Edge Compute Node

Protection Profile

Version Number	1.0.7
Updated On	September 4th, 2020

TABLE OF CONTENTS

LIST OF TABLES

1 Protection Profile Introduction

The main body of this document defines the Base Protection Profile for the security manager for Edge Compute Node. This Base-PP must be used in a PP-configuration with one of the PP-modules defined in the appendix B, C or D which extend the Base-PP with mutually exclusive means of implementing secure storage feature, cryptography and support for the secure boot. Refer to appendix E for the valid PP-configurations.

1.1 Protection Profile Reference

PP Title: **Edge Compute Node Base Protection Profile**

PP Version: version **1.0**.7, September 4th, 2020

CC Identification: CC for Information Technology (IT) Security Evaluation, Version 3.1, Revision 5, April 2017.

1.2 TOE Overview

In the context of Internet of Things (IoT), an Edge Compute Node (ECN) is a piece of hardware and software located between a network of IoT leaf devices (an IoT network) and an IoT Edge Cloud. It has the capability of performing local processing of data from IoT leaf devices through a runtime environment offered to developers and of acting as a bridge between the IoT Edge Cloud and IoT leaf devices. The Edge Compute Node can be provisioned and administrated from the IoT Edge Cloud by a trusted administrator.

For this Base-PP, the TOE is the ECN Security Manager in charge of providing the core security features needed for an Edge Compute Node. The TOE is illustrated by the red box in Figure 1.

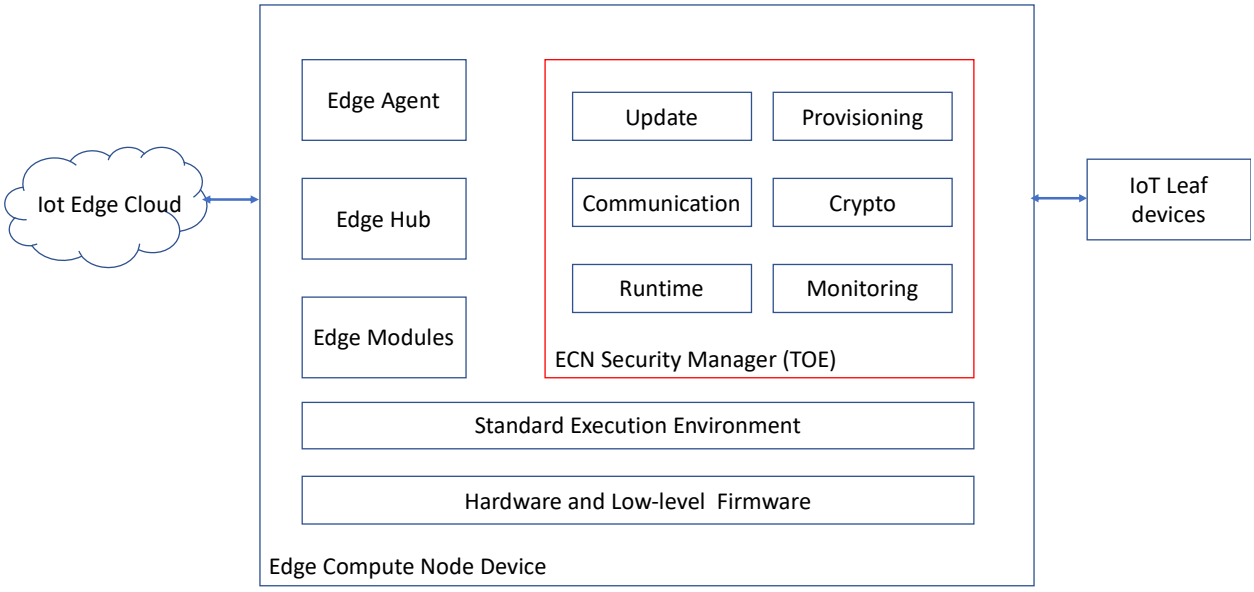


Figure 1 Base-PP Edge Compute Node TOE

This Base-PP TOE must be complemented with features defined in the PP-modules defined in the appendix B, C or D.

1.2.1 Usage and Major Security Features of a TOE

The security features of the ECN Security Manager (TOE) include the following:

- The **Update function**, which provides secure update.
- The **Edge Runtime**, which is the execution runtime for Edge modules.
- The **Provisioning Library**, which provides device identity lifecycle management.
- The **Secure Communication Library**, which provides support of TLS with X.509 certificates.
- The **Cryptographic Library**, which provides cryptographic services for the device, including cryptographic keys.
- The **Monitoring Library**, which generates and monitor security events for the TOE.

1.2.2 TOE Type

The TOE type is a software featuring the security manager for Edge Compute Node.

1.2.3 Available non-TOE hardware/software/firmware

The Available non-TOE hardware/software/firmware for the TOE consists of:

- The supporting **Operating System** (Standard Execution Environment) for the TOE, which provides a runtime environment for the TOE and additional services, such as memory isolation or secure storage for cryptographic keys.

- The **Edge Modules** that implement local edge computing functions for the network of leaf devices.
- The **Edge Hub** in charge of communications with the IoT Edge Cloud.
- The **Edge Agent** in charge of Edge module management.
- The hardware and low-level firmware supporting the TOE, typically based on an Intel or ARM device.
- The networked environment with the IoT Edge Cloud and the leaf devices.

1.3 TOE Security Services

This section summarizes the security services provided by the TOE:

- **Security Audit:** The TOE has the ability to collect audit data, review audit logs, protect audit logs from overflow, and restrict access to audit logs. Audit information generated by the system includes the date and time of the event, the user identity that caused the event to be generated, and other event specific data. Authorized administrators can review audit logs and have the ability to search and sort audit records. Authorized Administrators can also configure the audit system to include or exclude potentially auditable events to be audited based on a wide range of characteristics. In the context of this evaluation, the protection profile requirements cover generating audit events, selecting which events should be audited, and providing integrity protection for stored audit event entries.
- **Cryptographic Support:** The TOE provides cryptographic functions that support encryption/decryption, cryptographic signatures, cryptographic hashing, cryptographic key agreement, and random number generation. The TOE additionally provides support for public keys, credential management and certificate validation functions. In addition to using cryptography for its own security functions, the TOE offers access to the cryptographic support functions for Edge modules.
- **Identification and Authentication**: The TOE provides the ability to use, store, and protect certificates that are used for authentication of the IoT Edge Cloud and to authenticate the TOE (static and dynamic attestation).
- **Protection of the TOE Security Functions**: The TOE provides a number of features to ensure the protection of TOE security functions. It protects against unauthorized data disclosure. The TOE ensures process isolation security for all Edge modules, with support from the Standard Execution Environment. The TOE includes self-testing features that ensure the integrity of executable program images and its cryptographic functions. Finally, The TOE provides a trusted update mechanism to update the TOE binaries itself.
- **Trusted Path for Communications**: The TOE provides protected communications with the IoT Edge Cloud.

- **Security Management:** The TOE provides several functions to manage security policies. This includes management of Edge Modules, cryptographic keys and certificates and auditable events.

1.4 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the protection profile.

1.4.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements (SFRs): Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations.
    - Assignment: allows the specification of an identified parameter.
    - Selection: allows the specification of one or more elements from a list.
    - Refinement: allows the addition of details.

  The conventions for the assignment, selection, refinement, and iteration operations are described in Section 6.

- Other sections of the protection profile use a bold font to highlight text of special interest, such as captions.

1.4.2 Terminology

The following terminology is used in the protection profile:

Term	Definition
Access	Interaction between an entity and an object that results in the flow or modification of data.
Access control	Security service that controls the use of resources[1] and the disclosure and modification of data[2].
Assurance	A measure of confidence that the security features of an IT system are sufficient to enforce the IT system's security policy.

[1] Hardware and software

[2] Stored or communicated

Attack	An intentional act attempting to violate the security policy of an IT system.
Authentication	A security measure that verifies a claimed identity.
Authorization	Permission, granted by an entity authorized to do so, to perform functions and access data.
Availability	Timely[3], reliable access to IT resources.
Compromise	Violation of a security policy.
Common Application Developer	Application developers (or software companies) often produce many applications under the same name. ECN allow shared resources by such applications where otherwise resources would not be shared
Confidentiality	A security policy pertaining to disclosure of data.
Critical cryptographic security parameters	Security-related information appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.
Cryptographic key (key)	A parameter used in conjunction with a cryptographic algorithm that determines: • the transformation of plaintext data into ciphertext data • the transformation of ciphertext data into plaintext data • a digital signature computed from data • the verification of a digital signature computed from data • a data authentication code computed from data
Cryptographic module	The set of hardware, software, and/or firmware that implements approved security functions, including cryptographic algorithms and key generation, which is contained within the cryptographic boundary.
Cryptographic module security policy	A precise specification of the security rules under which a cryptographic module must operate.
Developer Modes	Developer modes are states in which additional services are available to a user in order to provide enhanced system access for debugging of software.
Enclave	A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or based on physical location and proximity.
General-Purpose Operating System	A general-purpose operating system is designed to meet a variety of goals, including protection between users and applications, fast response time

[3] According to a defined metric

	for interactive applications, high throughput for server applications, and high overall resource utilization.
Hardware-protected	Asset (such as a cryptographic key or certificates or cryptographic elements such as a hash) for which storage and processing is done in hardware and result of its usage is provided to software layer. The software layer has a restricted access to the raw data.
Operating environment	The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.
Persistent storage	All types of data storage media that maintain data across system boots (e.g., hard disk, removable media).
Protected data	Protected data is all non-TSF data (user data). Protected data includes all keys in secure key storage.
Public object	An object for which the TSF unconditionally permits all entities “read” access under the Discretionary Access Control SFP. Only the TSF or authorized administrators may create, delete, or modify the public objects.
Security-enforcing	A term used to indicate that the entity (e.g., module, interface, subsystem) is related to the enforcement of the TOE security policies.
Security-supporting	A term used to indicate that the entity (e.g., module, interface, subsystem) is not security-enforcing; however, the entity’s implementation must still preserve the security of the TSF.
System services	All services provided by the TOE to Edges modules through an application interface. Examples of system services include access to network interface, storage, cryptography. The TSS shall list all system services available for use by Edges modules.
Threat	Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.
Trust Anchor Database	A list of trusted root Certificate Authority certificates.
Trusted endpoints	Servers (IoT Edge Cloud) or IoT leaf devices the TOE is designed to communicate with.
Unauthorized individual	A type of threat agent in which individuals who have not been granted access to the TOE attempt to gain access to information or functions provided by the TOE.
Unauthorized user	A type of threat agent in which individuals who are registered and have been explicitly granted access to the TOE may attempt to access information or functions that they are not permitted to access.

Vulnerability	A weakness that can be exploited to violate the TOE security policy.

Table 1 Definitions

1.4.3 Acronyms

The acronyms used in this protection profile are specified **in List of Abbreviations**.

1.4.4 References

NIST SP 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, December 2001

NIST SP 800-38C, Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, May 2004 (Updated 7/20/2007)

NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007

NIST SP 800-35E, Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices, January 2010

NIST SP 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, December 2012

NIST SP 800-56A, Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, Revision 3, April 2018

NIST SP 800-56B, Recommendation for Pair-Wise Key-Establishment Using Integer Factorization Cryptography, Revision 2, March 2019

NIST SP 800-57, Recommendation for Key Management: Part 1 – General, Revision 5, May 2020

NIST SP 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Revision 1, June 2015

FIPS PUB 186-4, Digital Signature Standard (DSS), July 2013

FIPS PUB 197, Advanced Encryption Standard (AES), November 2001

FIPS PUB 198-1, The Keyed-Hash Message Authentication Code (HMAC), July 2008

IEEE 802.11, IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012

IEEE 802.11ac-2013, IEEE Standard for Information technology--Telecommunications and information exchange between systems—Local and metropolitan area networks--Specific requirements--Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications--Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz.

1.5 PP Organization

This Protection Profile contains the following additional sections:

- Conformance Claims (Section 2): Provides the conformance claims for the PP.
- Security Problem Definition (Section 3): Describes the threats, organizational security policies and assumptions that pertain to the TOE.
- Security Objectives (Section 4): Identifies the security objectives that are satisfied by the TOE and the TOE operational environment.
- Extended Components Definition (Section 5): Defines the extended components used in the security requirements.
- Security Requirements (Section 6): Presents the security functional and assurance requirements met by the TOE.
- Rationale for Security Requirements (Section 7): Presents the rationale for the security objectives, requirements, and TOE Summary Specification as to their consistency, completeness and suitability.
- List of Abbreviations (Appendix A)
- Secure Boot and File System Secure Storage PP-Module (Appendix B)
- Support for HSM-Based Secure Storage and Cryptography PP-Module (Appendix C)
- Support for Secure Enclave Secure Storage and Cryptography PP-Module (Appendix D)
- Supported PP-Configurations (Appendix E)
- Initialization Vector Requirements for NIST- Approved Cipher Modes (Appendix F)

2 Conformance Claims

2.1 CC Conformance Claims

This PP is conformant with the following specification:

- [CC1] *Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model*, Version 3.1, Revision 5, April 2017
- [CC2] *Common Criteria for Information Technology Security Evaluation Part 2: Security functional components*, Version 3.1, Revision 5, April 2017, extended (Part 2 extended)
- [CC3] *Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components*, Version 3.1, Revision 5 April 2017, extended with ALC_TSU_EXT.1 (Part 3 extended)

The PP is inspired from the following specification, although no conformance is claimed:

- [MDF PP] *Mobile Device Fundamentals Protection Profile*, Version 3.1, June 6th, 2017.

2.2 Conformance Claims of the PP

This PP does not claim conformance to any other PP.

2.3 Conformance Claims to a Package

The minimum assurance claims for a PP-Configuration with this Base-PP is EAL1 augmented by ASE_SPD.1, ASE_OBJ.2 and ASE_REQ.2 and augmented CC Part 3 ALC_TSU_EXT.1 assurance requirement.

This conformance claim also applies to the PP-configurations defined in this document.

2.4 Conformance Rationale

This PP does not provide a conformance rationale because it does not claim conformance to any other PP.

2.5 Conformance Statement

The Protection Profile requires strict conformance of the ST or PP claiming conformance to this PP.

3 Security Problem Definition

The security problem definition consists of the assets, threats to security, organizational security policies, and assumptions as they relate to the TOE.

3.1 Assets

Table 2 presents assets that need to be protected by the TOE.

Asset	Description
Device ID	The unique device identification set during manufacturing. *Properties: integrity*
Parameters	The parameters stored and managed by the TOE: • TSF version. • Device Root CA certificate. • IoT Edge Cloud connection information, set after device provisioning sequence. • Code signing certificates, used to verify integrity and authenticity of code. • Edge modules configuration and life-cycle state. *Properties: integrity*
Code	The code for Edge Modules managed by the TOE. *Properties: integrity*
Logs	The auditable events generated by the TOE. *Properties: integrity*
Edge data	Edge module data exchanged between the TOE and trusted endpoints (IoT Edge Cloud or leaf devices). *Properties: integrity, confidentiality*

Table 2 Assets

3.2 Threats

Table 3 presents known or presumed threats to protected resources that are addressed by Edge Compute Nodes.

Threat	Description
T.EAVESDROP	If positioned on a wireless communications channel or elsewhere on the network, a remote attacker may monitor and gain access

	to Edge data exchanged between the TOE and other trusted endpoints. *Threatened assets: Edge data (confidentiality)*
T.NETWORK	A remote attacker may initiate communications with the TOE or alter communications between the TOE and trusted endpoints to compromise the Edge data. *Threatened assets: Edge data (integrity)*
T.FLAWMOD	A local or remote attacker could abuse TOE interfaces, for instance through malicious or exploitable Edge Module code, in order to gain unauthorized access to the TOE, or additional privileges and the ability to conduct further malicious activities. TSF or user data may be compromised or altered. *Threatened assets: Device ID, Parameters, Code, Logs, Edge data (confidentiality and integrity).*
T.PERSISTENT	After successfully performing one or several adverse actions of the threats of this PP, the local or remote attacker also gain persistent undetected presence on TOE. TOE has lost integrity and cannot regain it. The TOE and its data may be controlled by an attacker going undetected by the TOE users. *Threatened assets: Device ID, Parameters, Code, Logs, Edge data (confidentiality and integrity).*

Table 3 Threats

3.3 Organizational Security Policies

There are no organizational security policies for this protection profile.

3.4 Assumptions

Table 4 describes the core security aspects of the environment in which the TOE is intended to be used. It includes information about the physical, personnel, procedural, and connectivity aspects of the environment.

The following specific conditions are assumed to exist in an environment where the TOE is employed in order to conform to the protection profile:

Assumption	Description
A.ADMIN	It is assumed that the TOE administrators correctly configure the TOE's security functions and the underlying platform (i.e. OS, hardware and low-level firmware) if applicable. TOE administrators keep the OS and related library up to date and

	apply security patches when available. OS updates are verified using digital signature.
A.KEYS	It is assumed that the TOE administrators ensure the confidentiality (for symmetric or private keys) and integrity of cryptographic keys and certificates used outside of the TOE to encrypt communications or to authenticate the TOE.
A.PLATFORM	It is assumed that the underlying platform (i.e. OS, libraries, hardware and low-level firmware) provides adequate security, including domain separation and non-bypassability. In particular, the platform ensures applicative memory separation (no other applicative process can access TOE memory).
A.SECURE_BOOT	It is assumed that the underlying platform (i.e. OS, libraries, hardware and low-level firmware) provides a secure boot feature which authenticates executable code loaded in memory, from the low-level firmware up to the TOE, prior its execution.
A.NO_GENERAL_PURPOSE	It is assumed that there will be no general-purpose computing capabilities (e.g., compilers or user applications) available on the underlying platform (i.e. OS), other than those services necessary for the operation, administration, and support of the TOE.
A.PHYSICAL	It is assumed that the TOE environment provides appropriate physical security, commensurate with the value of the assets protected by the TOE.
A.STORAGE	It is assumed that the underlying platform (i.e. OS) provides data-at-rest protection feature for cryptographic keys and certificates used by the TOE.

Table 4 Secure Usage Assumptions

4 Security Objectives

This section defines the security objectives of Edge Compute Nodes and their operational environment. Security objectives, categorized as either TOE security objectives or objectives by the operational environment, reflect the stated intent to counter identified threats, comply with any organizational security policies identified, or address identified assumptions. All of the identified threats, organizational policies, and assumptions are addressed under one of the categories below.

4.1 TOE Security Objectives

Table 5 describes the security objectives for Edge Compute Nodes.

Security Objective	Source
O.COMMS	The TOE will provide the capability to communicate using trusted channels, such as TLS, as a means to maintain the confidentiality and integrity of data that are transmitted between the TOE and trusted endpoints.
O.AUTH	The TOE will provide the capability to authenticate trusted endpoints and only allow authorized network connections with them.
O.CONFIG	The TOE will provide the capability to configure and apply security policies defined by TOE administrators.
O.INTEGRITY	The TOE will maintain the integrity of its critical functionality, software and data, through the capability to perform self-tests and security auditing, to verify the integrity of downloaded updates and to restrict applications to only have access to the system services and data they are permitted to interact with.

Table 5 Security Objectives for the TOE

4.2 Security Objectives for the Operational Environment

Table 6 describes the security objectives for the operational environment.

Environment Objective	Description
OE.ADMIN	TOE administrators configure TOE's security functions and the underlying platform if applicable following the security guidance. TOE administrators keep the OS and related library up to date and apply security patches when available. OS updates are verified using digital signature.
OE.KEYS	TOE administrators ensure the confidentiality (for symmetric or private keys) and integrity of cryptographic keys and certificates

	used outside of the TOE to encrypt communications or to authenticate the TOE.
OE.PLATFORM	The underlying platform (i.e. OS, hardware and low-level firmware) provides adequate security, including domain separation (such as a kernel and user mode and isolation between processes) and non-bypassability. In particular, the platform ensures applicative memory separation (no other applicative process can access TOE memory). **Application note:** Domain separation and non-bypassability at the OS level should also include anti-exploitation techniques, such as address space layout randomization (ASLR), memory page permissions, stack-based buffer overflow protection.
OE.SECURE_BOOT	The underlying platform (i.e. OS, libraries, hardware and low-level firmware) provides a secure boot feature which authenticates executable code loaded in memory, from the low-level firmware up to the TOE, prior its execution.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the underlying platform (i.e. OS), other than those services necessary for the operation, administration, and support of the TOE.
OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attacks that might compromise the TOE assets, with protections commensurate to the value of those assets.
OE.STORAGE	The underlying platform (i.e. OS) provides data-at-rest protection feature for cryptographic keys and certificates used by the TOE.

Table 6 Security Objectives for the Operational Environment

4.3 Security Objectives Rationale

This Section gives an evidence for sufficiency and necessity of the defined objectives. It shows that all threats and OSPs are addressed by the security objectives and it also shows that all assumptions are addressed by the security objectives for the TOE operational environment. The following table provides an overview for security objectives coverage (TOE and its environment).

	O.COMMS	O.AUTH	O.CONFIG	O.INTEGRITY	OE.ADMIN	OE.KEYS	OE.PLATFORM	OE.SECURE_BOOT	OE.NO_GENERAL_PURPOSE	OE.PHYSICAL	OE.STORAGE
T.EAVESDROP	X		X		X	X					X
T.NETWORK	X	X	X		X	X					X
T.FLAWMOD			X	X	X		X		X	X	
T.PERSISTENT			X	X	X		X	X	X	X	X
A.ADMIN					X						
A.KEYS						X					
A.PLATFORM							X				
A.SECURE_BOOT								X			
A.NO_GENERAL_PURPOSE									X		
A.PHYSICAL										X	
A.STORAGE											X

4.3.1 Security Objective Rationales: Threats

T.EAVESDROP: The combination of the following security objectives diminishes the eavesdropping of communication channels threat:

- O.COMMS ensures confidentiality of exchanged data through a secure communication channel such as TLS.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG.
- OE.KEYS and OE.STORAGE protect the keys and certificates used to communicate with the TOE outside of the TOE (i.e. trusted endpoints and underlying platform, respectively).

T.NETWORK: The combination of the following security objectives diminishes the alteration of communication threat:

- O.COMMS ensures integrity of exchanged data through a secure communication channel such as TLS.
- O.AUTH ensures authentication of communication with trusted end-points.

- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG.
- OE.KEYS and OE.STORAGE protect the keys and certificates used to communicate with the TOE outside of the TOE (i.e. trusted endpoints and underlying platform, respectively).

T.FLAWMOD: The combination of the following security objectives diminishes the TOE compromising threat:

- O.INTEGRITY ensures integrity of critical functionality, software and updates and controls access to system services.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG and ensures that the TOE is correctly configured and the underlying platform up-to-date.
- OE.PLATFORM provides OS support for domain separation and non-bypassability at the OS level mainly to protect TOE processes from other processes running in the OS with memory isolation and also includes anti-exploitation techniques to mitigate potential breaches.
- OE.NO_GENERAL_PURPOSE ensures that OS has not computing capabilities that could be used by an attacker reducing the exploitability of attacks.
- OE.PHYSICAL provides physical protection for the TOE against attackers with physical access to the TOE.

T.PERSISTENT: The combination of the following security objectives diminishes the persistent access to the TOE threat:

- O.INTEGRITY ensures integrity of critical functionality, software/firmware and data and updates and controls access to system services.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG and ensures that the TOE is correctly configured and the underlying platform up-to-date.
- OE.PLATFORM provides OS support for domain separation and non-bypassability at the OS level mainly to protect TOE processes from other processes running in the OS with memory isolation and also includes anti-exploitation techniques to mitigate potential breaches.
- OE.SECURE_BOOT provides support for authentication of the underlying platform code and the TOE.
- OE.NO_GENERAL_PURPOSE ensures that OS has not computing capabilities that could be used by an attacker reducing the exploitability of attacks.
- OE.PHYSICAL provides physical protection for the TOE against attackers with physical access to the TOE.
- OE.STORAGE cryptographic keys and certificates used by the TOE

4.3.2 Security Objective Rationales: Assumptions

A.ADMIN: The security objective for the environment OE.ADMIN directly upholds this assumption.

A.KEYS: The security objective for the environment OE.KEYS directly upholds this assumption.

A.PLATFORM: The security objective for the environment OE.PLATFORM directly upholds this assumption.

A.SECURE_BOOT: The security objective for the environment OE.SECURE_BOOT directly upholds this assumption.

A.NO_GENERAL_PURPOSE: The security objective for the environment OE.NO_GENERAL_PURPOSE directly upholds this assumption.

A.PHYSICAL: The security objective for the environment OE.PHYSICAL directly upholds this assumption.

A.STORAGE: The security objective for the environment OE.STORAGE directly upholds this assumption.

5 Extended Components Definition

5.1 Security Functional Requirements

This protection profile makes use of extended components, not defined in [CC2]. These components are identified in Table 7, with the related requirement class from [CC2].

New families are introduced for behaviors not specified in [CC2]. Extended components either specify more specific abilities compared to existing [CC2] component (the similar Part 2 requirement is given in Table 7) or specify new abilities for the TOE (new component family, no similar Part 2 requirement).

Requirement Class	Extended Requirement Component	Similar Part 2 Requirement
Cryptographic Support (FCS)	Extended: Random Bit Generation (FCS_RBG_EXT.1)	none
	Extended: Cryptographic Algorithm Services (FCS_SRV_EXT.1)	none
	Extended: TLS Protocol (FCS_TLS_EXT.1)	none
	Extended: Validation of Certificates (FCS_X509_EXT.1)	none
	Extended: X509 Certificate Authentication (FCS_X509_EXT.2)	none
	Extended: Request Validation of Certificates (FCS_X509_EXT.3)	none
Security Management (FMT)	Extended: Management of Security Functions Behavior (FMT_MOF_EXT.1)	FMT_MOF.1
Protection of the TSF (FPT)	Extended: Domain Isolation (FPT_AEX_EXT.1)	none
	Extended: Self-Test Failure (FPT_FLS_EXT.1)	none
	Extended: Specification of Remediation Actions (FPT_SRA_EXT.1)	none
	Extended: TSF Cryptographic Functionality Testing (FPT_TST_EXT.1)	none
	Extended: Trusted Update: TSF Version Query (FPT_TUD_EXT.1)	none
	Extended: Trusted Update Verification (FPT_TUD_EXT.2)	none

Table 7: Base-PP Extended Security Functional Requirements

Auditable events for extended components in this PP are given in **Table 9**.

Management activities for extended components in this PP are given in Table 17.

5.1.1 Cryptographic Support (FCS)

5.1.1.1 Definition of the Family FCS_RBG_EXT

The family FCS_RBG_EXT describes the functional requirements for random number generation used for cryptographic purposes.

Components Levelling:

This family has a single level.

FCS_RBG_EXT Random Bit Generation	1

5.1.1.1.1 Extended: Random Bit Generation (FCS_RBG_EXT.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RBG_EXT.1.1	The TSF shall perform all deterministic random bit generation services in accordance with [selection: *Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)*].
FCS_RBG_EXT.1.2	The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [assignment: *noise source*] with a minimum of [selection: *128 bits, 256 bits*] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.
FCS_RBG_EXT.1.3	The TSF shall be capable of providing output of the RBG to applications running on the TSF that request random bits.

Application Notes:

- Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES) are specified in NIST SP 800-90 A.
- Noise source can be software-based, or hardware-based if the TOE type from this Base-PP is extended in a PP-Module to also include hardware.

5.1.1.2 Definition of the Family FCS_SRV_EXT

The family FCS_SRV_EXT describes the functional requirements for the TSF to provide cryptographic algorithm services to outside of the TOE.

Components Levelling:

This family has a single level.

FCS_SRV_EXT Cryptographic Algorithm Services	1

5.1.1.2.1 Extended: Cryptographic Algorithm Services (FCS_SRV_EXT.1)

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation.

FCS_SRV_EXT.1.1 The TSF shall provide a mechanism for applications to request the TSF to perform the following cryptographic operations: [assignment: *list of cryptographic operations*].

FCS_SRV_EXT.1.2 The TSF shall provide a mechanism for applications to request the TSF to perform the following cryptographic operations: [assignment: *list of cryptographic operations*] by keys stored in the secure key storage.

5.1.1.3 Definition of the Family FCS_TLS_EXT

The family FCS_TLS_EXT describes the functional requirements for TLS protocol as a trusted channel.

Components Levelling:

This family has a single level.

5.1.1.3.1 Extended: TLS Protocol (FCS_TLS_EXT.1)

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation
FCS_X509_EXT.1 Extended: Validation of Certificates
FCS_X509_EXT.2 Extended: X509 Certificate Authentication.

FCS_ TLS_EXT.1.1 The TSF shall support the following TLS 1.2 (RFC 5246) ciphersuites:

- Mandatory Ciphersuites: [selection:
  - *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
  - *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*]
- Optional Ciphersuites: [selection:
  - *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,*
  - *TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,*
  - *TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,*
  - *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,*
  - *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,*
  - *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,*

- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,*
- *no other ciphersuite]*.

FCS_ TLS_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_ TLS_EXT.1.3 The TSF shall not establish a trusted channel if the peer certificate is invalid.

FCS_ TLS_EXT.1.4 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note: The SFR does not require full compliance with the TLS 1.2 (RFC 5246) standard, and compliance is restricted only to the specific aspects mentioned in the SFR.

5.1.1.4 Definition of the Family FCS_X509_EXT

The family FCS_X509_EXT describes the functional requirements for validation of X.509 certificates.

Components Levelling:

This family has three levels.

5.1.1.4.1 Extended: Validation of Certificates (FCS_X509_EXT.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a certificate in the Trust Anchor Database.

- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

FCS_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Application Note: The SFR does not require full compliance with the mentioned RFC standards, and compliance is restricted only to the specific aspects mentioned in the SFR.

5.1.1.4.2 Extended: X509 Certificate Authentication (FCS_X509_EXT.2)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for TLS, and [selection: *code signing for system software updates, code signing for applications, code signing for integrity verification, [assignment: other uses], no additional uses*].

FCS_X509_EXT.2.2 When the TSF cannot establish a connection to determine the revocation status of a certificate, the TSF shall [selection: *allow the administrator to choose whether to accept the certificate in these cases, allow the user to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

Application Note: The SFR does not require full compliance with the mentioned RFC standard, and compliance is restricted only to the specific aspects mentioned in the SFR.

5.1.1.4.3 Extended: Request Validation of Certificates (FCS_X509_EXT.3)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_X509_EXT.3.1 The TSF shall provide a certificate validation service to applications.

FCS_X509_EXT.3.2 The TSF shall respond to the requesting application with the success or failure of the validation.

5.1.2 Security Management (FMT)

5.1.2.1 Definition of the Family FMT_MOF_EXT

The family FMT_MOF_EXT describes the functional requirements for management of security functions behavior.

Components Levelling:

This family has a single level.

5.1.2.1.1 Extended: Management of Security Functions Behavior (FMT_MOF_EXT.1)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions.

FMT_MOF_EXT.1.1 The TSF shall restrict the ability to perform [assignment: *list of security functions*] to the user.

FMT_MOF_EXT.1.2 The TSF shall restrict the ability to perform [assignment: *list of security functions*] to the administrator when the device is enrolled and according to the administrator-configured policy.

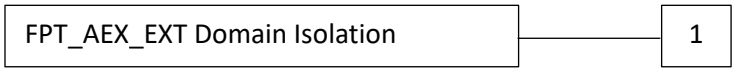
5.1.3 Protection of the TSF (FPT)

5.1.3.1 Definition of the Family FPT_AEX_EXT

The family FPT_AEX_EXT describes the functional requirements for anti-exploitation capabilities.

Components Levelling:

This family has a single level.



5.1.3.1.1 Extended: Domain Isolation (FPT_AEX_EXT.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.1.1 The TSF shall protect itself from modification by untrusted subjects.

FPT_AEX_EXT.1.2 The TSF shall enforce isolation of domains between applications.

Application Notes:

- For FPT_AEX_EXT.1.1, the ST author shall describe in the TSS the mechanisms in place to prevent Edge modules from modifying the TSF software or TSF data that governs the behavior of the TSF (such as boundary checking of inputs to APIs).

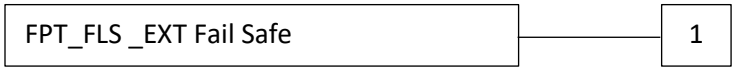
- For FPT_AEX_EXT.1.2, while memory separation is usually under control of the OS (cf. OE.PLATFORM), the TOE is responsible for separation of other domains, such as filesystem, network, IPC, process identifier. The ST author shall describe in the TSS the mechanisms in place to provide separation.
- The evaluator can test these mechanisms by creating and loading an application and try to modify the TSF software, TSF data or other application data through the APIs in place to access domains.

5.1.3.2 Definition of the Family FPT_FLS_EXT

The family FPT_FLS_EXT describes the functional requirements for fail safe.

Components Levelling:

This family has a single level.



5.1.3.2.1 Extended: Self-Test Failure (FPT_FLS_EXT.1)

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit Data Generation.

FPT_FLS_EXT.1.1 The TSF shall transition to non-operational mode, log failures in the audit record and [selection: *notify the administrator, [assignment: other actions], no other actions*] when the following types of failures occur:

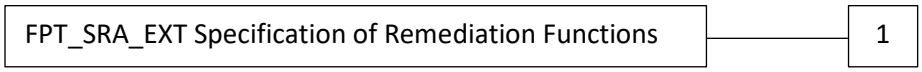
- failures of the self-test(s)
- TSF software integrity verification failures
- [selection: *no other failures, [assignment: other failures]*].

5.1.3.3 Definition of the Family FPT_SRA_EXT

The family FPT_SRA_EXT describes the functional requirements for specification of remediation functions.

Components Levelling:

This family has a single level.



5.1.3.3.1 Extended: Specification of Remediation Actions (FPT_SRA_EXT.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_SRA_EXT.1.1 The TSF shall offer [selection: *wipe of protected data, alert the administrator, remove application, [assignment: list other available remediation actions]*]

upon unenrollment and [selection: *[assignment: other administrator-configured triggers], no other triggers*].

5.1.3.4 Definition of the Family FPT_TST_EXT

The family FPT_TST_EXT describes the functional requirements for TSF self-tests.

Components Levelling:

This family has a single level.

5.1.3.4.1 Extended: TSF Cryptographic Functionality Testing (FPT_TST_EXT.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

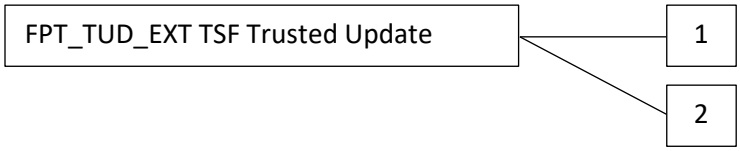
FPT_TST_EXT.1.1 The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of all cryptographic functionality.

5.1.3.5 Definition of the Family FPT_TUD_EXT

The family FPT_TUD_EXT describes the functional requirements for TSF trusted updates.

Components Levelling:

This family has two levels.



5.1.3.5.1 Extended: Trusted Update: TSF Version Query (FPT_TUD_EXT.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TUD_EXT.1.1 The TSF shall provide authorized users the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide authorized users the ability to query the current version of the hardware model of the device.

FPT_TUD_EXT.1.3 The TSF shall provide authorized users the ability to query the current version of installed applications.

5.1.3.5.2 Extended: Trusted Update Verification (FPT_TUD_EXT.2)

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation
FCS_X509_EXT.1 Extended: Validation of Certificates
FCS_X509_EXT.2 Extended: X509 Certificate Authentication.

FPT_TUD_EXT.2.1	The TSF shall verify that the digital signature verification key used for TSF updates is validated to a public key in the Trust Anchor Database.
FPT_TUD_EXT.2.2	The TSF shall verify application software using a digital signature mechanism prior to installation.
FPT_TUD_EXT.2.3	The TSF shall by default only install applications cryptographically verified by [selection: *a built-in X.509v3 certificate, a configured X.509v3 certificate*].
FPT_TUD_EXT.2.4	The TSF shall not install code if the code signing certificate is deemed invalid.
FPT_TUD_EXT.2.5	The TSF shall verify that software updates to the TSF are a current or later version than the current version of the TSF.

5.2 Security Assurance Requirements

5.2.1 Definition of the Family ALC_TSU_EXT

The objective of the family "Timely Security Updates (ALC_TSU_EXT)" is to ensure the developer has a well-defined process in place to deliver updates to mitigate known security flaws.

Components Levelling:

This family has a single level.

ALC_TSU Timely Security Updates	1

5.2.1.1 Timely Security Updates (ALC_TSU_EXT.1)

Dependencies: No dependencies.

Developer action elements:

ALC_TSU_EXT.1.1D The developer shall provide a description in the TSS of how timely security updates are made to the TOE.

Content and presentation elements:

ALC_TSU_EXT.1.1C The description shall include the process for creating and deploying security updates for the TOE software.

Application Note: The process description includes the TOE developer processes as well as any third-party (carrier) processes. The process description includes each deployment mechanism (e.g., over- the-air updates, per-carrier updates, downloaded updates).

ALC_TSU_EXT.1.2C The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

Application Note: The total length of time may be presented as a summation of the periods of time that each party (e.g., TOE developer, mobile carrier) on the critical path consumes. The time period until public availability per deployment mechanism may differ; each is described.

ALC_TSU_EXT.1.3C The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

Application Note: The reporting mechanism could include web sites, email addresses, as well as a means to protect the sensitive nature of the report (e.g., public keys that could be used to encrypt the details of a proof-of-concept exploit).

ALC_TSU_EXT.1.4C The description shall include where users can seek information about the availability of new updates including details (e.g. CVE identifiers) of the specific public vulnerabilities corrected by each update.

Application Note: The purpose of providing this information is so that users can determine which devices are susceptible to publicly known vulnerabilities so that they can make appropriate risk decisions, such as limiting access to resources until updates are installed.

Evaluator action elements:

ALC_TSU_EXT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Assurance Activity:

The evaluator shall verify that the TSS contains a description of the timely security update process used by the developer to create and deploy security updates. The evaluator shall verify that this description addresses the TOE OS, the firmware, and bundled applications, each. The evaluator shall also verify that, in addition to the TOE developer's process, any carrier or other third-party processes are also addressed in the description. The evaluator shall also verify that each mechanism for deployment of security updates is described.

The evaluator shall verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the TOE patching this vulnerability, to include any third- party or carrier delays in deployment. The evaluator shall verify that this time is expressed in a number or range of days.

The evaluator shall verify that this description includes the publically available mechanisms (including either an email address or website) for reporting security issues related to the TOE. The evaluator shall verify that the description of this mechanism includes a method for protecting the report either using a public key for encrypting email or a trusted channel for a website.

The evaluator shall verify that the description includes where users can seek information about the availability of new security updates including details of the specific public vulnerabilities corrected by each update. The evaluator shall verify that the description includes the minimum amount of time that the TOE is expected to be supported with security updates, and the process by which users can seek information about when the TOE is no longer expected to receive security updates.

6 Security Requirements

The section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) for the TOE.

Where applicable the following conventions are used to identify operations:

- **Iteration**: Iterated requirements (components and elements) are identified with letter following the base component identifier. For example, iterations of FCS_COP.1 are identified in a manner similar to FCS_COP.1(SIGN) (for the component) and FCS_COP.1.1(SIGN) (for the elements).
- **Assignment**: Assignments are identified in brackets and bold (e.g., **[assigned value]**).
    - Assignments to be filled in by the ST author appear in brackets and italics with an indication that an assignment has to be made (e.g., [assignment: *value to be assigned*]).
    - An assigned value in the protection profile can also include another assignment or selection (see below) to be filled in by the ST author (e.g., **[an assigned value, [assignment: *remaining value to be assigned*]]** or **[an assigned value, [selection: *remaining value to be selected*]]**).
    - Assignments can be transformed into a selection (see below), in which case the selection also appears in brackets and bold (e.g., **[selection: *value to be selected*]**).
- **Selection**: Selections are identified in brackets, bold, and italics (e.g., ***[selected value]***).
    - Selections to be filled in by the ST author appear in brackets and italics with an indication that an assignment has to be made (e.g., [selection: *value to be selected*]).
    - Assignments within selections are identified using the previous conventions, except that the assigned value would also be italicized and extra brackets would occur (e.g., ***[selected value [assigned value]]***).
- **Refinement**: Refinements are identified using underlined text (e.g., <u>added text</u>) for additions and strike-through text (e.g., ~~deleted text~~) for deletions.

6.1 TOE Security Functional Requirements

This section specifies the SFRs for the TOE.

Requirement Class	Requirement Component
Security Audit (FAU)	Audit Data Generation (FAU_GEN.1)
	Audit Review (FAU_SAR.1)
	Selective Audit (FAU_SEL.1)
	Audit Storage Protection (FAU_STG.1)
	Prevention of Audit Data Loss (FAU_STG.4)
	Cryptographic Key Generation (FCS_CKM.1)

Cryptographic Support (FCS)	Cryptographic Operation for Key Establishment (FCS_COP.1(KE))
	Cryptographic Operation for Data Encryption/Decryption (FCS_COP.1(SYM))
	Cryptographic Operation for Hashing (FCS_COP.1(HASH))
	Cryptographic Operation for Signature Algorithms (FCS_COP.1(SIGN))
	Cryptographic Operation for Keyed Hash Algorithms (FCS_COP.1(HMAC))
	Cryptographic Key Destruction (FCS_CKM.4)
	Extended: Random Bit Generation (FCS_RBG_EXT.1)
	Extended: Cryptographic Algorithm Services (FCS_SRV_EXT.1)
	Extended: TLS Protocol (FCS_TLS_EXT.1)
User Data Protection (FDP)	Subset Access Control (FDP_ACC.1)
	Security Attribute Access Control (FDP_ACF.1)
Identification & Authentication (FIA)	User identification before any action (FIA_UID.2)
	Extended: Validation of Certificates (FCS_X509_EXT.1)
	Extended: X509 Certificate Authentication (FCS_X509_EXT.2)
	Extended: Request Validation of Certificates (FCS_X509_EXT.3)
Security Management (FMT)	Extended: Management of Security Functions Behavior (FMT_MOF_EXT.1)
	Management of TSF data (FMT_MTD.1)
	Management of security attributes (FMT_MSA.1)
	Static attribute initialisation (FMT_MSA.3)
	Specification of Management Functions (FMT_SMF.1)
	Security Roles (FMT_SMR.1)
Protection of the TSF (FPT)	Extended: Domain Isolation (FPT_AEX_EXT.1)
	Extended: Self-Test Failure (FPT_FLS_EXT.1)
	Reliable Time Stamps (FPT_STM.1)
	Extended: Specification of Remediation Actions (FPT_SRA_EXT.1)
	Extended: TSF Cryptographic Functionality Testing (FPT_TST_EXT.1)
	Extended: Trusted Update: TSF Version Query (FPT_TUD_EXT.1)
	Extended: Trusted Update Verification (FPT_TUD_EXT.2)
Trusted Path/Channels (FTP)	Inter-TSF Trusted Channel (FTP_ITC.1)

Table 8 TOE Security Functional Requirements

6.1.1 Security Audit (FAU)

6.1.1.1 Audit Data Generation (FAU_GEN.1)

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and
[
c) Administrator management functions, as defined in the fourth column of Table 17;
d) Start-up and shutdown of the OS;
e) Specifically defined auditable events in Table 9;
f) [assignment: *other specifically defined auditable events*]].

FAU_GEN.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, [**additional information in Table 9**].

Requirement	Auditable Events	Additional Record Contents
FAU_GEN.1	None.	
FAU_SAR.1	None.	
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	No additional Information.
FAU_STG.1	None.	
FAU_STG.4	None.	
FCS_CKM.1	Failure of key generation activity.	No additional Information.
FCS_COP.1	None.	

FCS_RBG_EXT.1	Failure of the randomization process.	No additional information.
FCS_TLS_EXT.1	Failure to establish a TLS session.	Reason for failure.
	Failure to verify presented identifier.	Presented identifier and reference identifier.
	Establishment/termination of a TLS session.	Non-TOE endpoint of connection.
	Application initiation of trusted channel.	Name of application. Trusted channel protocol. Non-TOE endpoint of connection.
FDP_ACC.1	None.	
FDP_ACF.1	None.	
FIA_UAU_EXT.2	Action performed before authentication.	No additional information.
FCS_X509_EXT.1	Failure to validate X.509v3 certificate.	Reason for failure of validation.
FCS_X509_EXT.2	Failure to establish connection to determine revocation status.	No additional information.
FCS_X509_EXT.3	None.	
FMT_MOF_EXT.1.1	None.	
FMT_MOF_EXT.1.2	None.	
FMT_MTD.1	None.	
FMT_MSA.1	None.	
FMT_MSA.3	None.	
FMT_SMF.1	Change of settings.	Role of user that changed setting. Value of new setting.
	Success or failure of function.	Role of user that performed function. Function performed. Reason for failure
	Initiation of software update.	Version of update.

	Initiation of Edge module installation or update.	Name and version of Edge module.
	Addition or removal of certificate from Trust Anchor Database.	Subject name of certificate.
FPT_AEX_EXT.1	Blocked attempt to modify TSF data.	Identity of subject. Identity of TSF data.
FPT_FLS_EXT.1	Measurement of TSF software.	Integrity verification value.
FPT_STM.1	None.	
FPT_SRA_EXT.1	Unenrollment.	Identity of administrator. Remediation action performed.
FPT_TST_EXT.1	Initiation of self-test. Failure of self-test.	None
FPT_TUD_EXT.1	None.	
FPT_TUD_EXT.2	Success or failure of signature verification for software updates.	
	Success or failure of signature verification for Edge modules.	
FTP_ITC.1	Initiation and termination of trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.

Table 9 Auditable Events

6.1.1.2 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [assignment: *authorised users*] with the capability to read [**all audited events and record contents**] 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.

6.1.1.3 Selective Audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) [selection: *object identity, user identity, subject identity, host identity, event type*]

b) [assignment: *list of additional attributes that audit selectivity is based upon*].

6.1.1.4 Audit Storage Protection (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to [***prevent***] unauthorized modifications to the stored audit records in the audit trail.

6.1.1.5 Prevention of Audit Data Loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall [***overwrite the oldest stored audit records***] and [**no other action**] if the audit trail is full.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**selection: *row 1, row 2 or row 3 in column cryptographic algorithms of Table 10***] and specified cryptographic key sizes [**cryptographic key sizes of selected cryptographic key generation algorithms in Table 10**] that meet the following: [**list of standards of selected cryptographic key generation algorithms in Table 10**].

Cryptographic algorithm	Cryptographic key sizes	List of standards
RSA schemes	***[selection: 2048-bit, 3072-bit]***	***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3***
Elliptic curve-based key establishment schemes	***[selection: 256-bit, 384-bit, 521-bit]***	***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 or Curve25519 schemes that meet the following: RFC 7748***
Finite field-based key establishment schemes	***[selection: 2048-bit, 3072-bit]***	***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1***

Table 10: FCS_CKM.1 Cryptographic algorithms

6.1.2.2 Cryptographic Operation for Key Establishment (FCS_COP.1(KE))

FCS_COP.1.1(KE) The TSF shall perform [**cryptographic key establishment**] in accordance with a specified cryptographic algorithm [**algorithm of Table 11 and [selection: *algorithms of Table 12]***] and cryptographic key sizes [**selection: *cryptographic key sizes in FCS_CKM.1.1***] that meet the following: [**list of standards of Table 11 and list of standards of Table 12 for selected algorithms**].

Cryptographic algorithm	List of standards
RSA-based key establishment schemes	**NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"**

Table 11: FCS_COP.1(KE) Mandatory cryptographic algorithm

Cryptographic algorithm	List of standards
Elliptic curve-based key establishment schemes	***NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"***
Finite field-based key establishment schemes	***NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"***
No other schemes	

Table 12: FCS_COP.1(KE) Optional cryptographic algorithms

6.1.2.3 Cryptographic Operation for Data Encryption/Decryption (FCS_COP.1(SYM))

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_COP.1.1(SYM) The TSF shall perform [**encryption/decryption**] in accordance with a specified cryptographic algorithm [**algorithms of Table 13 and [selection: *algorithms of Table 14]]*** and cryptographic key sizes [**128-bit key sizes and [selection: *256-bit key sizes, no other key sizes***]] that meet the following: [**list of standards of Table 13 and list of standards of Table 14 for selected algorithms**].

Cryptographic algorithm	List of standards
AES-CBC FIPS	**FIPS PUB 197**
AES-CBC	**NIST SP 800-38A**
AES-CCMP	**NIST SP 800-38C and IEEE 802.11-2012**

Table 13: FCS_COP.1(SYM) Mandatory cryptographic algorithms

Cryptographic algorithm	List of standards
AES Key Wrap (KW)	***NIST SP 800-38F***
AES Key Wrap with Padding (KWP)	***NIST SP 800-38F***
AES-GCM	***NIST SP 800-38D***
AES-CCM	***NIST SP 800-38C***
AES-XTS	***NIST SP 800-38E***

AES-GCMP-256	***NIST SP 800-38D and IEEE 802.11ac-2013***
No other schemes	

Table 14: FCS_COP.1(SYM) Optional cryptographic algorithms

6.1.2.4 Cryptographic Operation for Hashing (FCS_COP.1(HASH))

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_COP.1.1(HASH) The TSF shall perform [**cryptographic hashing**] in accordance with a specified cryptographic algorithm [**selection: *SHA-256, SHA-384, SHA-512***] and cryptographic key sizes [**none**] that meet the following: [**FIPS Pub 180-4**].

6.1.2.5 Cryptographic Operation for Signature Algorithms (FCS_COP.1(SIGN))

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_COP.1.1(SIGN) The TSF shall perform [**cryptographic signature services (generation and verification)**] in accordance with a specified cryptographic algorithm [**algorithm of Table 15 and [selection: *algorithms of Table 16***]] and cryptographic key sizes [**128-bit key sizes and [selection: *256-bit key sizes, no other key sizes***]] that meet the following: [**list of standards of Table 15 and list of standards of Table 16 for selected algorithms**].

Cryptographic algorithm	Cryptographic key sizes	List of standards
RSA schemes	**[selection: *2048-bit, 3072-bit*]**	**FIPS PUB 186-4, "The RSA Digital Signature Algorithm", Section 5**

Table 15: FCS_COP.1(SIGN) Mandatory cryptographic algorithm

Cryptographic algorithm	Cryptographic key sizes	List of standards
ECDSA schemes	***[selection: 256-bit, 384-bit, 521-bit]***	***FIPS PUB 186-4, "Elliptic Curve Digital Signature Algorithm (ECDSA)", Section 6***
No other schemes		

Table 16: FCS_COP.1(SIGN) Optional cryptographic algorithms

6.1.2.6 Cryptographic Operation for Keyed Hash Algorithms (FCS_COP.1(HMAC))

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_COP.1.1(HMAC) The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**selection: *HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***] and cryptographic key sizes

[**assignment: *key size (in bits) used in HMAC***] that meet the following: [**FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-4, "Secure Hash Standard"**].

6.1.2.7 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [*assignment: cryptographic key destruction method]* that meets the following: [*assignment: list of standards*].

6.1.2.8 Extended: Random Bit Generation (FCS_RBG_EXT.1)

As in Section 5.1.

Application Note: The term "Applications" in FCS_RBG_EXT.1.3 has to be interpreted as "Edge modules".

6.1.2.9 Extended: Cryptographic Algorithm Services (FCS_SRV_EXT.1)

FCS_SRV_EXT.1.1 The TSF shall provide a mechanism for applications to request the TSF to perform the following cryptographic operations: [

- **The following algorithms in FCS_COP.1(SYM): AES-CBC, [selection: *AES Key Wrap, AES Key Wrap with Padding, AES-GCM, AES-CCM, no other modes*]**
- **All mandatory and selected algorithms in FCS_COP.1(SIGN)**
- **All mandatory and selected algorithms in FCS_COP.1(HASH)**
- **All mandatory and selected algorithms in FCS_COP.1(HMAC)**

].

FCS_SRV_EXT.1.2 The TSF shall provide a mechanism for applications to request the TSF to perform the following cryptographic operations: [

- **Algorithms in FCS_COP.1(SYM)**
- **Algorithms in FCS_COP.1(SIGN)**

]
by keys stored in the secure key storage.

Application Notes:

- The term "Applications" FCS_SRV_EXT.1.1 and FCS_SRV_EXT.1.2 has to be interpreted as "Edge modules".
- Secure key storage in FCS_SRV_EXT.1.2 is covered by OE.STORAGE. PP-modules specify means to implement this feature.

6.1.2.10 Extended: TLS Protocol (FCS_TLS_EXT.1)

As in Section 5.1.

6.1.3 User Data Protection (FDP)

6.1.3.1 Subset Access Control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the [**system service access control SFP**] on: [

- **Subjects: Edge module or group of Edge modules**
- **Object: Any information accessible through system services, Edge module data**
- **Operations: system services**].

6.1.3.2 Security Attribute Access Control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [**system service access control SFP**] to objects based on the following: [

- **Subjects: Edge module or group of Edge modules using system services**
- **Object: Any information accessible through system services, Edge module data**
- **Attributes: Privilege, System service access rights ('No application', 'Privileged' or' All applications'), [assignment: *SFP-relevant security attributes*]**].

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

- **No Edge module or group of Edge modules can access system services with System service access rights attribute set to 'No application';**
- **Only Edge module with Privilege attribute or group of Edge modules with Privilege attribute can access system services with System access rights attribute set to 'Privileged';**
- **All Edge modules or groups of Edge modules can access system services with System service access rights attribute set to 'All applications';**
- **Edge module or group of Edge modules [selection: *can only access public, cannot access*] data stored by other Edge modules or groups of Edge modules;**
- **[assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]**].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [

- **Access to Edge module or group of Edge modules data is explicitly authorized by [selection: *the user, the administrator, Common Application Developer*];**
- **[assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects***]].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

6.1.4 Identification and Authentication (FIA)

6.1.4.1 User identification before any action (FIA_UID.2)

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.

6.1.4.2 Extended: Validation of Certificates (FCS_X509_EXT.1)

As in Section 5.1.

6.1.4.3 Extended: X509 Certificate Authentication (FCS_X509_EXT.2)

As in Section 5.1.

Application Note: The term “Applications” in FCS_X509_EXT.2.1 has to be interpreted as “Edge modules”.

6.1.4.4 Extended: Request Validation of Certificates (FCS_X509_EXT.3)

As in Section 5.1.

Application Note: The term “Application” in FCS_X509_EXT.3.1 and FCS_X509_EXT.3.2 has to be interpreted as “Edge module”.

6.1.5 Security Management (FMT)

6.1.5.1 Extended: Management of Security Functions Behavior (FMT_MOF_EXT.1)

FMT_MOF_EXT.1.1 The TSF shall restrict the ability to perform [**the management functions in column 3 of Table 17**] to the user.

Application Note: The management functions that have an “M” in the third column are mandatory for this component, thus are restricted to the user, meaning that the administrator cannot manage those functions. The management functions that have an “O” in the third column are optional and may be selected; and those management functions with a “-” in the third are not applicable and may not be

selected. The ST author should select those management functions that only the user may perform (i.e., the ones the administrator may not perform).

The ST author may not select the same management function in both FMT_MOF_EXT.1.1 and FMT_MOF_EXT.1.2. A management function cannot contain an "M" in both column 3 and column 5.

FMT_MOF_EXT.1.2	The TSF shall restrict the ability to perform [**the management functions in column 5 of Table 17**] to the administrator when the device is enrolled and according to the administrator-configured policy.

Application Note: The management functions that have an "M" in the fifth column are mandatory for this component; the management functions that have an "O" in the fifth column are optional and may be selected; and those management functions with a "-" in the fifth are not applicable and may not be selected.

The ST author may not select the same management function in both FMT_MOF_EXT.1.1 and FMT_MOF_EXT.1.2.

The ST author should select those management functions that the administrator may restrict.

6.1.5.2 Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1	The TSF shall restrict the ability to [***query, [assignment: other operations]***] the [**set of audited events**] to [**administrator**].

6.1.5.3 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1	The TSF shall enforce the [**system service access control SFP**] to restrict the ability to [selection: *change_default, query, modify, delete, [assignment: other operations*]] the security attributes [assignment: *list of security attributes*] to [**the administrator**].

6.1.5.4 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1	The TSF shall enforce the [**system service access control SFP**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the [**the administrator**] to specify alternative initial values to override the default values when an object or information is created.

6.1.5.5 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: [**as in Table 17**].

M: Mandatory

O: Optional / Objective

Management Function	FMT_SMF.1	FMT_MOF EXT.1.1	Admin	FMT_MOF EXT.1.2
1. TSF wipe of protected data	M	-	M	-
2. configure Edge modules installation policy by a. restricting the sources of Edge modules, b. specifying a set of allowed Edge modules based on a digital signature or Edge modules name and version (an Edge modules whitelist), c. denying installation of Edge module***s***	M	-	M	M
3. import keys/secrets into the secure key storage	M	O	O	-
4. destroy imported keys/secrets and any other keys/secrets in the secure key storage	M	O	O	-
5. import X.509v3 certificates into the Trust Anchor Database	M	-	M	O
6. remove imported X.509v3 certificates and all X.509v3 certificates in the Trust Anchor Database	M	O	O	-
7. enroll the TOE in management	M	M	-	-
8. remove Edge modules	M	-	M	O
9. update TOE	M	-	M	0
10. install Edge modules	M	-	M	O
11. enable/disable developer modes	O	O	O	O
12. enable data-at rest protection	O	O	O	O
13. wipe Edge module data	O	O	O	-
14. approve import, removal by Edge modules of X.509v3 certificates in the Trust Anchor Database	O	O	O	O
15. configure whether to establish a trusted channel or disallow establishment if the TSF cannot establish a connection to determine the validity of a certificate	M	O	O	O
16. read audit logs kept by the TSF	O	O	O	-
17. configure certificate used to validate digital signature on Edge modules	O	O	O	O
18. configure the auditable events	O	-	O	O

Management Function	FMT_SMF .1	FMT_MOF EXT.1.1	Admin	FMT_MOF EXT.1.2
19. retrieve TSF-software integrity verification values	O	O	O	O

Table 17 Management Functions

Application Note: The secure storage feature mentioned in some of the management functions is covered by OE.STORAGE and also addressed in PP-Modules in the Appendices, depending of implementation choices for this feature.

The first column lists the management functions identified in the PP. In the following columns:

- 'M' means Mandatory
- 'O' means Optional

The second column (FMT_SMF.1) indicates whether the function is to be implemented. The ST author should select which Optional functions are implemented.

The third column (FMT_MOF_EXT.1.1) indicates functions that are to be restricted to the user (i.e., not available to the administrator).

The fourth column (Administrator) indicates functions that are available to the administrator. The functions restricted to the user (column 3) cannot also be available to the administrator. Functions available to the administrator can still be available to the user, as long as the function is not restricted to the administrator (column 5). Thus, if the TOE must offer these functions to the administrator to perform the fourth column shall be selected.

The fifth column (FMT_MOF_EXT.1.2) indicates whether the function is to be restricted to the administrator when the device is enrolled and the administrator applies the indicated policy. If the function is restricted to the administrator the function is not available to the user. This does not prevent the user from modifying a setting to make the function stricter, but the user cannot undo the configuration enforced by the administrator.

6.1.5.6 Security Roles (FMT_SMR.1)

FMT_SMR.1.1	The TSF shall maintain the roles [**user, administrator, a common application developer**].
FMT_SMR.2.1	The TSF shall be able to associate users with roles.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 Extended: Domain Isolation (FPT_AEX_EXT.1)

As in Section 5.1.

Application Notes:

- For FPT_AEX_EXT.1.2, the term "Application" in FPT_AEX_EXT.1.2 has to be interpreted as "Edge module".

6.1.6.2 Extended: Self-Test Failure (FPT_FLS_EXT.1)

As in Section 5.1.

6.1.6.3 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1	The TSF shall be able to provide reliable time stamps for its own use.

6.1.6.4 Extended: Specification of Remediation Actions (FPT_SRA_EXT.1)

As in Section 5.1.

Application Note: The term “Application” in FPT_SRA_EXT.1.1 has to be interpreted as “Edge module”.

6.1.6.5 Extended: TSF Cryptographic Functionality Testing (FPT_TST_EXT.1)

As in Section 5.1.

6.1.6.6 Extended: Trusted Update: TSF Version Query (FPT_TUD_EXT.1)

As in Section 5.1.

Application Note: The term “Applications” in FPT_TUD_EXT.1.3 has to be interpreted as “Edge modules”.

6.1.6.7 Extended: Trusted Update Verification (FPT_TUD_EXT.2)

As in Section 5.1.

Application Note: The term “Applications” in FPT_TUD_EXT.2.2 and FPT_TUD_EXT.2.3 has to be interpreted as “Edge modules”.

6.1.7 Trusted Path / Channels (FTP)

6.1.7.1 Inter-TSF Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for [**Administrator management functions as in the fourth column of Table 17**].

6.2 TOE Security Assurance Requirements

6.2.1 CC Part 3 Assurance Requirements

This section lists the set of SARs from CC part 3 that are required in evaluations against this PP. It consists of EAL1 augmented by ASE_SPD.1, ASE_OBJ.2 and ASE_REQ.2 and augmented CC Part 3

ALC_TSU_EXT.1 assurance requirement. Components highlighted in bold represent augmentations on the EAL1 assurance package.

Requirement Class	Assurance Component
ASE: Security Target	ASE_INT.1: ST introduction
	ASE_CCL.1: Conformance claims
	ASE_OBJ.2: Security objectives
	ASE_ECD.1: Extended components definition
	ASE_REQ.2: Derived security requirements
	ASE_SPD.1: Security problem definition
	ASE_TSS.1: TOE summary specification
ADV: Design	ADV_FSP.1: Basic functional specification
AGD: Guidance Documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle Support	ALC_CMC.1: Labeling of the TOE
	ALC_CMS.1: TOE CM Coverage
	ALC_TSU_EXT.1: Timely Security Updates
ATE: Testing	ATE_IND.1: Independent testing - sample
AVA: Vulnerability Assessment	AVA_VAN.1: Vulnerability survey

7 Rationale for Security Requirements

This section provides a rationale for the security functional requirements and security assurance requirements.

7.1 Security Functional Requirements Rationale

The mapping presented in Table 18 traces each SFR back to the security objectives of the TOE and demonstrates how the security objectives are met by the SFRs.

Table 18 Mapping of SFRs to TOE Security Objectives

SFR	O.COMMS	O.AUTH	O. CONFIG	O.INTEGRITY
FAU_GEN.1				X
FAU_SAR.1				X
FAU_SEL.1				X
FAU_STG.1				X
FAU_STG.4				X
FCS_CKM.1	X			
FCS_COP.1(KE)	X			
FCS_CKM.4	X			
FCS_COP.1(SYM)	X			
FCS_COP.1(HASH)	X			X
FCS_COP.1(SIGN)	X	X		X
FCS_COP.1(HMAC)	X	X		
FCS_RBG_EXT.1	X	X		
FCS_SRV_EXT.1	X			
FCS_TLS_EXT.1	X	X		
FDP_ACC.1				X
FDP_ACF.1				X
FMT_MSA.1				X
FMT_MSA.3				X
FCS_X509_EXT.1	X			
FCS_X509_EXT.2		X		

SFR	O.COMMS	O.AUTH	O. CONFIG	O.INTEGRITY
FCS_X509_EXT.3	X			
FIA_UID.2			X	
FMT_MOF_EXT.1			X	
FMT_MTD.1			X	
FMT_SMF.1			X	
FMT_SMR.1			X	
FPT_AEX_EXT.1				X
FPT_FLS_EXT.1				X
FPT_STM.1				X
FPT_SRA_EXT.1			X	X
FPT_TST_EXT.1				X
FPT_TUD_EXT.1				X
FPT_TUD_EXT.2				X
FTP_ITC.1	X			

O.COMMS is addressed by FTP_ITC.1 that provide a trusted channel protected in integrity and confidentiality between the TOE and another trusted IT products (trusted endpoints) used by trusted administrator to provision and administrate the TOE. This trusted channel relies on TLS and X.509 certificates, as addressed by the requirements FCS_TLS_EXT.1, FCS_X509_EXT.1 and FCS_X509_EXT.3. The cryptography required for the trusted channel is supported by FCS_CKM.1, FCS_COP.1(KE), FCS_CKM.4, FCS_COP.1(SYM), FCS_COP.1(HASH), FCS_COP.1(SIGN), FCS_COP.1(HMAC) and FCS_RBG_EXT.1 for random number generation and accessible to Edge Module through the requirement FCS_SRV_EXT.1.

O.AUTH is addressed specifically by the use of authentication algorithms supported by cryptographic requirements FCS_X509_EXT.2 for X.509 certificate authentication. The authentication occurs during TLS connection establishment addressed in requirements FCS_TLS_EXT.1 supported by the cryptography addressed in requirements FCS_COP.1(SIGN) and FCS_COP.1(HMAC).

O. CONFIG is addressed by the set of management requirements for the TOE FMT_MOF_EXT.1, FMT_MTD.1, FMT_SMF.1, FPT_SRA_EXT.1 and supporting identity requirements FIA_UID.2 and FMT_SMR.1.

O.INTEGRITY is addressed by several techniques. First, self-tests for the TOE, as addressed in FPT_TST_EXT.1, self-protection of the TOE as in FPT_AEX_EXT.1 and access control to TOE system services as specified in the SFP of FDP_ACC.1, FDP_ACF.1, FMT_MSA.1 and FMT_MSA.3. Then the detection of a breach of integrity through the audit requirements FAU_GEN.1, FAU_SAR.1, FAU_SEL.1, FAU_STG.1, FAU_STG.4 and FPT_STM.1 and the reaction to failures as defined in FPT_FLS_EXT.1 and

FPT_SRA_EXT.1. And finally by the verification of integrity of TOE update, as addressed in requirements FPT_TUD_EXT.1, FPT_TUD_EXT.2 supporting by cryptography requirements FCS_COP.1(HASH) and FCS_COP.1(SIGN).

7.2 Security Requirements Dependency Rationale

SFR	Dependencies	Resolved
FAU_GEN.1	FPT_STM.1	Yes: FPT_STM.1
FAU_SAR.1	FAU_GEN.1	Yes: FAU_GEN.1
FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	Yes: FAU_GEN.1 Yes: FMT_MTD.1
FAU_STG.1	FAU_GEN.1	Yes: FAU_GEN.1
FAU_STG.4	FAU_STG.1	Yes: FAU_STG.1
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Yes: FCS_COP.1(KE), FCS_COP.1(SIGN), Yes: FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Yes: FCS_CKM.1
FCS_COP.1(KE)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes: FCS_CKM.1 Yes: FCS_CKM.4
FCS_COP.1(SYM)		
FCS_COP.1(HASH)		
FCS_COP.1(SIGN)		
FCS_COP.1(HMAC)		
FCS_RBG_EXT.1	No dependencies	
FCS_SRV_EXT.1	FCS_COP.1	Yes: FCS_COP.1(SYM), FCS_COP.1(HASH), FCS_COP.1(SIGN), FCS_COP.1(HMAC)
FCS_TLS_EXT.1	FCS_COP.1 FCS_X509_EXT.1 FCS_X509_EXT.2	Yes: FCS_COP.1(KE) Yes: FCS_X509_EXT.1 Yes: FCS_X509_EXT.2
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Yes: FDP_ACC.1 Yes: FMT_MSA.3

SFR	Dependencies	Resolved
FCS_X509_EXT.1	No dependencies	
FCS_X509_EXT.2	No dependencies	
FCS_X509_EXT.3	No dependencies	
FIA_UID.2	No dependencies	
FMT_MOF_EXT.1	FMT_SMR.1 FMT_SMF.1	Yes: FMT_SMR.1 Yes: FMT_SMF.1
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	Yes: FMT_SMR.1 Yes: FMT_SMF.1
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	Yes: FDP_ACC.1 Yes: FMT_SMR.1 Yes: FMT_SMF.1
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Yes: FMT_MSA.1 Yes: FMT_SMR.1
FMT_SMF.1	No dependencies	
FMT_SMR.1	FIA_UID.1	Yes: FIA_UID.2 (hierarchical to FIA_UID.1)
FPT_AEX_EXT.1	No dependencies	
FPT_FLS_EXT.1	FAU_GEN.1	Yes: FAU_GEN.1
FPT_STM.1	No dependencies	
FPT_SRA_EXT.1	No dependencies	
FPT_TST_EXT.1	No dependencies	
FPT_TUD_EXT.1	No dependencies	
FPT_TUD_EXT.2	FCS_COP.1 FCS_X509_EXT.1 FCS_X509_EXT.2	Yes: FCS_COP.1(SIGN) Yes: FCS_X509_EXT.1 Yes: FCS_X509_EXT.2
FTP_ITC.1	No dependencies	

Table 19: Security Requirements Dependency Rationale

7.3 Security Assurance Requirements Rationale

The statement of security assurance requirements (SARs) found in section **6.2 TOE Security Assurance Requirements**, is aimed to protect against software attacks with low attack potential. As the TOE is connected to a network and can be targeted by an attacker in case of a new reported vulnerability, the augmented component ALC_TSU_EXT.1 provides timely security updates.

Appendix A List of Abbreviations

Abbreviation	Meaning
3DES	Triple DES
ACL	Access Control List
ACP	Access Control Policy
AES	Advanced Encryption Standard
AGD	Administrator Guidance Document
API	Application Programming Interface
CA	Certificate Authority
CBC	Cipher Block Chaining
CC	Common Criteria
CM	Configuration Management; Control Management
CP	Content Provider
CPU	Central Processing Unit
CRL	Certificate Revocation List
CSP	Cryptographic Service Provider
DES	Data Encryption Standard
DH	Diffie-Hellman
DMA	Direct Memory Access
DNS	Domain Name System
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ECB	Electronic Code Book
ECN	Edge Compute Node
FIPS	Federal Information Processing Standard
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP
HW	Hardware
I/O	Input / Output

I&A	Identification and Authentication
IA	Information Assurance
ID	Identification
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPv4	IP Version 4
IPv6	IP Version 6
IPC	Inter-process Communication
IPI	Inter-process Interrupt
IT	Information Technology
IV	Initialisation Vector
KDF	Key Derivation Function
LAN	Local Area Network
NIST	National Institute of Standards and Technology
OS	Operating System
PKI	Public Key Infrastructure
PP	Protection Profile
RAM	Random Access Memory
RNG	Random Number Generator
RSA	Rivest, Shamir and Adleman
SA	Security Association
SAR	Security Assurance Requirement
SE	Secure Element
SHA	Secure Hash Algorithm
SF	Security Functions
SFP	Security Functional Policy
SFR	Security Functional Requirement
ST	Security Target
TCP	Transmission Control Protocol

TLS	Transport Layer Security
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSC	TOE Scope of Control
TSF	TOE Security Functions
TSS	TOE Summary Specification

Appendix B Secure Boot and File System Secure Storage PP-Module

B.1 PP-Module Introduction

This PP-module must be flattened with the base-PP for the PP-configuration called **Edge Compute Node with Secure Boot and File System Secure Storage**, identified in Section E.1 using the content of this Appendix**.**

B.1.1 Protection Profile Module Reference

PP-Module Title: **Secure Boot and File System Secure Storage PP-Module**

Related Base-PP Title: Edge Compute Node Protection Profile

PP-Module Version: version **1.0**.7, September 4th, 2020

CC Identification: CC for Information Technology (IT) Security Evaluation, Version 3.1, Revision 5, April 2017.

B.1.2 TOE Overview

This PP-Module extends the Base-PP with a secure boot feature and a secure storage for protected data (data-at-rest protection) on a persistent memory of the Edge Compute Node. The related TOE is composed of the ECN Security Manager, as in the base-PP, extended with the secure boot component and the secure storage component that includes cryptography required for secure storage. The TOE is illustrated in red in Figure 20 with the additional components for the TOE compared to the base-PP represented with a '+' sign on the corner.

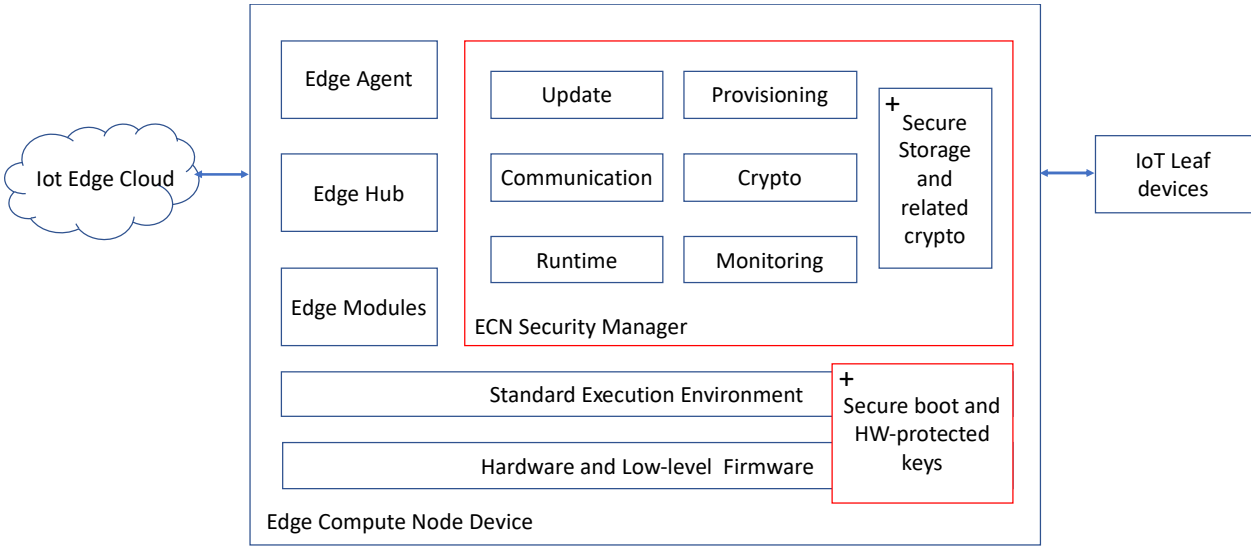


Figure 20 Edge Compute Node with Software-Based Secure Storage TOE

B.1.2.1 Usage and Major Security Features of a TOE

The additional security features for the TOE of this PP-module compared to the Base-PP include the following components:

- The **Secure storage and related crypto,** which protects user data at rest and provides secure storage of cryptographic keys and certificates.
- The **Secure boot and hardware-protected keys,** which authenticates executable code loaded from boot prior to its execution based on a hardware-protected certificate and provides hardware protection for the cryptographic keys used for secure storage. This low-level firmware and possibly related support from the Standard Execution Environment is outside of the ECN Security Manager and may be device-specific.

B.1.2.2 TOE Type

The TOE type is a combination of hardware and software components of an Edge Compute Node featuring a software security manager and hardware support for secure boot and secure storage.

B.1.2.3 Available non-TOE hardware/software/firmware

Compared to the base-PP, parts of the hardware and low-level firmware and supporting Operating System related to Secure boot and secure storage are now in the TOE.

The available non-TOE hardware/software/firmware then consists of:

- The parts of the supporting **Operating System** (Standard Execution Environment) for the TOE not in charge of the secure boot nor secure storage (which have been moved to the TOE).
- The **Edge Modules** that implement local edge computing functions for the network of leaf devices.
- The **Edge Hub** in charge of communications with the IoT Edge Cloud.
- The **Edge Agent** in charge of Edge module management.
- The parts of hardware and low-level firmware not in charge of the secure boot (which have been moved to the TOE).
- The networked environment with the IoT Edge Cloud and the leaf devices.

B.1.3 TOE Security Services

This section summarizes the additional security services provided by the TOE along with the ones inherited from the base-PP and detailed in section 1.3:

- **User Data Protection**: The TOE protects user data at rest and provides secure storage of cryptographic keys and certificates.
- **Secure boot**: The TOE authenticates executable code loaded from boot prior to its execution.

B.2 Conformance Claims

B.2.1 CC Conformance Claims

This PP-Module is CC Part 2 [CC2] extended and CC Part 3 [CC3] extended.

B.2.2 Conformance Claims of the PP

This PP does not claim conformance to any other PP.

B.2.3 Conformance Claims to a Package

This PP-Module inherits the package claims of its base-PP, as stated in Section 2.3.

B.2.4 Conformance Rationale

This PP-module does not provide a conformance rationale because it does not claim conformance to any other PP.

B.2.5 Conformance Statement

This PP-Module inherits from its base-PP the strict conformance as defined in [CC1] for all Security Targets and Protection Profiles claiming conformance to it.

B.2.6 Consistency Rationale

The consistency rationale is given in Section B.7.4.

B.3 Security Problem Definition

This PP-module extends the base-PP SPD.

B.3.1 Assets

Table *21* presents the additional assets that need to be protected by the TOE defined for this PP-module. All other assets from the base-PP apply to this PP-module.

Asset	Description
Secrets	The cryptographic secrets stored and managed by the TOE: • Root Encryption Key (REK), tied to the device used to encrypt other keys. • Key Encryption Key (KEK), used to encrypt other keys, such as DEKs or storage that contains keys. • Data Encryption Key (DEK), used to encrypt data-at-rest. • Other cryptographic private keys or symmetric keys.

	Properties: integrity and confidentiality

Table 21 Assets

B.3.2 Threats

Table *22* presents the additional known or presumed threats to protected resources that are addressed by the TOE. All other threats from the base-PP apply to this PP-module.

Threat	Description
T.STORAGE	A remote or local attacker may gain access to the secrets stored by the TOE and compromise them and indirectly data protected by those secrets. *Threatened assets:* Secrets *(confidentiality and integrity).*
T.BOOT	A remote or local attacker may attempt to tamper with the integrity of TOE software in order to insert and execute malicious code during the bootstrap process. *Threatened assets: All (confidentiality and integrity).*
T.PHYSICAL	A local attacker may attempt to access TOE assets, including secrets, by physically interacting with the TOE. For instance, the attacker may attempt to access the device through external hardware ports. *Threatened assets: All (confidentiality and integrity).*

Table 22 Threats

B.3.3 Organizational Security Policies

There are no organizational security policies for this PP-module.

B.3.4 Assumptions

All assumptions from the base-PP apply to this PP-module except for A.PHYSICAL, A.STORAGE and A.SECURE_BOOT that move to threats.

B.4 Security Objectives

This PP-module introduces two new TOE security objectives. All security objectives from the base-PP apply to this PP-module, except for OE.PHYSICAL which is superseded by the new TOE security objective O.PHYSICAL, OE. SECURE_BOOT superseded by O.SECURE_BOOT, and OE.STORAGE superseded by O.STORAGE.

B.4.1 TOE Security Objectives

Table 23 describes the additional security objectives for the TOE of this PP-Module.

Security Objective	Source
O.STORAGE	The TOE will provide data-at-rest protection and the capability of encrypting cryptographic keys and certificates managed by the TOE and Edge modules, to prevent unauthorized access to stored data.
O.SECURE_BOOT	The TOE will provide a secure boot feature which authenticates the TOE software during the bootstrap process.
O.PHYSICAL	The TOE will detect physical attacks that might compromise TOE assets.

Table 23 Security Objectives for the TOE

B.4.2 Security Objectives Rationale

The security objectives rationale for this PP-Module is based on the base-PP rationale defined in Section 4.3 updated due to the superseding and reassigned done in this PP-Module.

	O.COMMS	O.AUTH	O.CONFIG	O.INTEGRITY	O.STORAGE	O.SECURE_BOOT	O.PHYSICAL	OE.ADMIN	OE.KEYS	OE.PLATFORM	OE.NO_GENERAL_PURPOSE
T.EAVESDROP	X		X		X			X	X		
T.NETWORK	X	X	X		X			X	X		
T.FLAWMOD			X	X			X	X		X	X
T.PERSISTENT			X	X	X	X	X	X		X	X
T.STORAGE					X		X				
T.BOOT						X	X				
T.PHYSICAL							X				
A.ADMIN								X			
A.KEYS									X		
A.PLATFORM										X	
A.NO_GENERAL_PURPOSE											X

B.4.2.1 Security Objective Rationales: Threats

T.EAVESDROP: The combination of the following security objectives diminishes the eavesdropping of communication channels threat:

- O.COMMS ensures confidentiality of exchanged data through a secure communication channel using TLS.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG.
- OE.KEYS protects the keys and certificates used to communicate with the TOE outside of the TOE (i.e. trusted endpoints).
- O.STORAGE protects keys and certificates within the TOE.

T.NETWORK: The combination of the following security objectives diminishes the alteration of communication threat:

- O.COMMS ensures integrity of exchanged data through a secure communication channel using TLS.
- O.AUTH ensures authentication of communication with trusted end-points.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG.
- OE.KEYS protects the keys and certificates used to communicate with the TOE outside of the TOE (i.e. trusted endpoints).
- O.STORAGE protects keys and certificates within the TOE

T.FLAWMOD: The combination of the following security objectives diminishes the TOE compromise threat:

- O.INTEGRITY ensures integrity of critical functionality, software and updates and controls access to system services.
- O.PHYSICAL provides detection of physical attacks on the TOE.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG and ensures that the TOE is correctly configured and the underlying platform up-to-date.
- OE.PLATFORM provides OS support for domain separation and non-bypassability at the OS level mainly to protect TOE processes from other processes running in the OS with memory isolation and also includes anti-exploitation techniques to mitigate potential breaches.
- OE.NO_GENERAL_PURPOSE ensures that OS has not computing capabilities that could be used by an attacker reducing the exploitability of attacks.

T.PERSISTENT: The combination of the following security objectives diminishes the persistent access to the TOE threat:

- O.INTEGRITY ensures integrity of critical functionality, software/firmware and data and updates and controls access to system services.
- O.SECURE_BOOT provides support for authentication of the underlying platform code and the TOE.
- O.PHYSICAL provides detection of physical attacks on the TOE.
- O.STORAGE protects keys and certificates within the TOE.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG and ensures that the TOE is correctly configured and the underlying platform up-to-date.
- OE.PLATFORM provides OS support for domain separation and non-bypassability at the OS level mainly to protect TOE processes from other processes running in the OS with memory isolation and also includes anti-exploitation techniques to mitigate potential breaches.
- OE.NO_GENERAL_PURPOSE reduces the attack surface for the supporting OS.

T.STORAGE: The combination of the following security objectives diminishes this threat:

- O.STORAGE provides data-at-rest protection for cryptographic keys and certificates managed by the TOE and Edge modules.
- O.PHYSICAL provides detection of physical attacks on the TOE.

T.BOOT: The combination of the following security objectives diminishes this threat:

- O.SECURE_BOOT provides support for authentication of the underlying platform code and the TOE.
- O.PHYSICAL provides detection of physical attacks on the TOE.

T.PHYSICAL: The security objective O.PHYSICAL diminishes this threat.

B.4.2.2 Security Objective Rationales: Assumptions

A.ADMIN: The security objective for the environment OE.ADMIN directly upholds this assumption.

A.KEYS: The security objective for the environment OE.KEYS directly upholds this assumption.

A.PLATFORM: The security objective for the environment OE.PLATFORM directly upholds this assumption.

A.NO_GENERAL_PURPOSE: The security objective for the environment OE.NO_GENERAL_PURPOSE directly upholds this assumption.

B.5 Extended Components Definition

This protection profile makes use of extended components, not defined in [CC2]. These components are identified in Table 24, with the related requirement class from [CC2].

Requirement Class	Requirement Component	Similar Part 2 Requirement
Cryptographic Support (FCS)	Extended: Cryptographic Key Support for Root Encryption Key (FCS_CKM_EXT.1)	none
	Extended: Cryptographic Key Random Generation for Data Encryption Keys (FCS_CKM_EXT.2)	none
	Extended: Cryptographic Key Generation for Key Encryption Keys (FCS_CKM_EXT.3)	none
	Extended: Salt Generation (FCS_CKM_EXT.4)	none
	Extended: Initialization Vector Generation (FCS_CKM_EXT.5)	none
	Extended: Cryptographic Key Storage (FCS_STG_EXT.1)	none
	Extended: Encrypted Cryptographic Key Storage (FCS_STG_EXT.2)	none
	Extended: Encrypted Integrity of Cryptographic Key Storage (FCS_STG_EXT.3)	none
User Data Protection (FDP)	Extended: Data at Rest Encryption (FDP_DAR_EXT.1)	none
	Extended: Data at Rest Wipe (FDP_DAR_EXT.2)	none
Protection of the TSF (FPT)	Extended: TSF Integrity Testing (FPT_TST_EXT.2)	none
	Extended: Key Storage (FPT_KST_EXT.1)	none
	Extended: No Key Transmission (FPT_KST_EXT.2)	none
	Extended: No Plaintext Key Export (FPT_KST_EXT.3)	none

Table 24: Extended Security Functional Requirements

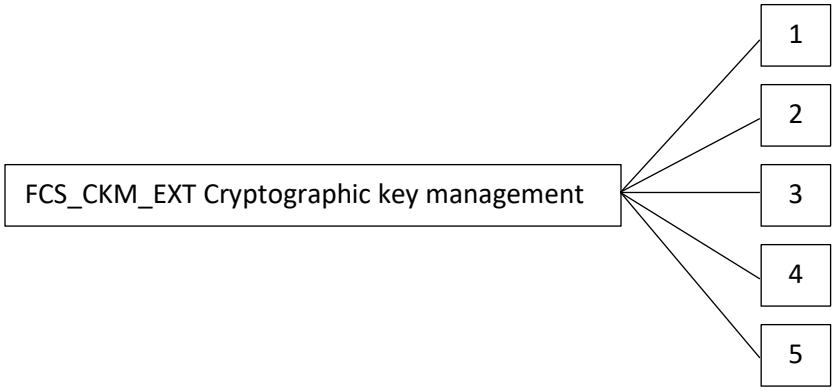
B.5.1.1 Cryptographic Support (FCS)

B.5.1.1.1 Definition of the Family FCS_CKM_EXT

The family FCS_CKM_EXT describes additional functional requirements for cryptographic key management.

Components Levelling:

This family has five levels.



B.5.1.1.1.1 Extended: Cryptographic Key Support for Root Encryption Key (FCS_CKM_EXT.1)

Hierarchical to: No other components.

Dependencies: FCS_RBG_EXT.1 Extended: Random Bit Generation

FCS_CKM_EXT.1.1	The TSF shall support a [selection: *mutable hardware-protected, immutable hardware-protected*] REK with a [selection: *symmetric, asymmetric*] key of strength [selection: *112 bits, 128 bits, 192 bits, 256 bits*].
FCS_CKM_EXT.1.2	A REK shall not be able to be read from or exported from the hardware.
FCS_CKM_EXT.1.3	Each REK shall be generated by a RBG in accordance with FCS_RBG_EXT.1.

Application Notes:

- The raw key material of “mutable hardware-protected” REK(s) is computationally processed by hardware and software can change or sanitize the raw key material but not read or export it.
- The raw key material of “immutable hardware-protected” REK(s) is computationally processed by hardware and software cannot access the raw key material. Thus if “immutable hardware-protected” is selected in FCS_CKM_EXT.1.1 it implicitly meets FCS_CKM_EXT.1.2.
- The TSS shall include a description of the generation mechanism including what triggers a generation, how the functionality described by FCS_RBG_EXT.1 is invoked, and whether a separate instance of the RBG is used for REK(s).

B.5.1.1.1.2 Extended: Cryptographic Key Random Generation for Data Encryption Keys (FCS_CKM_EXT.2)

Hierarchical to: No other components.

Dependencies: FCS_RBG_EXT.1 Extended: Random Bit Generation

FCS_CKM_EXT.2.1 All DEKs shall be randomly generated with entropy corresponding to the security strength (according to NIST SP 800-57) of AES key sizes of [selection: *128, 256*] bits.

B.5.1.1.1.3 Extended: Cryptographic Key Generation for Key Encryption Keys (FCS_CKM_EXT.3)

Hierarchical to: No other components.

Dependencies: FCS_CKM_EXT.1 Cryptographic Key Support for Root Encryption Key.

FCS_CKM_EXT.3.1 The TSF shall use [selection:

- *asymmetric KEKs of [assignment: security strength greater than or equal to 112 bits] security strength,*
- *symmetric KEKs of [selection: 128-bit, 256-bit] security strength corresponding to at least the security strength of the keys encrypted by the KEK*

].

FCS_CKM_EXT.3.2 The TSF shall generate all KEKs using one of the following methods:

[selection:

a) *generate the KEK using a key generation scheme that meets this profile (as specified in FCS_CKM_EXT.1)*
b) *Combine the KEK from other KEKs in a way that preserves the effective entropy of each factor by [selection: using an XOR operation, concatenating the keys and use a KDF (as described in SP 800-108), encrypting one key with another].*

].

Application Notes: The terms **"**Security strength" in FCS_CKM_EXT.3.1 have to be interpreted according to NIST SP 800-57.

B.5.1.1.1.4 Extended: Salt Generation (FCS_CKM_EXT.4)

Hierarchical to: No other components.

Dependencies: FCS_RBG_EXT.1 Extended: Random Bit Generation

FCS_CKM_EXT.4.1 The TSF shall generate all salts using a RBG that meets FCS_RBG_EXT.1.

B.5.1.1.2 Extended: Initialization Vector Generation (FCS_CKM_EXT.5)

Hierarchical to: No other components.

Dependencies: No dependencies.

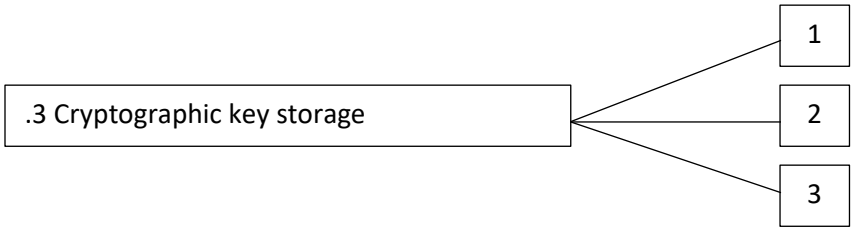
FCS_CKM_EXT.5.1 The TSF shall generate IVs in accordance with Table 46: References and IV Requirements for NIST-approved Cipher Modes.

B.5.1.1.3 Definition of the Family FCS_STG_EXT

The family FCS_STG_EXT describes the functional requirements for cryptographic key storage.

Components Levelling:

This family has three levels.



B.5.1.1.3.1 Extended: Cryptographic Key Storage (FCS_STG_EXT.1)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions.

FCS_STG_EXT.1.1 The TSF shall provide [selection: *hardware-based, software-based*] secure key storage for asymmetric private keys and [selection: *symmetric keys, persistent secrets, no other keys*].

FCS_STG_EXT.1.2 The TSF shall be capable of importing keys/secrets into the secure key storage upon request of [selection: *the user, the administrator*] and [selection: *applications running on the TSF, no other subject*].

FCS_STG_EXT.1.3 The TSF shall be capable of destroying keys/secrets in the secure key storage upon request of [selection: *the user, the administrator*].

FCS_STG_EXT.1.4 The TSF shall have the capability to allow only the application that imported the key/secret the use of the key/secret. Exceptions may only be explicitly authorized by [selection: *the user, the administrator, a common application developer*].

FCS_STG_EXT.1.5 The TSF shall allow only the application that imported the key/secret to request that the key/secret be destroyed. Exceptions may only be explicitly authorized by [selection: *the user, the administrator, a common application developer*].

Application Notes: The selection "*the user, the administrator*" in FCS_STG_EXT.1.2 and FCS_STG_EXT.1.3 must be consistent with the choice of supported management functions for user and administrator in Table 17 for the management functions "import keys/secrets into the secure key storage" and "destroy imported keys/secrets and any other keys/secrets in the secure key storage".

B.5.1.1.3.2 Extended: Encrypted Cryptographic Key Storage (FCS_STG_EXT.2)

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation.

FCS_STG_EXT.2.1 The TSF shall encrypt all DEKs and KEKs and [selection: *persistent TLS key material, all software-based key storage, no other keys*] by KEKs that are protected by the REK with [selection:

a. encryption by a REK,
b. encryption by a KEK chaining to a REK
c. *encryption by a KEK that is derived from a REK*].

FCS_STG_EXT.2.2 DEKs and KEKs and [selection: *persistent TLS key material, all software-based key storage, no other keys*] shall be encrypted using one of the following methods: [selection: *using a SP800-56B key establishment scheme, using AES in the [selection: Key Wrap (KW) mode, Key Wrap with Padding (KWP) mode, GCM, CCM, CBC mode*]].

B.5.1.1.3.3 Extended: Encrypted Integrity of Cryptographic Key Storage (FCS_STG_EXT.3)

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation.

FCS_STG_EXT.3.1 The TSF shall protect the integrity of any encrypted DEKs and KEKs and [selection: *persistent TLS key material, all software-based key storage, no other keys*] by [assignment: *list of cryptographic operations*].

FCS_STG_EXT.3.2 The TSF shall verify the integrity of the [selection: *hash, digital signature, MAC*] of the stored key prior to use of the key.

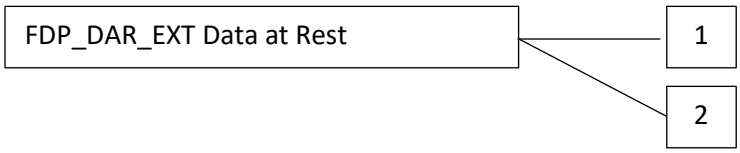
B.5.1.2 User Data Protection (FDP)

B.5.1.2.1 Definition of the Family FDP_DAR_EXT

The family FDP_DAR_EXT describes the functional requirements for data at rest protection purposes.

Components Levelling:

This family has two levels.



B.5.1.2.1.1 Extended: Data at Rest Encryption (FDP_DAR_EXT.1)

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation.

FDP_DAR_EXT.1.1 Encryption shall cover all protected data.

FDP_DAR_EXT.1.2 Encryption shall be performed using DEKs with AES in the [selection: *XTS, CBC, GCM*] mode with key size [selection: *128, 256*] bits.

B.5.1.2.1.2 Extended: Data at Rest Wipe (FDP_DAR_EXT.2)

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_DAR_EXT.2.1 The TSF shall wipe all protected data by [assignment: *data wipe procedure*].

FDP_DAR_EXT.2.2 The TSF shall perform a power cycle on conclusion of the wipe procedure.

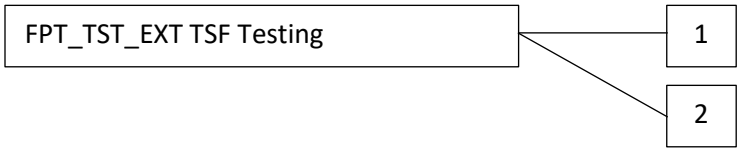
B.5.1.3 Protection of the TSF (FPT)

B.5.1.3.1 Definition of the Family FPT_TST_EXT

The family FPT_TST_EXT describes the functional requirements for TSF self-tests. An additional level is added to the one from the base-PP.

Components Levelling:

This family has two levels.



B.5.1.3.1.1 Extended: TSF Integrity Testing (FPT_TST_EXT.2)

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation
FCS_X509_EXT.1 Extended: Validation of Certificates
FCS_X509_EXT.2 Extended: X509 Certificate Authentication.

FPT_TST_EXT.2.1 The TSF shall verify the integrity of the bootchain up through the Standard Execution Environment, and [selection: *all executable code stored in mutable media, [assignment: list of other executable code], no other executable code*], stored in mutable media prior to its execution through the use of [selection: *a digital signature using an immutable hardware-protected asymmetric key, an immutable hardware-protected hash of an asymmetric key, an immutable hardware-protected hash, a digital signature using a mutable hardware-protected asymmetric key*].

FPT_TST_EXT.2.2 The TSF shall not execute code if the code signing certificate is deemed invalid.

Application Notes:

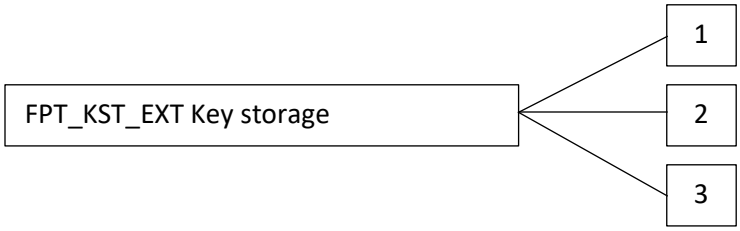
- 'Immutable hardware-protected asymmetric keys' cannot be changed, sanitized, read or exported by software.
- 'Mutable hardware-protected asymmetric keys' can be changed or sanitized by software but not read or exported.
- 'Immutable hardware-protected asymmetric hashes' cannot be changed, sanitized, read or exported by software.
- In this SFR, the term 'hardware' relates to the 'Secure boot and hardware-protected keys' components as defined in the TOE Overview, Section B.1.2.

B.5.1.3.2 Definition of the Family FPT_KST_EXT

The family FPT_KST_EXT describes the functional requirements for key storage purposes.

Components Levelling:

This family has three levels.



B.5.1.3.2.1 Extended: Key Storage (FPT_KST_EXT.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.1.1 The TSF shall not store any plaintext key material in readable nonvolatile memory.

B.5.1.3.2.2 Extended: No Key Transmission (FPT_KST_EXT.2)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.2.1 The TSF shall not transmit any plaintext key material outside the security boundary of the TOE.

B.5.1.3.2.3 Extended: No Plaintext Key Export (FPT_KST_EXT.3)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.3.1 The TSF shall ensure it is not possible for the TOE user(s) to export plaintext keys.

B.6 Security Requirements

The section defines the Security Functional Requirements (SFRs) for the TOE.

B.6.1 TOE Security Functional Requirements

This PP-module introduces or refines from the base-PP the following SFRs. All other SFRs from the base-PP also apply to this PP-module.

Requirement Class	Requirement Component	Relation to base-PP
Security Audit (FAU)	Audit Data Generation (FAU_GEN.1)	Refinement
Cryptographic Support (FCS)	Extended: Cryptographic Key Support for Root Encryption Key (FCS_CKM_EXT.1)	New
	Extended: Cryptographic Key Random Generation for Data Encryption Keys (FCS_CKM_EXT.2)	New
	Extended: Cryptographic Key Generation for Key Encryption Keys (FCS_CKM_EXT.3)	New
	Cryptographic Key Destruction (FCS_CKM.4(Storage))	New
	Extended: Salt Generation (FCS_CKM_EXT.4)	New
	Extended: Initialization Vector Generation (FCS_CKM_EXT.5)	New
	Extended: Cryptographic Key Storage (FCS_STG_EXT.1)	New
	Extended: Encrypted Cryptographic Key Storage (FCS_STG_EXT.2)	New
	Extended: Encrypted Integrity of Cryptographic Key Storage (FCS_STG_EXT.3)	New
User Data Protection (FDP)	Extended: Data at Rest Encryption (FDP_DAR_EXT.1)	New
	Extended: Data at Rest Wipe (FDP_DAR_EXT.2)	New
Protection of the TSF (FPT)	Extended: TSF Integrity Testing (FPT_TST_EXT.2)	New
	Extended: Key Storage (FPT_KST_EXT.1)	New
	Extended: No Key Transmission (FPT_KST_EXT.2)	New
	Extended: No Plaintext Key Export (FPT_KST_EXT.3)	New
	Passive Detection of Physical Attack (FPT_PHP.1)	New

Table 25 TOE Security Functional Requirements

B.6.1.1 Security Audit (FAU)

B.6.1.1.1 Audit Data Generation (FAU_GEN.1)

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and
[
c) Administrator management functions, as defined in the fourth column of Table 17;
d) Start-up and shutdown of the OS;
e) Specifically defined auditable events in Table 9 and Table 26;
f) [assignment: *other specifically defined auditable events*]].

FAU_GEN.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, [**additional information in Table 9 and Table 26**].

Requirement	Auditable Events	Additional Record Contents
FCS_CKM_EXT.1	Generation of a REK	No additional Information.
FCS_CKM_EXT.2	None.	
FCS_CKM_EXT.3	None.	
FCS_CKM.4(Storage)	None.	
FDP_DAR_EXT.2	Success or failure of the wipe.	No additional Information.
FCS_CKM_EXT.4	None.	
FCS_CKM_EXT.5	None.	
FCS_STG_EXT.1	Import or destruction of key.	Identity of key. Role and identity of requestor.
FCS_STG_EXT.2	None.	

FCS_STG_EXT.3	Failure to verify integrity of stored key.	Identity of key being verified.
FDP_DAR_EXT.1	Failure to encrypt/decrypt data.	No additional information.
FPT_TST_EXT.2	Start-up of TOE.	Boot Mode.
	Detected integrity violations.	The TSF code that caused the integrity violation.
FPT_KST_EXT.1	None.	
FPT_KST_EXT.2	None.	
FPT_KST_EXT.3	None.	
FPT_PHP.1	Detected of physical tampering.	The detected event.

Table 26 Auditable Events

B.6.1.2 Cryptographic Support (FCS)

B.6.1.2.1 Extended: Cryptographic Key Support for Root Encryption Key (FCS_CKM_EXT.1)

As in Section B.5.

Application Notes: In this SFR, the term 'hardware' relates to the 'Secure boot and hardware-protected keys' components as defined in the TOE Overview, Section B.1.2.

B.6.1.2.2 Extended: Cryptographic Key Random Generation for Data Encryption Keys (FCS_CKM_EXT.2)

As in Section B.5.

B.6.1.2.3 Extended: Cryptographic Key Generation for Key Encryption Keys (FCS_CKM_EXT.3)

As in Section B.5.

B.6.1.2.4 Extended: Salt Generation (FCS_CKM_EXT.4)

As in Section B.5.

B.6.1.2.5 Extended: Initialization Vector Generation (FCS_CKM_EXT.5)

As in Section B.5.

B.6.1.2.6 Cryptographic Key Destruction (FCS_CKM.4(Storage))

FCS_CKM.4.1(Storage) The TSF shall destroy cryptographic keys in accordance with the specified cryptographic key destruction methods [**by clearing the KEK encrypting the target key and destroying all plaintext keying material and critical security parameters when no longer needed**] that meets the following: [**none**]

Application Note: Depending of the type of memory, key destruction can be performed by one of the following methods, to be specified in TSS.

- For volatile memory, the destruction shall be executed by a single direct overwrite consisting of a pseudo-random pattern using the TSF's RBG or consisting of zeroes.
- For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), followed by a read-verify.
- For non-volatile flash memory, that is not wear-leveled, the destruction shall be executed by a single direct overwrite consisting of zeros followed by a read-verify or by a block erase that erases the reference to memory that stores data as well as the data itself.
- For non-volatile flash memory, that is wear-leveled, the destruction shall be executed by a single direct overwrite consisting of zeros or by a block erase.
- For non-volatile memory other than EEPROM and flash, the destruction shall be executed by a single direct overwrite with a random pattern that is changed before each write.

B.6.1.2.7 Extended: Cryptographic Key Storage (FCS_STG_EXT.1)

As in Section B.5.

Application Note: The term "Application" in FCS_STG_EXT.1 has to be interpreted as "Edge module".

B.6.1.2.8 Extended: Encrypted Cryptographic Key Storage (FCS_STG_EXT.2)

As in Section B.5.

B.6.1.2.9 Extended: Encrypted Integrity of Cryptographic Key Storage (FCS_STG_EXT.3)

FCS_STG_EXT.3.1 The TSF shall protect the integrity of any encrypted DEKs and KEKs and [selection: *persistent TLS key material, all software-based key storage, no other keys*] by **[selection:**

- ***[selection: GCM, CCM, Key Wrap, Key Wrap with Padding] cipher mode for encryption according to FCS_STG_EXT.2;***
- ***a hash (FCS_COP.1(HASH)) of the stored key that is encrypted by a key protected by FCS_STG_EXT.2;***
- ***a keyed hash (FCS_COP.1(HMAC)) using a key protected by a key protected by FCS_STG_EXT.2;***
- ***a digital signature of the stored key using an asymmetric key protected according to FCS_STG_EXT.2*****]**.

FCS_STG_EXT.3.2 The TSF shall verify the integrity of the [selection: *hash, digital signature, MAC*] of the stored key prior to use of the key.

B.6.1.3 User Data Protection (FDP)

B.6.1.3.1 Extended: Data at Rest Encryption (FDP_DAR_EXT.1)

As in Section B.5.

B.6.1.3.2 Extended: Data at Rest Wipe (FDP_DAR_EXT.2)

FDP_DAR_EXT.2.1 The TSF shall wipe all protected data by **[selection:**

- ***Cryptographically erasing the encrypted DEKs and/or the KEKs in non-volatile memory by following the requirements in FCS_CKM.4 (Storage);***
- ***Overwriting all protected data according to the following rules:***
  - ***For EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1, followed by a read-verify.***
  - ***For flash memory, that is not wear-leveled, the destruction shall be executed [selection: by a single direct overwrite consisting of zeros followed by a read- verify, by a block erase that erases the reference to memory that stores data as well as the data itself].***
  - ***For flash memory, that is wear-leveled, the destruction shall be executed [selection: by a single direct overwrite consisting of zeros, by a block erase].***
  - ***For non-volatile memory other than EEPROM and flash, the destruction shall be executed by a single direct overwrite with a random pattern that is changed before each write.*****]**

FDP_DAR_EXT.2.2 The TSF shall perform a power cycle on conclusion of the wipe procedure.

B.6.1.4 Protection of the TSF (FPT)

B.6.1.4.1 Extended: TSF Integrity Testing (FPT_TST_EXT.2)

FPT_TST_EXT.2.1 The TSF shall verify the integrity of the bootchain up through the Standard Execution Environment, and [***all executable code stored in mutable media***], stored in mutable media prior to its execution through the use of [selection: *a digital signature using an immutable hardware asymmetric key, an immutable hardware hash of an asymmetric key, an immutable hardware hash, a digital signature using a mutable hardware asymmetric key*].

FPT_TST_EXT.2.2 The TSF shall not execute code if the code signing certificate is deemed invalid.

B.6.1.4.2 Extended: Key Storage (FPT_KST_EXT.1)

As in Section B.5.

B.6.1.4.3 Extended: No Key Transmission (FPT_KST_EXT.2)

As in Section B.5.

B.6.1.4.4 Extended: No Plaintext Key Export (FPT_KST_EXT.3)

As in Section B.5.

B.6.1.4.5 Passive detection of physical attack (FPT_PHP.1)

FPT_PHP.1.1	The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.
FPT_PHP.1.2	The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

B.7 Rationale for Security Requirements

This section provides a rationale for the security functional requirements and security assurance requirements.

B.7.1 Security Functional Requirements

The mapping presented in Table 27 traces the SFRs from this PP-module and related SFRs from the base-PP back to the O.STORAGE and O.INTEGRITY security objectives and demonstrates how this security objectives are met by the SFRs.

Table 27 Rationale for SFRs

SFR	O.STORAGE	O.INTEGRITY	O.PHYSICAL	O.SECURE_BOOT
FAU_GEN.1		X		
FCS_CKM_EXT.1	X			
FCS_CKM_EXT.2	X			
FCS_CKM_EXT.3	X			
FCS_CKM_EXT.4	X			
FCS_CKM_EXT.5	X			
FCS_CKM.4(Storage)	X			
FCS_COP.1(SYM)	X			
FCS_COP.1(HASH)	X			
FCS_COP.1(SIGN)	X			
FCS_COP.1(HMAC)	X			
FCS_RBG_EXT.1	X			

SFR	O.STORAGE	O.INTEGRITY	O.PHYSICAL	O.SECURE_BOOT
FCS_STG_EXT.1	X			
FCS_STG_EXT.2	X			
FCS_STG_EXT.3	X			
FDP_DAR_EXT.1	X			
FDP_DAR_EXT.2	X			
FPT_TST_EXT.2				X
FPT_KST_EXT.1	X			
FPT_KST_EXT.2	X			
FPT_KST_EXT.3	X			
FPT_PHP.1			X	

The rationale for mapping for **O.INTEGRITY** as presented in Section 7.1 is modified as follows:

- FAU_GEN.1 now refers to the SFR from the PP-module, which refines the one from the base-PP.

O.STORAGE is addressed by requirements FDP_DAR_EXT.1 and FDP_DAR_EXT.2 that provide secure storage features (read, write, delete) for protected data. Cryptographic support for encryption is provided by requirements FCS_CKM_EXT.1, FCS_CKM_EXT.2, FCS_CKM_EXT.3, FCS_CKM_EXT.4, FCS_CKM_EXT.5 and FCS_CKM.4(Storage) for generation and destruction of the cryptographic keys; and by FCS_COP.1(SYM), FCS_COP.1(HASH), FCS_COP.1(SIGN), FCS_COP.1(HMAC) and FCS_RBG_EXT.1 for related cryptographic operations and random number generation. Storage of encryption keys is addressed by requirements FCS_STG_EXT.1, FCS_STG_EXT.2 and FCS_STG_EXT.3. Finally, protection of encryption keys in plaintext is addressed by requirements FPT_KST_EXT.1, FPT_KST_EXT.2 and FPT_KST_EXT.3.

O.SECURE_BOOT is directly addressed by requirement FPT_TST_EXT.2 that verifies integrity of the bootchain for all executable code stored in mutable media.

O.PHYSICAL is addressed by requirement FPT_PHP.1 that provides passive detection of physical attacks.

The rationale for mapping for **O.COMMS**, **O.AUTH** and **O. CONFIG** as presented in Section 7.1 still applies as no changes are introduced by the PP-module over those objectives.

B.7.2 Security Requirements Dependency Analysis

SFR	Dependencies	Resolved
FAU_GEN.1	FPT_STM.1	Yes: FPT_STM.1 from base-PP
FCS_CKM_EXT.1	FCS_RBG_EXT.1	Yes: FCS_RBG_EXT.1 from base-PP

SFR	Dependencies	Resolved
FCS_CKM_EXT.2	FCS_RBG_EXT.1	Yes: FCS_RBG_EXT.1 from base-PP
FCS_CKM_EXT.3	FCS_CKM_EXT.1	Yes: FCS_CKM_EXT.1
FCS_CKM.4(Storage)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Yes: FCS_CKM.1
FCS_CKM_EXT.4	FCS_RBG_EXT.1	Yes: FCS_RBG_EXT.1 from base-PP
FCS_CKM_EXT.5	No dependencies	
FCS_STG_EXT.1	FMT_SMR.1 FMT_SMF.1	Yes: FMT_SMR.1 Yes: FMT_SMF.1
FCS_STG_EXT.2	FCS_COP.1	Yes: FCS_COP.1(SYM) from base-PP
FCS_STG_EXT.3	FCS_COP.1	Yes: FCS_COP.1(SYM) from base-PP, FCS_COP.1(HASH) from base-PP, FCS_COP.1(HMAC) from base-PP
FDP_DAR_EXT.1	FCS_COP.1	Yes: FCS_COP.1(SYM) from base-PP
FDP_DAR_EXT.2	No dependencies	
FPT_TST_EXT.2	FCS_COP.1 FCS_X509_EXT.1 FCS_X509_EXT.2	Yes: FCS_COP.1(SIGN) from base-PP, FCS_COP.1(HASH) from base-PP, FCS_COP.1(HMAC) from base-PP Yes: FCS_X509_EXT.1 from base-PP Yes: FCS_X509_EXT.2 from base-PP
FPT_KST_EXT.1	No dependencies	
FPT_KST_EXT.2	No dependencies	
FPT_KST_EXT.3	No dependencies	
FPT_PHP.1	No dependencies	

B.7.3 Security Assurance Requirements

The rationale for security assurance requirements (SARs) is identical to the one of the base-PP, section 7.2.

B.7.4 Consistency Rationale

This PP-Module extends the Base-PP with local secure storage for protected data, including cryptographic keys, and secure boot.

This PP-Module extends the Base-PP TOE type with hardware components of the Edge Compute Node Device.

This PP-Module refines the assets of the Base-PP by specifying additional secrets, used for secure storage, that the TOE shall ensure integrity and confidentiality.

This PP-Module refines SFR FAU_GEN.1 by specifying additional events to be audited. It adds new SFRs which are dedicated to the management of secure storage and secure boot.

Appendix C Support for HSM-Based Secure Storage and Cryptography PP-Module

C.1 PP-Module Introduction

This PP-module must be flattened with the base-PP for the configuration called **Edge Compute Node with Support for HSM-Based Secure Storage and Cryptography**, identified in Section E.2, using the content of this Appendix.

C.1.1 Protection Profile, TOE, and Common Criteria (CC) Identification

PP-Module Title: **Support for HSM-Based Secure Storage and Cryptography PP-Module**

Related Base-PP Title: Edge Compute Node Protection Profile

PP-Module Version: version **1.0**.7, September 4th, 2020

CC Identification: CC for Information Technology (IT) Security Evaluation, Version 3.1, Revision 5, April 2017.

C.1.2 TOE Overview

This PP-Module extends the Base-PP with a secure boot feature and a secure storage for protected data (data-at-rest protection) supported by a HSM located in the operational environment of the TOE. The related TOE is composed of the ECN Security Manager, as in the Base-PP, extended with support of the interaction with the HSM. The TOE is illustrated in red in Figure 28 where the additional components for the TOE compared to the base-PP are represented with a '+' sign on the corner.

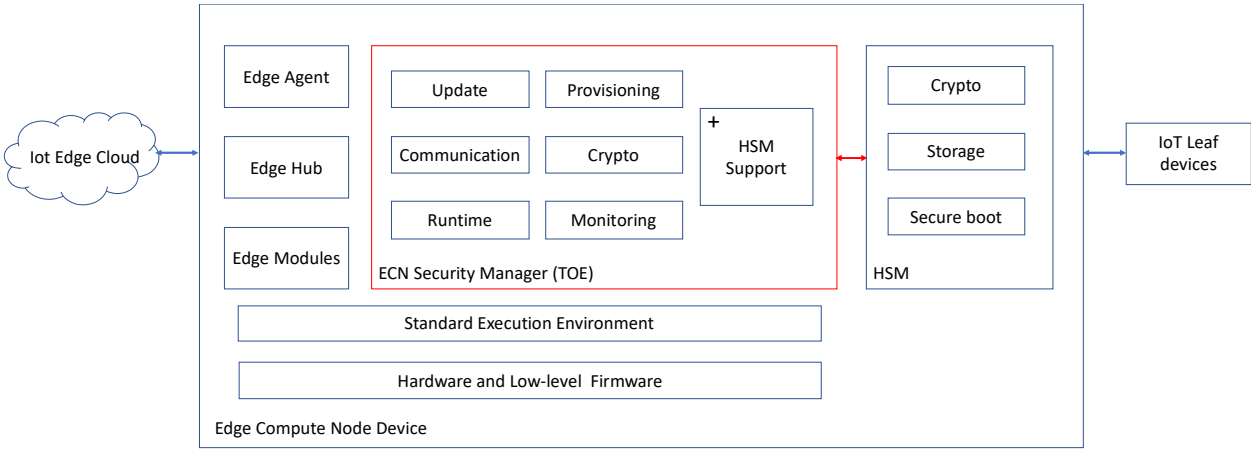


Figure 28 Edge Compute Node with HSM-Based Secure Storage and Cryptography TOE

C.1.2.1 Usage and Major Security Features of a TOE

The additional security feature for the TOE of this PP-module compared to the Base-PP includes the following:

- Secure communication with trusted IT product (HSM).

C.1.2.2 TOE Type

The TOE type is a software featuring the security manager for Edge Compute Node extended with secure communication with a trusted IT product.

C.1.2.3 Available non-TOE hardware/software/firmware

Compared to the base-PP, the non-TOE hardware/software/firmware is extended with a Hardware Security Module (HSM) peripheral, such as Trusted Platform Module (TPM) or a Dedicated Security Component (DSC).

This HSM is used as a root of trust for the TOE and is responsible for:

- Contributing to the secure boot of the platform and the TOE, by measuring executable code prior to execution and comparing this measure to a reference value;
- Managing sensitive assets for the TOE, in particular cryptographic keys and certificates;
- Offering cryptographic operation services to the TOE, based on the keys managed by the HSM.

C.2 Conformance Claims

C.2.1 CC Conformance Claims

This PP-Module is CC Part 2 [CC2] extended and CC Part 3 [CC3] extended.

C.2.2 Conformance Claims of the PP

This PP does not claim conformance to any other PP.

C.2.3 Conformance Claims to a Package

This PP-Module inherits the package claims of its base-PP, as stated in Section 2.3.

C.2.4 Conformance Rationale

This PP-module does not provide a conformance rationale because it does not claim conformance to any other PP.

C.2.5 Conformance Statement

This PP-Module inherits from its base-PP the strict conformance as defined in [CC1] for all Security Targets and Protection Profiles claiming conformance to it.

C.2.6 Consistency Rationale

The consistency rationale is given in Section C.6.4.

C.3 Security Problem Definition

This PP-module extends the base-PP SPD with a new threat and a new assumption that supersedes A.SECURE_BOOT and A.STORAGE from the base-PP. All SPD elements from the base-PP apply to this PP-module.

C.3.1 Assets

Table 29 presents the additional asset that need to be protected by the TOE, compared to the Base-PP.

Asset	Description
HSM data	Data exchanged between the TOE and the HSM. *Properties: integrity, confidentiality*

Table 29 Assets

C.3.2 Threats

Table 30 presents the additional known or presumed threats to protected resources that are addressed by the TOE.

Threat	Description
T.HSM_COMM	A local or remote attacker may attempt to illegally access or modify HSM data exchanged between the TOE and HSM. *Threatened assets: HSM data (confidentiality and integrity).*

Table 30: Threats

C.3.3 Organizational Security Policies

There are no organizational security policies for this PP-module.

C.3.4 Assumptions

Table 31 presents the additional condition that is assumed to exist in an environment where the TOE is employed. This assumption supersedes A.SECURE_BOOT and A.STORAGE from the base-PP.

Assumption	Description
A.HSM	It is assumed that the OS provides data-at-rest protection feature for cryptographic keys and certificates used by the TOE in combination with a HSM.

	It is assumed that the HSM is used as a root of trust by the TOE for the operations described in Section C.1.2.3 (secure boot, cryptographic operation services). It is assumed that the HSM is FIPS 140-2 or FIPS 140-3 certified. It is also assumed that the HSM is certified at least EAL3 augmented with ALC_FLR.1 and AVA_VAN.3 according to either: • [TPM PP] TCG, *Protection Profile for PC Client Specific TPM 2.0*, 16 June 2018, Version 1.1. • [DSC PP] *collaborative Protection Profile for Dedicated Security Component*, May 1st 2019, Version 1.0d.

Table 31: Assumptions

C.4 Security Objectives

This PP-module introduces a new security objective for the TOE and a new security objective for the Operational Environment. All security objectives from the base-PP apply to this PP-module except for the security objectives for the Operational Environment OE.SECURE_BOOT and OE.STORAGE from the base-PP which are superseded by OE.HSM.

C.4.1 TOE Security Objectives

Table 32 describes the additional TOE security objective for the TOE of this PP-module.

Environment Objective	Description
O.HSM_COMM	The TOE will provide the capability to communicate with the HSM using a trusted channel as a means to maintain the confidentiality and integrity of data that are transmitted between the TOE and HSM.

Table 32 TOE Security Objectives of the HSM-Based PP-Module

C.4.2 Security Objectives for the Operational Environment

Table 33 describes the additional security objectives for the operational environment of this PP-module.

Environment Objective	Description
OE.HSM	The OS provides data-at-rest protection feature for cryptographic keys and certificates used by the TOE in combination with a HSM. The HSM is used as a root of trust by the TOE for the operations described in Section C.1.2.3 (secure boot, cryptographic operation services). The HSM is FIPS 140-2 or FIPS 140-3 certified.

	The HSM is also certified at least EAL3 augmented with ALC_FLR.1 and AVA_VAN.3 according to either: • [TPM PP] TCG, *Protection Profile for PC Client Specific TPM 2.0*, 16 June 2018, Version 1.1. • [DSC PP] *collaborative Protection Profile for Dedicated Security Component*, May 1st 2019, Version 1.0d.

Table 33 Security Objectives for the Operational Environment of the HSM-Based PP-Module

C.4.3 Security Objectives Rationale

This Section gives an evidence for sufficiency and necessity of the defined objectives. The following table provides an overview for security objectives coverage (TOE and its environment).

	O.COMMS	O.AUTH	O.CONFIG	O.INTEGRITY	O.HSM_COMM	OE.ADMIN	OE.KEYS	OE.PLATFORM	OE.HSM	OE.NO_GENERAL_PURPOSE	OE.PHYSICAL
T.EAVESDROP	X		X		X	X	X		X		
T.NETWORK	X	X	X		X	X	X		X		
T.FLAWMOD			X	X		X		X		X	X
T.PERSISTENT			X	X	X	X		X	X	X	X
T.HSM_COMM					X						X
A.ADMIN						X					
A.KEYS							X				
A.PLATFORM								X			
A.HSM									X		
A.NO_GENERAL_PURPOSE										X	
A.PHYSICAL											X

C.4.3.1 Security Objective Rationales: Threats

T.EAVESDROP: The combination of the following security objectives diminishes the eavesdropping of communication channels threat:

- O.COMMS ensures confidentiality of exchanged data through a secure communication channel such as TLS.
- O.HSM_COMM ensures a trusted channel with the HSM that protects cryptographic keys used for secure communication channel.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG.
- OE.KEYS and OE.HSM protect the keys and certificates, used to communicate with the TOE, outside of the TOE (i.e. trusted endpoints and underlying platform, respectively).

T.NETWORK: The combination of the following security objectives diminishes the alteration of communication threat:

- O.COMMS ensures integrity of exchanged data through a secure communication channel such as TLS.
- O.HSM_COMM ensures a trusted channel with the HSM that protects cryptographic keys used for secure communication channel.
- O.AUTH ensures authentication of communication with trusted end-points.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG.
- OE.KEYS and OE.HSM protect the keys and certificates used to communicate with the TOE outside of the TOE (i.e. trusted endpoints and underlying platform, respectively).

T.FLAWMOD: The combination of the following security objectives diminishes the TOE compromising threat:

- O.INTEGRITY ensures integrity of critical functionality, software and updates and controls access to system services.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG and ensures that the TOE is correctly configured and the underlying platform up-to-date.
- OE.PLATFORM provides OS support for domain separation and non-bypassability at the OS level mainly to protect TOE processes from other processes running in the OS with memory isolation and also includes anti-exploitation techniques to mitigate potential breaches.
- OE.NO_GENERAL_PURPOSE ensures that OS has not computing capabilities that could be used by an attacker reducing the exploitability of attacks.
- OE.PHYSICAL provides physical protection for the TOE against attackers with physical access to the TOE.

T.PERSISTENT: The combination of the following security objectives diminishes the persistent access to the TOE threat:.

- O.INTEGRITY ensures integrity of critical functionality, software/firmware and data and updates and controls access to system services.

- O.HSM_COMM ensures a trusted channel with the HSM that protects cryptographic keys used code authentication.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG and ensures that the TOE is correctly configured and the underlying platform up-to-date.
- OE.PLATFORM provides OS support for domain separation and non-bypassability at the OS level mainly to protect TOE processes from other processes running in the OS with memory isolation and also includes anti-exploitation techniques to mitigate potential breaches.
- OE.HSM provides support for authentication of the underlying platform code and the TOE.
- OE.NO_GENERAL_PURPOSE ensures that OS has not computing capabilities that could be used by an attacker reducing the exploitability of attacks.
- OE.PHYSICAL provides physical protection for the TOE against attackers with physical access to the TOE.

T.HSM_COMM The combination of the following security objectives diminishes this threat:

- O.HSM_COMM that provides a trusted communication channel with the HSM.
- OE.PHYSICAL provides physical protection for the TOE against attackers with physical access to the TOE.

C.4.3.2 Security Objective Rationales: Assumptions

A.ADMIN: The security objective for the environment OE.ADMIN directly upholds this assumption.

A.KEYS: The security objective for the environment OE.KEYS directly upholds this assumption.

A.PLATFORM: The security objective for the environment OE.PLATFORM directly upholds this assumption.

A.HSM: The security objective for the environment OE.HSM directly upholds this assumption.

A.NO_GENERAL_PURPOSE: The security objective for the environment OE.NO_GENERAL_PURPOSE directly upholds this assumption.

A.PHYSICAL: The security objective for the environment OE.PHYSICAL directly upholds this assumption.

C.5 Security Requirements

The section defines the Security Functional Requirements (SFRs) for the TOE.

C.5.1 TOE Security Functional Requirements

This PP-module introduces or refines from the base-PP the following SFRs. All other SFRs from the base-PP apply to this PP-module.

Requirement Class	Requirement Component	Relation to base-PP

Security Audit (FAU)	Audit Data Generation (FAU_GEN.1)	Refinement
Protection of the TSF (FPT)	Extended: Self-Test Failure (FPT_FLS_EXT.1)	Refinement
	Inter-TSF basic TSF data consistency (FPT_TDC.1)	New
	Testing of external entities (FPT_TEE.1)	New
Trusted Path/Channels (FTP)	Inter-TSF Trusted Channel (FTP_ITC.1(HSM))	New

Table 34 TOE Security Functional Requirements

C.5.1.1 Security Audit (FAU)

C.5.1.1.1 Audit Data Generation (FAU_GEN.1)

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and
[
c) Administrator management functions, as defined in the fourth column of Table 17;
d) Start-up and shutdown of the OS;
e) Specifically defined auditable events in Table 9 and Table 35;
f) [assignment: *other specifically defined auditable events*]].

FAU_GEN.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, [**additional information in Table 9 and Table 35**].

Requirement	Auditable Events	Additional Record Contents
FPT_FLS_EXT.1	Measurement of TSF software.	Integrity verification value.
FPT_TDC.1	Failure of data consistency checks.	
FPT_TEE.1	Initiation of external entity test. Failure of external entity test.	None
FTP_ITC.1(HSM)	None	None

Table 35 Auditable Events

C.5.1.2 Protection of the TSF (FPT)

C.5.1.2.1 Extended: Self-Test Failure (FPT_FLS_EXT.1)

FPT_FLS_EXT.1.1 The TSF shall transition to non-operational mode, log failures in the audit record and [selection: *notify the administrator, [assignment: other actions], no other actions*] when the following types of failures occur:

- failures of the self-test(s)
- TSF software integrity verification failures
- HSM integrity verification failures
- [selection: *no other failures, [assignment: other failures]*].

C.5.1.2.2 Inter-TSF basic TSF data consistency (FPT_TDC.1)

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret [**data exchanged with the HSM**] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use [**specification of HSM commands / responses**] when interpreting the TSF data from another trusted IT product.

C.5.1.2.3 Testing of external entities (FPT_TEE.1)

FPT_TEE.1.1 The TSF shall run a suite of tests [selection: *during initial start-up, periodically during normal operation, at the request of an authorised user, [assignment: other conditions]*] to check the fulfillment of [**integrity of the HSM**].

FPT_TEE.1.2 If the test fails, the TSF shall [**perform actions in FPT_FLS_EXT.1, [assignment: *action(s)*]**].

Application Note: In order to check integrity of the HSM, the TSF can for instance check ID of the HSM, use HSM attestation service, read integrity registers, check tamper-detection registers, perform known answer tests for cryptographic operations.

C.5.1.3 Trusted Path / Channels (FTP)

C.5.1.3.1 Inter-TSF Trusted Channel (FTP_ITC.1(HSM))

FTP_ITC.1.1(HSM) The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2(HSM) The TSF shall permit [**the TSF**] to initiate communication via the trusted channel.

FTP_ITC.1.3(HSM) The TSF shall initiate communication via the trusted channel for [**all cryptographic and secure storage functions provided by the HSM**].

C.6 Rationale for Security Requirements

This section provides a rationale for the security functional requirements and security assurance requirements.

C.6.1 Security Functional Requirements

The mapping presented in Table 36 traces the SFR from this PP-module back to the O.INTEGRITY and O.HSM_COMM security objectives and demonstrates how these security objectives are met by the SFRs.

Table 36 Rationale for SFRs

SFR	O.INTEGRITY	O.HSM_COMM
FAU_GEN.1	X	
FPT_FLS_EXT.1	X	
FPT_TDC.1		X
FPT_TEE.1	X	
FTP_ITC.1(HSM)		X

The rationale for mapping for **O.INTEGRITY** as presented in Section 7.1 is modified as follows:

- FAU_GEN.1 and FPT_FLS_EXT.1 now refer to the SFRs from the PP-module, which refine the ones from the base-PP.
- TOE integrity is now also addressed by requirement FPT_TEE.1 that check integrity of the HSM, as an incorrect data from the HSM may corrupt the TOE.

O.HSM_COMM is addressed by requirements FTP_ITC.1(HSM) which provides a trusted channel between the TOE and the HSM and FPT_TDC.1 which verifies consistency of data exchanged between the HSM and the TOE.

The rationale for mapping for **O.COMMS**, **O.AUTH** and **O. CONFIG** as presented in Section 7.1 still applies as no changes are introduced by the PP-module over those objectives.

C.6.2 Security Requirements Dependency Analysis

SFR	Dependencies	Resolved
FAU_GEN.1	FPT_STM.1	Yes: FPT_STM.1 from base-PP
FPT_FLS_EXT.1	FAU_GEN.1	Yes: FAU_GEN.1
FPT_TDC.1	No dependencies	
FPT_TEE.1	No dependencies	
FTP_ITC.1(HSM)	No dependencies	

C.6.3 Security Assurance Requirements

The rationale for security assurance requirements (SARs) is identical to the one of the base-PP, section 7.3.

C.6.4 Consistency Rationale

This PP-Module extends the Base-PP with additional support of cryptographic operations and secure storage managed by a HSM as an external entity.

This PP-Module has the same TOE type as the Base-PP.

This PP-Module refines SFR FAU_GEN.1 by specifying additional events to be audited. It adds new SFRs which are dedicated to verification of integrity of HSM and data exchanged with HSM.

Appendix D Support for Secure Enclave Secure Storage and Cryptography PP-Module

D.1 PP-Module Introduction

This PP-module must be flattened with the base-PP for the configuration called **Edge Compute Node with Support for Secure Enclave-Based Secure Storage and Cryptography**, identified in Section E.3, using the content of this Appendix.

D.1.1 Protection Profile, TOE, and Common Criteria (CC) Identification

PP-Module Title: **Support for Secure Enclave-Based Secure Storage and Cryptography PP-Module**

Related Base-PP Title: Edge Compute Node Protection Profile

PP-Module Version: version **1.0**.7, September 4th, 2020

CC Identification: CC for Information Technology (IT) Security Evaluation, Version 3.1, Revision 5, April 2017.

D.1.2 TOE Overview

This PP-Module extends the Base-PP with a secure boot feature and a secure storage for protected data (data-at-rest protection) supported by a Secure Enclave located in the operational environment of the TOE. The related TOE is composed of the ECN Security Manager, as in the Base-PP, extended with support of the interaction with the Secure Enclave. The TOE is illustrated in red in Figure 37 where the additional components for the TOE compared to the base-PP are represented with a '+' sign on the corner.

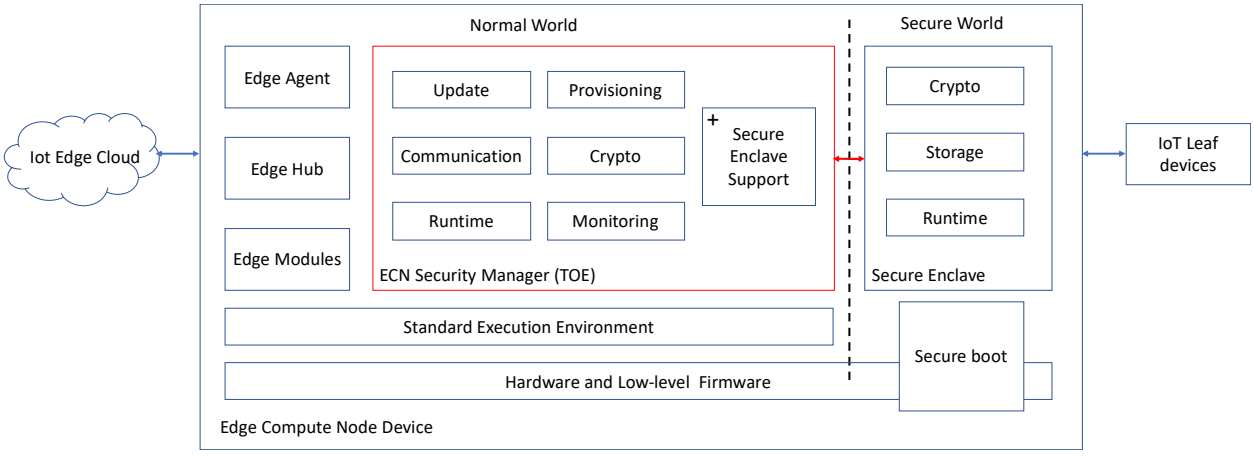


Figure 37 Edge Compute Node with Secure Enclave TOE

D.1.2.1 Usage and Major Security Features of a TOE

The additional security feature for the TOE of this PP-module compared to the Base-PP includes the following:

- Secure communication with trusted IT product (Secure Enclave).

D.1.2.2 TOE Type

The TOE type is a software featuring the security manager for Edge Compute Node extended with secure communication with a trusted IT product.

D.1.2.3 Available non-TOE hardware/software/firmware

Compared to the base-PP, the non-TOE hardware/software/firmware is extended with a Secure Enclave isolated from the Standard Execution Environment with hardware support, such as ARM TrustZone® or Intel® SGX (Software Guard Extension).

This Secure Enclave is used as a root of trust for the TOE. It is responsible for:

- Contributing to the secure boot of the platform and the TOE, by measuring executable code prior to execution and comparing this measure to a reference value;
- Managing sensitive assets for the TOE, in particular cryptographic keys and certificates;
- Offering cryptographic operation services to the TOE, based on the keys managed by the Secure Enclave.

D.2 Conformance Claims

D.2.1 CC Conformance Claims

This PP-Module is CC Part 2 [CC2] extended and CC Part 3 [CC3] extended.

D.2.2 Conformance Claims of the PP

This PP does not claim conformance to any other PP.

D.2.3 Conformance Claims to a Package

This PP-Module inherits the package claims of its base-PP, as stated in Section 2.3.

D.2.4 Conformance Rationale

This PP-module does not provide a conformance rationale because it does not claim conformance to any other PP.

D.2.5 Conformance Statement

This PP-Module inherits from its base-PP the strict conformance as defined in [CC1] for all Security Targets and Protection Profiles claiming conformance to it.

D.2.6 Consistency Rationale

The consistency rationale is given in Section D.6.4.

D.3 Security Problem Definition

This PP-module extends the base-PP SPD with a new threat and a new assumption that supersedes A.SECURE_BOOT and A.STORAGE from the base-PP. All SPD elements from the base-PP apply to this PP-module.

D.3.1 Assets

Table 38 presents the additional asset that need to be protected by the TOE, compared to the Base-PP.

Asset	Description
Secure Enclave data	Data exchanged between the TOE and the Secure Enclave. *Properties: integrity, confidentiality*

Table 38 Assets

D.3.2 Threats

Table 43 presents the additional known or presumed threats to protected resources that are addressed by the TOE.

Threat	Description
T.ENCLAVE_COMM	A local or remote attacker may attempt to illegally access or modify data exchanged between the TOE and Secure Enclave. *Threatened assets: Secure Enclave data (confidentiality and integrity.*

Table 39: Threats

D.3.3 Organizational Security Policies

There are no organizational security policies for this PP-module.

D.3.4 Assumptions

Table 40 presents the additional condition that is assumed to exist in an environment where the TOE is employed. This assumption supersedes A.SECURE_BOOT and A.STORAGE from the base-PP.

Assumption	Description
A.ENCLAVE	It is assumed that the OS provides data-at-rest protection feature for cryptographic keys and certificates used by the TOE in combination with a Secure Enclave.

	It is assumed that the Secure Enclave is used by the TOE for the operations described in Section D.1.2.3 (secure boot, cryptographic operation services). It is assumed that the Secure Enclave is FIPS 140-2 or FIPS 140-3 certified. It is also assumed that the Secure Enclave is certified according to [TEE PP] or [TEE PP] with the Trusted I/O PP-Module [TEE PP I/O]. [TEE PP] GlobalPlatform Device Committee, *TEE Protection Profile*, Version 1.2.1, November 2016. [TEE PP I/O] TEE Trusted I/O PP-Module, Version 1.0, June 2020

Table 40: Assumptions

D.4 Security Objectives

This PP-module introduces a new security objective for the TOE and a new security objective for the Operational Environment. All security objectives from the base-PP apply to this PP-module except for the security objectives for the Operational Environment OE.SECURE_BOOT and OE.STORAGE from the base-PP which are superseded by OE.ENCLAVE.

D.4.1 TOE Security Objectives

Table 41 describes the additional TOE security objective for the TOE of this PP-module.

Environment Objective	Description
O.ENCLAVE_COMM	The TOE will provide the capability to communicate with the Secure Enclave using a trusted channel as a means to maintain the confidentiality and integrity of data that are transmitted between the TOE and Secure Enclave.

Table 41 TOE Security Objectives of the Secure Enclave PP-Module

D.4.2 Security Objectives for the Operational Environment

Table 42 the additional security objectives for the operational environment of this PP-module.

Environment Objective	Description
OE.ENCLAVE	The OS provides data-at-rest protection feature for cryptographic keys and certificates used by the TOE in combination with a Secure Enclave. The Secure Enclave is used by the TOE for the operations described in Section D.1.2.3 (secure boot, cryptographic operation services). The Secure Enclave is FIPS 140-2 or FIPS 140-3 certified.

	The Secure Enclave is also certified according to [TEE PP] or [TEE PP] with the Trusted I/O PP-Module [TEE PP I/O]. [TEE PP] GlobalPlatform Device Committee, *TEE Protection Profile*, Version 1.2.1, November 2016. [TEE PP I/O] TEE Trusted I/O PP-Module, Version 1.0, June 2020

Table 42 Security Objectives for the Operational Environment of the Secure Enclave PP-Module

D.4.3 Security Objectives Rationale

This Section gives an evidence for sufficiency and necessity of the defined objectives. The following table provides an overview for security objectives coverage (TOE and its environment).

	O.COMMS	O.AUTH	O.CONFIG	O.INTEGRITY	O.ENCLAVE_COMM	OE.ADMIN	OE.KEYS	OE.PLATFORM	OE.ENCLAVE	OE.NO_GENERAL_PUR	OE.PHYSICAL
T.EAVESDROP	X		X		X	X	X		X		
T.NETWORK	X	X	X		X	X	X		X		
T.FLAWMOD			X	X		X		X		X	X
T.PERSISTENT			X	X	X	X		X	X	X	X
T.ENCLAVE_COMM					X						X
A.ADMIN						X					
A.KEYS							X				
A.PLATFORM								X			
A.ENCLAVE									X		
A.NO_GENERAL_PURPOSE										X	
A.PHYSICAL											X

D.4.3.1 Security Objective Rationales: Threats

T.EAVESDROP: The combination of the following security objectives diminishes the eavesdropping of communication channels threat:

- O.COMMS ensures confidentiality of exchanged data through a secure communication channel such as TLS.
- O.ENCLAVE_COMM ensures a trusted channel with the Secure Enclave that protects cryptographic keys used for secure communication channel.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG.
- OE.KEYS and OE.ENCLAVE protect the keys and certificates, used to communicate with the TOE, outside of the TOE (i.e. trusted endpoints and underlying platform, respectively).

T.NETWORK: The combination of the following security objectives diminishes the alteration of communication threat:

- O.COMMS ensures integrity of exchanged data through a secure communication channel such as TLS.
- O.ENCLAVE_COMM ensures a trusted channel with the Secure Enclave that protects cryptographic keys used for secure communication channel.
- O.AUTH ensures authentication of communication with trusted end-points.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG.
- OE.KEYS and OE.ENCLAVE protect the keys and certificates, used to communicate with the TOE, outside of the TOE (i.e. trusted endpoints and underlying platform, respectively).

T.FLAWMOD: The combination of the following security objectives diminishes the malicious or exploitable edge module threat:

- O.INTEGRITY ensures integrity of critical functionality, software and updates and controls access to system services.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG and ensures that the TOE is correctly configured and the underlying platform up-to-date.
- OE.PLATFORM provides OS support for domain separation and non-bypassability at the OS level mainly to protect TOE processes from other processes running in the OS with memory isolation and also includes anti-exploitation techniques to mitigate potential breaches.
- OE.NO_GENERAL_PURPOSE ensures that OS has not computing capabilities that could be used by an attacker reducing the exploitability of attacks.
- OE.PHYSICAL provides physical protection for the TOE against attackers with physical access to the TOE.

T.PERSISTENT: The combination of the following security objectives diminishes the persistent access to the TOE threat:

- O.INTEGRITY ensures integrity of critical functionality, software/firmware and data.

- O.ENCLAVE_COMM ensures a trusted channel with the Secure Enclave that protects cryptographic keys used code authentication.
- OE.ADMIN ensures that the TOE is configured properly following the security guidance using the features provided by O.CONFIG and ensures that the TOE is correctly configured and the underlying platform up-to-date.
- OE.PLATFORM provides OS support for domain separation and non-bypassability at the OS level mainly to protect TOE processes from other processes running in the OS with memory isolation and also includes anti-exploitation techniques to mitigate potential breaches.
- OE.ENCLAVE provides support for authentication of the underlying platform code and the TOE.
- OE.NO_GENERAL_PURPOSE ensures that OS has not computing capabilities that could be used by an attacker reducing the exploitability of attacks.
- OE.PHYSICAL provides physical protection for the TOE against attackers with physical access to the TOE.

T.ENCLAVE_COMM The combination of the following security objectives diminishes this threat:

- O.ENCLAVE_COMM that provides a trusted communication channel with the Secure Enclave.
- OE.PHYSICAL provides physical protection for the TOE against attackers with physical access to the TOE.

D.4.3.2 Security Objective Rationales: Assumptions

A.ADMIN: The security objective for the environment OE.ADMIN directly upholds this assumption.

A.KEYS: The security objective for the environment OE.KEYS directly upholds this assumption.

A.PLATFORM: The security objective for the environment OE.PLATFORM directly upholds this assumption.

A.ENCLAVE: The security objective for the environment OE.ENCLAVE directly upholds this assumption.

A.NO_GENERAL_PURPOSE: The security objective for the environment OE.NO_GENERAL_PURPOSE directly upholds this assumption.

A.PHYSICAL: The security objective for the environment OE.PHYSICAL directly upholds this assumption.

D.5 Security Requirements

The section defines the Security Functional Requirements (SFRs) for the TOE.

D.5.1 TOE Security Functional Requirements

This PP-module introduces or refines from the base-PP the following SFRs. All other SFRs from the base-PP apply to this PP-module.

Requirement Class	Requirement Component	Relation to base-PP

Security Audit (FAU)	Audit Data Generation (FAU_GEN.1)	Refinement
Protection of the TSF (FPT)	Extended: Self-Test Failure (FPT_FLS_EXT.1)	Refinement
	Inter-TSF basic TSF data consistency (FPT_TDC.1)	New
	Testing of external entities (FPT_TEE.1)	New
Trusted Path/Channels (FTP)	Inter-TSF Trusted Channel (FTP_ITC.1(Enclave))	New

Table 43 TOE Security Functional Requirements

D.5.1.1 Security Audit (FAU)

D.5.1.1.1 Audit Data Generation (FAU_GEN.1)

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and
[
c) Administrator management functions, as defined in the fourth column of Table 17;
d) Start-up and shutdown of the OS;
e) Specifically defined auditable events in Table 9 and Table 44;
f) **[assignment: *other specifically defined auditable events*]**].

FAU_GEN.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, [**additional information in Table 9 and Table 44**].

Requirement	Auditable Events	Additional Record Contents
FPT_FLS_EXT.1	Measurement of TSF software.	Integrity verification value.
FPT_TDC.1	Failure of data consistency checks.	
FPT_TEE.1	Initiation of external entity test. Failure of external entity test.	None
FTP_ITC.1(Enclave)	None	None

Table 44 Auditable Events

D.5.1.2 Protection of the TSF (FPT)

D.5.1.2.1 Extended: Self-Test Failure (FPT_FLS_EXT.1)

FPT_FLS_EXT.1.1 The TSF shall transition to non-operational mode, log failures in the audit record and [selection: *notify the administrator, [assignment: other actions], no other actions*] when the following types of failures occur:

- failures of the self-test(s)
- TSF software integrity verification failures
- Secure Enclave integrity verification failures
- [selection: *no other failures, [assignment: other failures]*].

D.5.1.2.2 Inter-TSF basic TSF data consistency (FPT_TDC.1)

FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret [**data exchanged with the Secure Enclave**] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use [**specification of Secure Enclave commands / responses**] when interpreting the TSF data from another trusted IT product.

D.5.1.2.3 Testing of external entities (FPT_TEE.1)

FPT_TEE.1.1	The TSF shall run a suite of tests [selection: *during initial start-up, periodically during normal operation, at the request of an authorised user, [assignment: other conditions]*] to check the fulfillment of [**integrity of the Secure Enclave**].
FPT_TEE.1.2	If the test fails, the TSF shall [**perform actions in FPT_FLS_EXT.1, [assignment: *action(s)*]**].

Application Note: In order to check integrity of the Secure Enclave, the TSF can for instance check ID of the Secure Enclave, use Secure Enclave attestation service, read integrity registers, check tamper-detection registers, perform known answer tests for cryptographic operations.

D.5.1.3 Trusted Path / Channels (FTP)

D.5.1.3.1 Inter-TSF Trusted Channel (FTP_ITC.1(Enclave))

FTP_ITC.1.1(Enclave)	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2(Enclave)	The TSF shall permit [**the TSF**] to initiate communication via the trusted channel.
FTP_ITC.1.3(Enclave)	The TSF shall initiate communication via the trusted channel for [**all cryptographic and secure storage functions provided by the Secure Enclave**].

D.6 Rationale for Security Requirements

This section provides a rationale for the security functional requirements and security assurance requirements.

D.6.1 Security Functional Requirements

The mapping presented in Table 36 traces the SFR from this PP-module back to the O.INTEGRITY and O.ENCLAVE_COMM security objectives and demonstrates how these security objectives are met by the SFRs.

Table 45 Rationale for SFRs

SFR	O.INTEGRITY	O.ENCLAVE_COMM
FAU_GEN.1	X	
FPT_FLS_EXT.1	X	
FPT_TDC.1		X
FPT_TEE.1	X	
FTP_ITC.1(Enclave)		X

The rationale for mapping for **O.INTEGRITY** as presented in Section 7.1 is modified as follows:

- FAU_GEN.1 and FPT_FLS_EXT.1 now refer to the SFRs from the PP-module, which refine the ones from the base-PP.
- TOE integrity is now also addressed by requirement FPT_TEE.1 that check integrity of the Secure Enclave, as an incorrect data from the Secure Enclave may corrupt the TOE.

O.ENCLAVE_COMM is addressed by requirements FTP_ITC.1(Enclave) which provides a trusted channel between the TOE and the Secure Enclave and FPT_TDC.1 which verifies consistency of data exchanged between the HSM and the Secure Enclave.

The rationale for mapping for **O.COMMS**, **O.AUTH** and **O. CONFIG** as presented in Section 7.1 is unchanged.

D.6.2 Security Requirements Dependency Analysis

SFR	Dependencies	Resolved
FAU_GEN.1	FPT_STM.1	Yes: FPT_STM.1 from base-PP
FPT_FLS_EXT.1	FAU_GEN.1	Yes: FAU_GEN.1
FPT_TDC.1	No dependencies	
FPT_TEE.1	No dependencies	
FTP_ITC.1(Enclave)	No dependencies	

D.6.3 Security Assurance Requirements

The rationale for security assurance requirements (SARs) is identical to the one of the base-PP, section 7.3.

D.6.4 Consistency Rationale

This PP-Module extends the Base-PP with additional support of cryptographic operations and secure storage managed by a Secure Enclave as an external entity.

This PP-Module has the same TOE type as the Base-PP.

This PP-Module refines SFR FAU_GEN.1 by specifying additional events to be audited. It adds new SFRs which are dedicated to verification of integrity of Secure Enclave and data exchanged with Secure Enclave.

Appendix E Supported PP-Configurations

This appendix describes the supported PP-Configurations defined in this document:

- Edge Compute Node with Secure Boot and File System Secure Storage
- Edge Compute Node with Support for HSM-Based Secure Storage and Cryptography
- Edge Compute Node with Support for Secure Enclave

E.1 Edge Compute Node with Secure Boot and File System Secure Storage

PP-Configuration Title: **Edge Compute Node with Secure Boot and File System Secure Storage PP-Configuration**

Related Base-PP Title: Edge Compute Node Protection Profile

Related PP-Module: Secure Boot and File System Secure Storage PP-Module

PP-Configuration Version: version **1.0**.7, September 4th, 2020

PP-Configuration Conformance Statement: As in the Base-PP, strict conformance as defined in [CC1] for all Security Targets and Protection Profiles claiming conformance to it.

PP-Configuration SAR Statement: As in the Base-PP, EAL1 augmented by ASE_SPD.1, ASE_OBJ.2 and ASE_REQ.2 and augmented CC Part 3 ALC_TSU_EXT.1.

E.2 Edge Compute Node with Support for HSM-Based Secure Storage and Cryptography

PP-Configuration Title: **Edge Compute Node with Support for HSM-Based Secure Storage and Cryptography PP-Configuration**

Related Base-PP Title: Edge Compute Node Protection Profile

Related PP-Module: Support for HSM-Based Secure Storage and Cryptography PP-Module

PP-Configuration Version: version **1.0**.7, September 4th, 2020

PP-Configuration Conformance Statement: As in the Base-PP, strict conformance as defined in [CC1] for all Security Targets and Protection Profiles claiming conformance to it.

PP-Configuration SAR Statement: As in the Base-PP, EAL1 augmented by ASE_SPD.1, ASE_OBJ.2 and ASE_REQ.2 and augmented CC Part 3 ALC_TSU_EXT.1.

E.3 Edge Compute Node with Support for Secure Enclave Based Secure Storage and Cryptography

PP-Configuration Title: **Edge Compute Node with Support for Secure Enclave Based Secure Storage and Cryptography PP-Configuration**

Related Base-PP Title: Edge Compute Node Protection Profile

Related PP-Module: Support for Secure Enclave Secure Storage and Cryptography PP-Module

PP-Configuration Version: version **1.0**.7, September 4th, 2020

PP-Configuration Conformance Statement: As in the Base-PP, strict conformance as defined in [CC1] for all Security Targets and Protection Profiles claiming conformance to it.

PP-Configuration SAR Statement: As in the Base-PP, EAL1 augmented by ASE_SPD.1, ASE_OBJ.2 and ASE_REQ.2 and augmented CC Part 3 ALC_TSU_EXT.1.

Appendix F Initialization Vector Requirements for NIST- Approved Cipher Modes

Cipher Mode	Reference	IV Requirement
Electronic Codebook (ECB)	SP 800-38A	No IV
Counter (CTR)	SP 800-38A	"Initial Counter" shall be non-repeating. No counter value shall be repeated across multiple messages with the same secret key.
Cipher Block Chaining (CBC)	SP 800-38A	IVs shall be unpredictable. Repeating IVs leak information about whether the first one or more blocks are shared between two messages, so IVs should be non-repeating in such situations.
Output Feedback (OFB)	SP 800-38A	IVs shall be non-repeating and shall not be generated by invoking the cipher on another IV.
Cipher Feedback (CFB)	SP 800-38A	IVs should be non-repeating as repeating IVs leak information about the first plaintext block and about common shared prefixes in messages.
XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing (XTS)	SP 800-38E	No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non- negative integer.
Cipher-based Message Authentication Code (CMAC)	SP 800-38B	No IV
Key Wrap and Key Wrap with Padding	SP 800-38F	No IV
Counter with CBC-Message Authentication Code (CCM)	SP 800-38C	No IV. Nonces shall be non-repeating.
Galois Counter Mode (GCM)	SP 800-38D	V shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key unless an implementation only uses 96-bit IVs (default length).

Table 46: References and IV Requirements for NIST-approved Cipher Modes