FDP_ACC.1	FDP_ACC.1/Crypto	The dependencies are met.
	FMT_MSA.3	FMT_MSA.3/Crypto	
FMT_MSA.3/Crypto	FMT_MSA.1	FMT_MSA.1/Crypto	The dependencies are met.
	FMT_SMR.1	FMT_SMR.1	
FMT_MSA.1/Crypto	FDP_ACC.1 or FDP_IFC.1	FDP_ACC.1/Crypto	The dependencies are met.
	FMT_SMR.1	FMT_SMR.1	
	FMT_SMF.1	FMT_SMF.1/Crypto	
FMT_SMF.1/Crypto	-	-	There is no dependency.
FPT_TEE.1/Crypto	-	-	There is no dependency.
FDP_IFC.1/KM	FDP_IFF.1	FDP_IFF.1/KM	The dependency is met.
FDP_IFF.1/KM	FDP_IFC.1	FDP_IFC.1/KM	The dependencies are met.
	FMT_MSA.3	FMT_MSA.3/KM	
FMT_MSA.3/KM	FMT_MSA.1	-	The dependency on FMT_MSA.1 does not apply because there is no need for management of security attributes by users. The creator fixes the key permissions at creation, and the encryption status is only managed by the TSF.
	FMT_SMR.1	FMT_SMR.1	
FDP_ITC.1/KM	FDP_ACC.1 or FDP_IFC.1	FDP_IFC.1/KM	The dependencies are met.
	FMT_MSA.3	FMT_MSA.3/KM	
FDP_ETC.1/KM	FDP_ACC.1 or FDP_IFC.1	FDP_IFC.1/KM	The dependency is met.
FPT_TEE.1/HDIM	-	-	There is no dependency.
FDP_IFC.1/HUDI	FDP_IFF.1	FDP_IFF.1/HUDI	The dependency is met.
FDP_IFF.1/HUDI	FDP_IFC.1	FDP_IFC.1/HUDI	The dependency on FMT_MSA.3 does not apply, because the subject identities are pre-defined, and the default value and further management of the authentication status is completely governed by
	FMT_MSA.3	-	
FDP_ITC.1/HUDI	FDP_ACC.1 or FDP_IFC.1	FDP_IFC.1/HUDI	
	FMT_MSA.3	-	

			the additional rules in FDP_IFF.1.3.
FDP_ETC.1/HUDI	FDP_ACC.1 or FDP_IFC.1	FDP_IFC.1/HUDI	The dependency is met.
FPT_RCV.4	-	-	There is no dependency.
FPT_TEE.1/Upgrade	-	-	There is no dependency.
FPT_FLS.1	-	-	There is no dependency.
FPT_RCV.2	AGD_OPE.1	AGD_OPE.1	The dependency is met.
FDP_RIP.1	-	-	There is no dependency.
FDP_IFC.1/Comm	FDP_IFF.1	FDP_IFF.1/Comm	The dependency is met.
FDP_IFF.1/Comm	FDP_IFC.1	FDP_IFC.1/Comm	The dependencies are met.
	FMT_MSA.3	FMT_MSA.3/Comm	
FMT_MSA.3/Comm	FMT_MSA.1	-	The dependency on FMT_MSA.1 does not apply, because all security attribute management is done by the TSF.
	FMT_SMR.1	FMT_SMR.1	
FMT_SMR.1	FIA_UID.1	FIA_UID.1	The dependency is met.
FIA_UID.1	-	-	There is no dependency.
FIA_UAU.1	FIA_UAU.1	FIA_UID.1	The dependency is met.
FDP_ACC.1/Service	FDP_ACF.1	FDP_ACF.1/Service	The dependency is met.
FDP_ACF.1/Service	FDP_ACC.1	FDP_ACC.1/Service	The dependencies are met.
	FMT_MSA.3	FMT_MSA.3/Service	
FMT_MSA.3/Service	FMT_MSA.1	-	The dependency on FMT_MSA.1 does not apply, because the security attributes have immutably fixed values.
	FMT_SMR.1	FMT_SMR.1	
FDP_ACC.1/Storage	FDP_ACF.1	FDP_ACF.1/Storage	The dependency is met.
FDP_ACF.1/Storage	FDP_ACC.1	FDP_ACC.1/Storage	The dependencies are met.
	FMT_MSA.3	FMT_MSA.3/Storage	
FMT_MSA.3/Storage	FMT_MSA.1	FMT_MSA.1/Storage	The dependencies are met.
	FMT_SMR.1	FMT_SMR.1	
FMT_MSA.1/Storage	FDP_ACC.1	FDP_ACC.1/Storage	The dependencies are met.

	or FDP_IFC.1		
	FMT_SMR.1	FMT_SMR.1	
FDP_ACC.1/Attestation	FDP_ACF.1	FDP_ACF.1/ Attestation	The dependencies is met.
FDP_ACF.1/Attestation	FDP_ACC.1	FDP_ACF.1/ Attestation	The dependencies are met.
	FMT_MSA.3	FMT_MSA.3/Attestation	
FMT_MSA.3/Attestation	FMT_MSA.1	FMT_MSA.3/Comm	The dependencies are met.
	FMT_SMR.1	FMT_SMR.1	
FAU_SAG.1	FPT_STM.1	-	The dependency on FPT_STM.1 does not apply because, as described by the application note, the time stamps comprise a non-persistent counter. Reliable time stamps are not required.

6.2.1.2 Tracing

Table 5 shows an overview of the tracing from SFRs to the Security Objectives. The details are provided in the following.

Table 7 Tracing from Security Functional Requirements to Security Objectives

Security Objectives SFRs	O.Cryptographic-Service	O.Key-Management-Service	O.HDIM	O.HUDI	O.Upgrade	O.Communication	O.Secure-Storage	O.HRA	O.Audit
FCS_COP.1/RSASignPadding	√								
FCS_COP.1/RSAEdPadding	√								
FDP_ACC.1/Crypto	√								
FDP_ACF.1/Crypto	√								
FMT_MSA.3/Crypto	√								
FMT_MSA.1/Crypto	√								
FMT_SMF.1/Crypto	√								

Security Objectives / SFRs	O.Cryptographic-Service	O.Key-Management-Service	O.HDIM	O.HUDI	O.Upgrade	O.Communication	O.Secure-Storage	O.HRA	O.Audit
FPT_TEE.1/Crypto	√								
FDP_IFC.1/KM		√							
FDP_IFF.1/KM		√							
FMT_MSA.3/KM		√							
FDP_ITC.1/KM		√							
FDP_ETC.1/KM		√							
FPT_TEE.1/HDIM			√						
FDP_IFC.1/HUDI				√					
FDP_IFF.1/HUDI				√					
FDP_ITC.1/HUDI				√					
FDP_ETC.1/HUDI				√					
FPT_RCV.4					√				
FPT_TEE.1/Upgrade					√				
FPT_FLS.1					√				
FPT_RCV.2					√				
FDP_RIP.1					√				
FDP_IFC.1/Comm						√			
FDP_IFF.1/Comm						√			
FMT_MSA.3/Comm						√			
FMT_SMR.1						√			
FIA_UID.1						√			

Security Objectives / SFRs	O.Cryptographic-Service	O.Key-Management-Service	O.HDIM	O.HUDI	O.Upgrade	O.Communication	O.Secure-Storage	O.HRA	O.Audit
FIA_UAU.1						√			
FDP_ACC.1/Service						√			
FDP_ACF.1/Service						√			
FMT_MSA.3/Service						√			
FDP_ACC.1/Storage							√		
FDP_ACF.1/Storage							√		
FMT_MSA.3/Storage							√		
FMT_MSA.1/Storage							√		
FDP_ACC.1/Attestation								√	
FDP_ACF.1/Attestation								√	
FDP_MSA.3/Attestation								√	
FAU_SAG.1									√

The Security Objective **O.Cryptographic-Service** states that the TOE shall provide access to cryptographic services. Most of these are provided by the hardware environment, but the TOE implements part of these services as required by FCS_COP.1. Furthermore, the TOE restricts the access to cryptographic services according to the access control policy defined by the SFRs FDP_ACC.1/Crypto, FDP_ACF.1/Crypto, FMT_MSA.3/Crypto, FMT_MSA.1/Crypto, and FMT_SMF.1/Crypto. Finally, the TOE allows the user to test the external hardware cryptographic support as required by FPT_TEE.1/Crypto.

The Security Objective **O.Key-Management-Service** regarding the import and export of cryptographic keys is directly met by requiring the TOE to implement an information flow control policy (including import and export) as defined by the SFRs FDP_IFC.1/KM, FDP_IFF.1/KM, FMT_MSA.3/KM, FDP_ITC.1/KM, and FDP_ETC.1/KM.

The Security Objective **O.HDIM** for dynamic measurements on the external SoC memory is directly met by the SFR FPT_TEE.1/HDIM that requires the TOE to test this external entity.

The Security Objective **O.HUDI** requiring the TOE to export device identity information is realized through the SFRs FDP_IFC.1/HUDI, FDP_IFF.1/HUDI, FDP_ITC.1/HUDI, and FDP_ETC.1/HUDI, which specify the information flow control rules that the TOE must implement to provide this feature.

The Security Objective **O.Upgrade** is that the TOE shall authenticate upgrade images with the support of the hardware platform, preserving a secure state in terms of failure. This is directly met by requirement FPT_RCV.4, whereas FPT_FLS.1 describes the failures and secure state. FPT_TEE.1/Upgrade and FPT_RCV.2 require that the TOE verify the availability and integrity of a recovery image, and that automated procedures ensure that a secure state is maintained. Finally, FDP_RIP.1 requires that all sensitive data is cleared from memory after the operation.

The Security Objective **O.Communication** requires a flow control policy on the incoming and outgoing communication data to help prevent unauthorized modification and disclosure. This is directly met by the SFRs FDP_IFC.1/Comm, FDP_IFF.1/Comm, FMT_MSA.3/Comm, FMT_SMR.1, FIA_UID.1, and FIA_UAU.1 that define the corresponding information flow control policy. It additionally requires a that the TOE shall implement role-based service access control, which is directly met by the SFRs FDP_ACC.1/Service, FDP_ACF.1/Service, and FMT_MSA.3/Service, in addition to the aforementioned FMT_SMR.1, FIA_UID.1, and FIA_UAU.1, which together define this access control policy.

The Security Objective **O.Secure-Storage** requires that the non-persistent objects in memory are protected by an access control policy, which is directly met by FDP_ACC.1/Storage, FDP_ACF.1/Storage, FMT_MSA.3/Storage, FMT_MSA.1/Storage, FIA_UID.1, and FIA_UAU.1 that define the corresponding access control policy.

The Security Objective **O.HRA** requires the software register PCR and operations are protected by access control policy,which is directly met by FDP_ACC.1/Attestation, FDP_ACF.1/Attestation,FMT_MSA.3/Attestation that define the corresponding access control policy.

The Security Objective **O.Audit** is directly met by FAU_SAG.1 that defines the audit policy.

6.2.2 Security Assurance Requirements

6.2.2.1 Dependencies

The dependencies of the claimed package are internally consistent. Additionally, Table 6 shows that the dependencies for the augmentations are met as well.

Table 8 Security Assurance Requirements dependencies for augmentations

SAR	Dependencies	Met by	Justification
AVA_VAN.5	ADV_ARC.1	ADV_ARC.1	Part of EAL4
	ADV_FSP.4	ADV_FSP.4	Part of EAL4
	ADV_TDS.3	ADV_TDS.3	Part of EAL4
	ADV_IMP.1	ADV_IMP.1	Part of EAL4

	AGD_OPE.1	AGD_OPE.1	Part of EAL4
	AGD_PRE.1	AGD_PRE.1	Part of EAL4
ALC_DVS.2	-	-	N/A

6.2.2.2 Justification

An assurance level of EAL4 with the augmentations AVA_VAN.5 and ALC_DVS.2 are required for this TOE since it is intended to defend against sophisticated attacks. Maximum assurance can be gained from positive security engineering based on good commercial practices. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defence against such attacks, the evaluators will have access to the low level design and source code.

7 TOE summary specification

This chapter provides information to potential users of the TOE how the TOE satisfies the Security Functional Requirements.

7.1 SF.Crypto

SF.Crypto implements digest encoding according to EMSA-PSS-ENCODE and digest decoding according to EMSA-PSS-VERIFY (both as defined in [PKCS#1]) with key sizes 2048/3072/4096. This functionality is provided as a service to the user, and thereby ensures that the TOE meets FCS_COP.1/RSASignPadding.

It further implements message encoding/decoding according to EME-OAEP encoding and decoding (as defined in [PKCS#1]) with key sizes 2048/3072/4096. This functionality is provided as a service to the user, and thereby ensures that the TOE meets FCS_COP.1/RSAEdPadding.

It also implements an access control policy based on user roles to verify that requesting users are allowed to access cryptographic services, thereby ensuring that the TOE meets FDP_ACC.1/Crypto, FDP_ACF.1/Crypto, FMT_MSA.3/Crypto, FMT_MSA.1/Crypto, and FMT_SMF.1/Crypto.

Finally, it tests the external hardware environment providing the cryptographic operations to ensure their proper working during start-up and periodically during normal operation, thereby ensuring that the TOE meets FPT_TEE.1/Crypto.

7.2 SF.KM

SF.KM implements the key management information control policy. Specifically, it ensures that keys are allowed to be exported before exporting and ensures that they are only exported in encrypted format if required. For both import and export, it ensures that the requesting user is allowed to perform these operations. As a result, it ensures that the TOE meets FDP_IFC.1/KM, FDP_IFF.1/KM, FMT_MSA.3/KM, FDP_ITC.1/KM, and FDP_ETC.1/KM.

7.3 SF.HDIM

SF.HDIM implements a mechanism for dynamic measurement of the external memory contents (three specified memory spaces outside of HSM and four ranges in the external flash memory), which is performed periodically or on request of a user. Both upon success and on failure of this measurement, the user will receive evidence of the result. This ensures that the TOE meets FPT_TEE.1/HDIM.

7.4 SF.HUDI

SF.HUDI implements a process to communicate device identity information stored in the hardware environment to the requesting user, thereby ensuring that the TOE meets FDP_IFC.1/HUDI, FDP_IFF.1/HUDI, FDP_ITC.1/HUDI, and FDP_ETC.1/HUDI.

7.5 SF.Upgrade

SF.Upgrade implements the secure upgrade process of the update the second-stage bootloader image or the TOE image from the external flash memory, before the upgrade operation，run a suite of tests to check the fulfillment of integrity of images, including the handling of hardware initialization failure, recovery image integrity verification failure, software authentication failure, and any other unrecoverable exceptions. In case an authentication failure occurs, SF.Upgrade will attempt an automated recovery using the backup copy of the software image. If any failure occurs, SF.Upgrade ensures that a secure state is maintained by

- Aborting the upgrade process
- Clearing sensitive data such as decryption key and plaintext image for the upgrade

In case such failures do not occur, SF.Upgrade nevertheless clears sensitive data for the upgrade. Therefore, SF.Upgrade ensures that the TOE meets FPT_RCV.4, FPT_RCV.2, FPT_FLS.1, FPT_TEE.1/Upgrade and FDP_RIP.1.

7.6 SF.CCM

SF.CCM is the Communication Core Module that implements the communication with the user. It facilitates that command headers, command payloads, and responses are processed correctly by the hardware, which ensures that the TOE meets FDP_IFC.1/Comm, FDP_IFF.1/Comm, and FMT_MSA.3/Comm. Furthermore, it verifies that users are authenticated according to their role and restricts access to services accordingly, thereby ensuring that the TOE meets FMT_SMR.1, FIA_UID.1, FIA_UAU.1, FDP_ACC.1/Service, FDP_ACF.1/Service, and FMT_MSA.3/Service.

7.7 SF.Storage

SF.Secure-Storage implements an access control mechanism on non-persistent objects that are involved in the process of running the system, thereby ensuring that the TOE meets FDP_ACC.1/Storage, FDP_ACF.1/Storage, FMT_MSA.3/Storage, and FMT_MSA.1/Storage.

7.8 SF.HRA

SF.HRA implements an access control mechanism on software register PCR that store the sensitive measurement results involved in remote attestation with server, thereby ensuring that the TOE meets FDP_ACC.1/Attestation, FDP_ACF.1/ Attestation, FMT_MSA.3/ Attestation.

7.9 SF.Audit

SF.Audit implements an audit function for the auditable events, thereby ensuring that the TOE meets FAU_SAG.1 Audit data generation.

8 Bibliography

8.1 Evaluation Documents

[1] Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017

[2] Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017

[4] Common Methodology for Information Technology Security Evaluation: Evaluation Methodology, Version 3.1, Revision 5, April 2017

8.2 Developer Documents

[5] Huawei HSM 2.0 management firmware Guidance Document, Version 1.7

[6] Huawei HSM 2.0 management firmware service command specification, Version 1.6

8.3 Other Documents

[7] [FIPS180-4], FIPS Publication 180-4, Secure Hash Standard (SHS), Federal Information Processing Standards, August 2015

[8] [FIPS198-1], FIPS Publication 198-1, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards, July 2008

[9] [PKCS#1], PKCS #1 v2.2: RSA Cryptography Standard, RSA Laboratories, October 27, 2012

[10] [SP800-38D], NIST Special Publication 800-38D, Recommendations for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, National Institute of Standards and Technology, November 2007

[11] [SP800-132], NIST Special Publication 800-132, Recommendations for Password-Based Key Derivation: Part 1: Storage Applications, December 2010