THALES

Thales Luna K7 Cryptographic Module

SECURITY TARGET

Used as a Standalone Device OR as an Embedded Device in Thales Luna Network HSM

COMMON CRITERIA / ISO 15408, EAL4+

Document Information

Document Part Number	002-010985-001
Release Date	25th September 2020

Revision History

Revision	Date	Reason
J	25th Sept 2020	Final Release

Trademarks, Copyrights, and Third-Party Software

Disclaimer

All information herein is either public information or is the property of and owned solely by Thales. and/or its subsidiaries who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales's information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

> The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.

> This document shall not be posted on any network computer or broadcast in any media other than by NSCIB website and on the CC portal and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein.

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances,

shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

CONTENTS

TERMINOLOGY AND ACRONYMS

Terminology

For the purposes of this document, the acronyms, terms and definitions given in [CEN EN 419221-1] apply.

Common Criteria terms and definitions are given in [CC1].

Additional terms defined for the purposes of this document are listed below:

Term	Definition
Assigned Key	A key (usually a secret key) with the 'Assigned Flag' attribute set to 'assigned', meaning that: > the 'Re-authorization conditions' and 'Key Usage' attributes cannot be changed > the Authorization Data attribute can only be changed by presentation of the current Authorization Data – it cannot be changed or reset by an Administrator > the key cannot be imported or exported. These properties of an Assigned Key support the sole control of a key that is required for secret keys used to create digital signatures.
Authorization Data	Data, including data particular to the user, which is used to control access to (and thus use of) a key. Data particular to the user may include data derived from a secret known only by the user, data derived from a device held by the user and/or data derived from biometric features of the user. Other parts of the authorization data may include data held within the cryptographic module, data held by administrator(s) or data provided by the application.
Electronic Seal	Data in electronic form which is attached to or logically associated with other data in electronic form to ensure the latter's origin and integrity.
Electronic Timestamp	Data in electronic form which binds other data in electronic form to a particular time establishing evidence that the latter data existed at that time.
Secret Key	Either a secret key used in symmetric cryptographic functions, or a private key used in asymmetric cryptographic functions.
Trust Service	Electronic service which enhances trust and confidence in electronic transactions Such trust services are typically but not necessarily using cryptographic techniques or involving confidential material.

Acronyms and abbreviations

API	Application Programming Interface
CC	Common Criteria
DTBS	Data To Be Signed
DTBS/R	Data to be signed or its unique representation
EAL	Evaluation Assurance Level
FM	Functional Module
HSM	Hardware Security Module
IT	Information Technology
PCB	Printed Circuit Board
PCI-E	Peripheral Component Interconnect Express
PP	Protection Profile
RNG	Random Number Generator
SAR	Security Assurance Requirements
SFP	Security Function Policy
SFR	Security Functional Requirements
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	Trust Service Provider

1 Security Target Introduction

1.1 Security Target Reference

Title: Thales Luna K7 Cryptographic Module - Security Target

Version: Rev J

Author: Thales

Reference: 002-010985-001

1.2 TOE Reference

TOE name: Thales Luna K7 Cryptographic Module

Firmware version: 7.7.0

Bootloader version: 1.1.1, 1.1.2 or 1.1.4

Hardware versions[1] [2]: 808-000048-002

808-000073-001

808-000066-001

808-000069-001

808-000070-001

Guidance documentation

> 007-013968-001, Thales Luna K7(+) Cryptographic Module, Common Criteria User Guidance – Part1: Preparative Procedures, Revision E, 25th September 2020.

> 007-000465-001, Thales Luna K7(+) Cryptographic Module, Common Criteria User Guidance – Part2: Operational Guidance (General), Revision F, 25th September 2020.

> 007-000466-001, Thales Luna K7(+) Cryptographic Module, Common Criteria User Guidance – Part3: eIDAS Guidance, Revision E, 25th September 2020.

[1] The hardware version listed identifies the Thales Luna K7 Cryptographic Module hardware alongside the non-reconfigurable firmware loaded during production. Non-reconfigurable firmware includes: (i) one time programmable micro-code loaded onto the Andretta processor during manufacture. It is not possible for the user to identify the version of the micro-code loaded on Andretta 2.0 post manufacture independent of the hardware part number. Any modification to these elements made by Thales will result in a new part number.

[2] A single HW part number (i.e. 808-XXXXXX-XXX) identifies the TOE with 4 variants being covered by this Security Target where each unique variant replaces 'XXXXXX-XXX' with its unique variant identifier e.g. '000048-002'.

> 007-000467-001, Thales Luna K7(+) Cryptographic Module, Common Criteria User Guidance – Part4: TOE Integration for use in Composite Evaluation, Revision D, 25th September 2020.

1.3 TOE Overview

The Thales Luna K7 Cryptographic Module (i.e. the TOE) is a Hardware Security Module (HSM) in the form of a PCI-E card (Thales Luna PCIe HSM). It is operated in a controlled environment and can be used either as a standalone device to be inserted in a server, or as a device embedded in a Thales Luna Network HSM.

The TOE can fulfill general purpose HSM use cases, where assured cryptographic services alongside generation and management of cryptographic keys is required.

The TOE is also suitable for use in support of electronic signature and electronic sealing operations, certificate issuance and revocation, time stamp operations, and authentication services, as identified by the (EU) No 910/2014 regulation of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market (eIDAS). For that purpose, the present Security Target has been explicitly written to comply with the [CEN EN 419221-5] Protection Profile; the TOE supports Assigned Keys, External Key Storage and Key Import/Export operations as defined in the PP.

The TOE also supports the option for further customization of the HSM for a given integration through the ability to load 3rd party developed code in the form of Functional Modules (FM) onto the HSM[3].

All TOE Security Functions are met by the Thales Luna K7 Cryptographic Module when installed in a suitable environment and configured as per the supplied CC User Guidance document [CC User Guidance].

The following non-TOE components are needed for the TOE to be operational in its environment:

> If the TOE is configured to support PED authentication[4]:
  - PED model number PED-04-0103 with minimal PED Firmware version 2.7.4 or higher, or
  - PED model number PED-06-0001 with minimal PED firmware version 2.9.0 or higher

> LUNA client version 10.3.0 or higher (to be installed on remote or local servers)

> If the TOE is embedded in the network appliance: Thales Luna Network HSM with appliance software version 7.7.0 or higher.

[3] Although kernel level application sandboxes are implemented to minimise the risk from loading 3rd party code onto the HSM, the isolation property is not assured as part of this certification.

[4] In addition to the two currently listed PED model numbers (PED-04-0103 and PED-06-0001), future PED hardware could be introduced and identified as compliant with Luna firmware 7.7.0 by Thales.

1.4 TOE Description

1.4.1 TOE Boundaries

The TOE physical boundary is the PCI-E card of which an example embodiment is shown in Figure 1-1.

This Security Target considers the following hardware variants of the TOE:

Table 1-1 – TOE hardware variants within evaluation scope

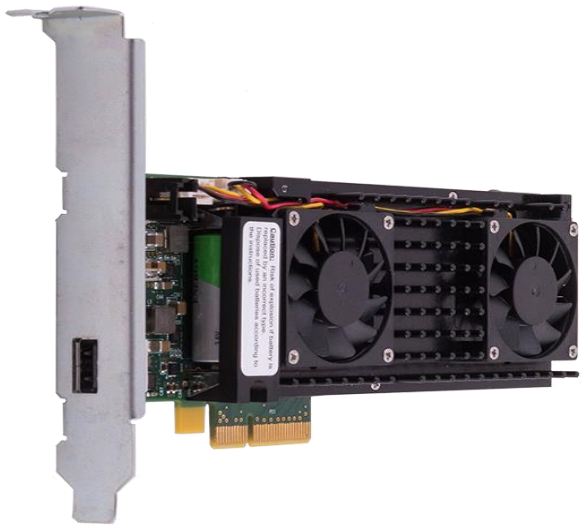
Hardware part number	Description
808-000048-002	Thales Luna K7 Cryptographic Module with Fans
808-000073-001	Thales Luna K7 Cryptographic Module without Fans
808-000066-001	Thales Luna K7 Cryptographic Module without Fans (legacy part number)
808-000069-001	Thales Luna K7+ Cryptographic Module with Fans
808-000070-001	Thales Luna K7+ Cryptographic Module without Fans

The K7 and K7+ denomination is related to the type of shield providing physical protection of the TOE Printed Circuit Board (PCB):

Table 1-2 – TOE shield naming

Shield naming	Description
K7	Passive shield (epoxy coating)
K7+	Active shield (electrically wired)

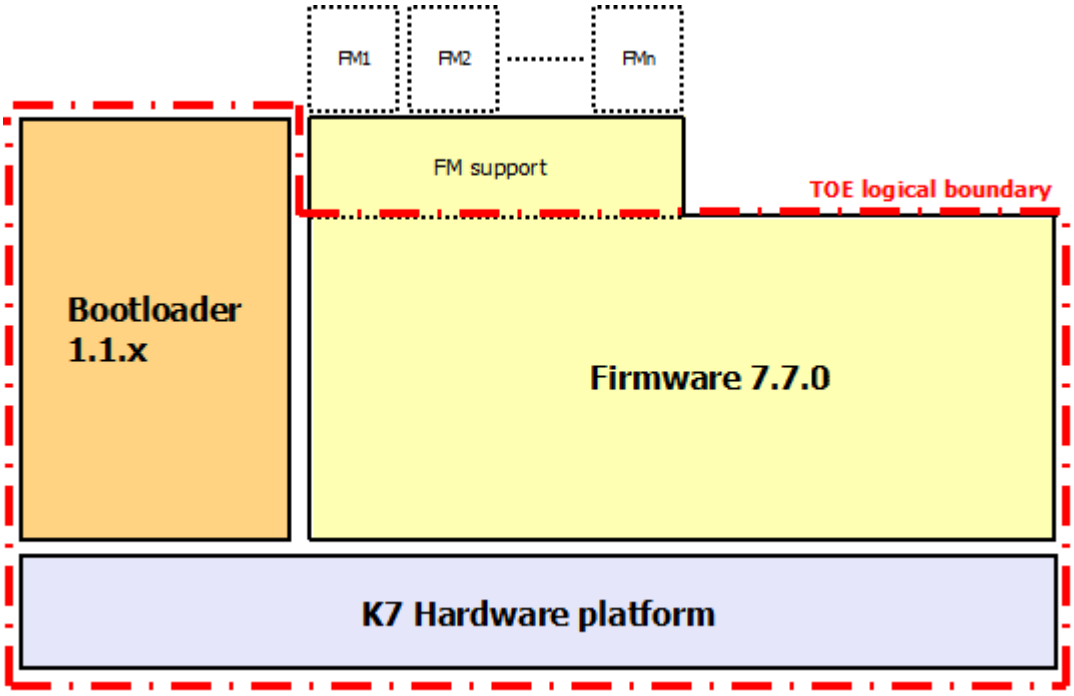
Figure 1-1 – Thales Luna K7 Cryptographic Module, fan variant without tamper wrap (808-000048-002)



Note: as mentioned in section 1.3, the TOE can be used either as a standalone device to be inserted in a generic server, or as a device embedded in a Thales Luna Network HSM. In both cases, the generic server and the Thales Luna Network HSM represent the 'Hardware Appliance Boundary' illustrated in Figure 1-3.

The TOE logical boundary is the 'main' part of the firmware of the Figure 1-2:

Figure 1-2 – Thales Luna K7 Cryptographic Module, TOE logical boundary



As mentioned in section 1.3 "TOE overview", the TOE supports the ability to load 3rd party developed code in the form of Functional Modules (FM) onto the HSM. FMs are running in a dedicated 'FM process' which

is distinct from the 'main' firmware process. An "FM support" component is implemented within the firmware to support the FM capability[5].

For the present certification, no FMs are loaded within the HSM and the "FM support" piece of firmware is out of the TOE[6].

Note that the guidance documentation [CC User Guidance] is also part of the TOE, as referenced in section 1.2.

1.4.2 TOE Delivery

The TOE components are delivered to the end-user as follows:

> **Hardware:** shipped by tracked courier directly to the end-user. Details of shipment tracking references, HSM serial number alongside serial numbers of all tamper labels used in the unit packaging are emailed to the customer on the confirmation of shipment.

- Hardware can be identified as one of the valid HW part (808-000048-002, 808-000073-001, 808-000066-001, 808-000069-001, 808-000070-001) being shown on the product label.

> **Firmware:** is either pre-loaded onto the Hardware[7] during manufacturing or alternatively may be downloaded from the Thales Customer Support Portal[8] as:

- '621-000192-010_fwupdate_7.7.0_PCI_HSM_RevA.fuf' (if TOE is used as a standalone device)
- '630-010740-007_SPKG_APPL_Net_HSM_7.7.0_FW-7.7.0_BU_G7-7.7.1_BU_G5-6.28.0_RevA.tar' (if TOE is embedded in a Thales Luna Network HSM)

> **Common Criteria User Guidance Documentation** [CC User Guidance] can be downloaded from the Thales Customer Support Portal as:

- '007-013968-001_K7_CC_User_Guidance_Part1_Rev_E.pdf'
- '007-000465-001_K7_CC_User_Guidance_Part2_Rev_F.pdf'
- '007-000466-001_K7_CC_User_Guidance_Part3_Rev_E.pdf'
- '007-000467-001_K7_CC_User_Guidance_Part4_Rev_D.pdf'

Details on accessing the Thales Customer Support Portal are provided with the shipment confirmation supplied on shipment of the TOE hardware.

1.4.3 TOE Architecture

The TOE is a set of configured software and hardware that fits the generic TOE architecture shown in Figure 1-3.

[5] The "FM support" component enables the FMs to run in a dedicated process. It also contains a library translating PKCS#11 commands into the set of commands natively supported by the HSM.
[6] Note that even if it is considered in the TOE environment, the "FM support" piece of firmware is provided to the Evaluator as an input to the TOE vulnerability analysis (to verify that it cannot be used by an attacker to threaten the TOE assets)
[7] Identified as 'FW 7.7.0' in the output either from the 'hsm show' LunaSH command or 'hsm showinfo' for the LunaCM interface.
[8] See [CC User Guidance] for further information on how to check and install firmware.

The TOE provides cryptographic functions that support trust services, and manages (and protects) the cryptographic keys used by these functions. Note that the TOE is not aware of the context in which a cryptographic function is used. Any such context is therefore the responsibility of client applications used by the trust service provider or operator, and these client applications need to use the cryptographic functions in an appropriate way. In general this will be achieved by suitable configuration of the TOE and its stored data[9].

Local client applications reside in the same hardware appliance as the TOE, e.g. in the case of the TOE being a PCI-E card inside a server, local client applications are the applications running within the same server boundary and using the TOE's services through the PCI-E bus. Another example of local client application is an embedded application running inside the physical boundary of the TOE and using the Luna FM API. Note that the secure environment is considered sufficient to provide the authentication, confidentiality and integrity protection needed for communication between the TOE and local applications.

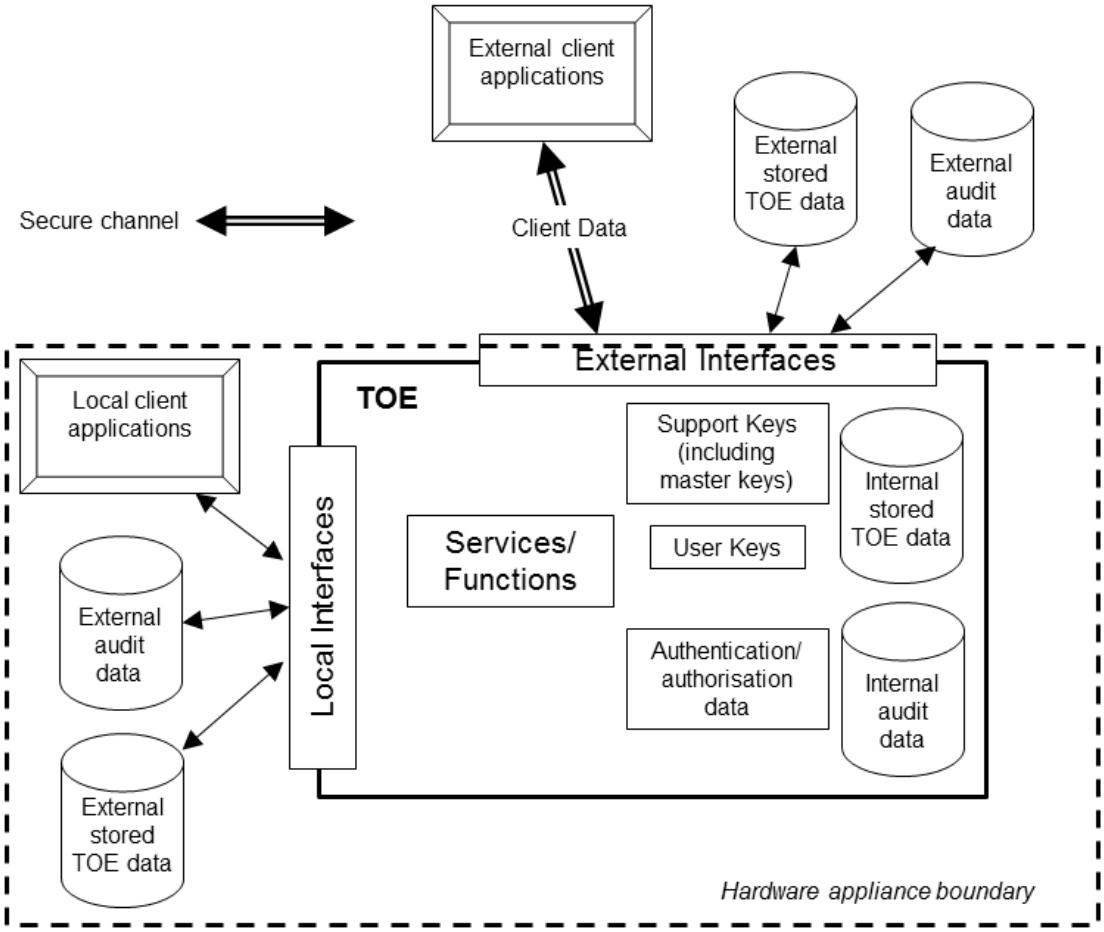
External client applications communicate remotely with the TOE through a network connection and over a secure channel identified as Secure Trusted Channel (STC) which provides authentication of its end-points and protection of confidentiality and integrity of data sent over the channel.

Secure channels may also exist between external and local client applications, but these are not covered within the scope of this Security Target.

In all cases, the Client Application is outside the scope of the TOE.

[9] For example: to ensure that secret keys intended for electronic signature creation are only available for use by the signatory to whom they are linked, the client application must follow an appropriate process to generate the key pair, to maintain sole control (if required) of the secret key by the intended signatory, and to ensure that the key can only be used for signing.

Figure 1-3: TOE Architecture



1.4.4 Admin and User partitions

Cryptographic keys are stored and managed inside containers called partitions. The TOE supports two types of partitions:

> The Admin partition is mandatory. Its primary purpose is to support administrative tasks at the HSM level. Specific roles are defined in the Admin partition and there can be only one Admin partition inside the TOE.

> User partitions are optional. The primary purpose of a User partition (also called Application partition) is to group all keys belonging to a same entity or application in a dedicated container isolated from the other partitions. Specific roles are supported at the user partition level, which are different from the roles defined in the Admin partition.

Any type of partition (Admin partition or User partition) can contain cryptographic keys. For a given partition, the management and usage of the related key material is restricted to the roles assigned to that partition, therefore enforcing a strict isolation between the different partitions managed inside the TOE.

The following section provides further details about the roles supported in partitions.

1.4.5 HSM roles

The following authenticated roles are supported by the HSM:

Table 1-3 –HSM roles

Role	Description
HSM Security Officer (HSM SO)	– Admin partition role – Authorized to install and configure the TOE, set and maintain global HSM security policies, create and delete application partitions. The HSM SO can also create the Administrator role and reset the Administrator password (configuration dependent). The HSM SO can also perform key management tasks and cryptographic operations for the Admin partition. Note that the HSM SO can only use key objects if he/she can also successfully authenticate as the Key Owner. The TOE can have only one HSM SO.
Administrator	– Admin partition role – Optional Admin partition role, performs key management tasks and cryptographic operations for the Admin partition, without having privileges to unblock keys / assign keys / reset key authentication data. Note that the Administrator can only use key objects if he/she can also successfully authenticate as the Key Owner.
Audit User	– Admin partition role – Initializes the Audit container containing the secret key used to generate Message Authentication Code (MAC) for secure audit messages alongside configuring logging levels for the HSM.
Application Partition Security Officer (Partition SO)	– User partition role – Creates partition level roles, activates partition, sets and changes partition-level policies, option to reset Crypto Officer password (configuration dependent).
STC User	Implicit role whose main purpose is to authenticate remote client connections from outside the TOE hardware appliance boundary[10].
Application Partition Crypto Officer (Partition CO)	– User partition role – Partition role authorized to create, use, destroy and transfer key objects for a given partition. Can optionally create the Partition Limited CO and Partition CU, and perform initial assignment of key authorization data. Note that the Partition CO can only use key objects if he/she can also successfully authenticate as the Key Owner.

[10] Usage of the STC is mandatory to authenticate remote client applications. Note that local client applications can be authenticated through the STC as well, although this is not mandatory - as explained in section 7.5 “Trusted Channel”.

Role	Description
Application Partition Limited Crypto Officer (Partition Limited CO)	– User partition role – Optional partition role authorized to create, use and delete key objects, and perform initial assignment of key authorization data without having privileges to create Partition Limited CO or CU / reset login credentials. Note that the Partition Limited CO can only use and delete key objects if he/she can also successfully authenticate as the Key Owner.
Crypto User (Partition CU)	– User partition role – Partition role authorized to use the key objects within the partition (e.g., sign, encrypt/decrypt). Note that the Partition CU can only use key objects if he/she can also successfully authenticate as the Key Owner.
Key Owner	Implicit role used to authenticate the owner of a key through verification of the related key authorization data.

On a User Partition, the Partition CO, Partition Limited CO, Partition CU and Key Owner can communicate with the TOE for cryptographic operations using PKCS #11 which is part of the IT Environment. The HSM SO and Administrator can do it as well on the Admin Partition.

The HSM SO and Partition SO use a separate Command Line Interface (CLI), which is part of the IT Environment, to send command data to perform HSM configuration, to make security policy setting changes and to perform user creation/deletion. The CLI can also be used by the Partition CO to initiate the transfer of cryptographic objects between distinct TOEs or between the TOE and a compatible Trusted IT Product via a Trusted Channel. The network channels used to provide access to the CLI are a requirement of the IT environment.

The TOE allows for the creation of multiple users in the Partition CO, Partition Limited CO and Partition CU roles. Each user is created within a cryptographically separated partition in the Luna PCI-E cryptographic module. Each partition *must* have a user assigned to the Partition CO role (a maximum of one user assigned to the Partition CO role is permitted). A partition *may* optionally assign a distinct user to the Partition Limited CO and/or Partition CU role.

Table 1-4 provides the mapping between the roles supported by the TOE and the subjects defined in the PP [CEN EN 419221-5].

Table 1-4 – Mapping between HSM roles and PP-defined subjects

Function	HSM Role(s)	Related PP subject
Initialisation, configuration	HSM SO, Partition SO	S.Admin
Audit Log Configuration	Audit User	S.Admin
Key Management	Partition CO (for User partitions) HSM SO, Administrator (for Admin partition)	S.Admin

Function		HSM Role(s)	Related PP subject
Client Application Authentication		STC User	S.Application
Key Use	Session authentication	Partition CO, Partition Limited CO, Partition CU (for User partitions, and depending on the configuration of TOE and deployment goals) HSM SO, Administrator (for Admin partition)	S.User
	Key usage	Key Owner	

Note: according to [CEN EN 419221-5], the subject S.Application represents "a client application, or process acting on behalf of a client application and that communicates with the TOE over a local or external interface". In the above table, it can be directly mapped to the HSM implicit role "STC User" when the Secure Trusted Channel (STC) is enforced between the client application and the TOE. As mentioned in section 7.5 "Trusted Channel", usage of the STC is mandatory for remote client applications and optional for local ones. In the latter case, when STC is not used, S.Application is not mapped to any specific role supported by the TOE, and the first authentication step occurs at the session level.

1.4.6 Authentication/Authorization

The TOE implements separate authentication or authorization[11] for all categories of subjects:

- Administrators of the TOE (S.Admin)
- Application users of TOE cryptographic functions (S.Application)
- Users of secret keys[12] (S.User).

External Client applications (S.Application) communicate with the TOE over a secure channel and are authenticated through the verification of a digital certificate.

Two models for authentication are supported for all other roles (except the Key Owner):

- **Token based authentication** - where authentication data is provided by means of a trusted PIN Entry Device (PED). The PED can be connected locally through a separate port to the TOE (local PED) or remotely (remote PED). The PED and software required to operate it are the responsibility of the IT Environment.
- **Password based authentication** – where authentication is based on a configurable length password.

The model of authentication to be used is determined during HSM manufacturing and this model is used for all roles.

[11] In this Security Target 'authentication' implies that the user is specifically identified, whereas 'authorization' implies that the authority of the user to use the key is established but the identity of the individual may not be known (e.g. where a single key is available to a number of individuals using a shared passphrase). As noted elsewhere, it is the responsibility of client applications to ensure that they use the correct mechanism for the context of the relevant keys and cryptographic functions.

[12] Which in at least some cases need to have their use limited to a certain natural person or legal person. More details of these requirements and the definitions of natural and legal persons can be found in [Regulation].

The Key Owner is authenticated based on a shared secret presented to the TOE over a secure channel protecting against access to the shared secret by any other roles authenticated to the HSM.

Authorization as the Key Owner is always separately required before a key can be actually used in a cryptographic function (or exported), regardless of any other authorization that may have been established for administrators (S.Admin) or client applications (S.Application). This requirement reflects the distinct activities that are being authorized in each case:

- Authorization to act as an administrator is an authorization to carry out management activities on the TOE, but not to use keys (in fact the requirement to be able to support sole control of a signature key means that in such cases an administrator must not be able to use the key nor to access its value unless the administrator happens also to demonstrate authorization as the owner of that key).
- Client applications are authorized to connect to the TOE in order to invoke cryptographic functions, but the ownership of keys used in such functions must be separately verified, as the client application may e.g. supply a signature service to a number of different users.

Moreover, a cryptographic function will only be carried out by the TOE if authorization is obtained for use with a key that can be used with that cryptographic function. Thus, a request by a client to use a specific cryptographic function will fail if the attributes of the key supplied do not allow its use for that operation.

1.4.7 Cryptographic functions

The TOE provides the following cryptographic functions:

- Digital signature generation and verification
- Message digest generation
- Message authentication code generation and verification
- Encryption and decryption (symmetric and asymmetric)
- Key generation
- Key derivation
- Generation of shared secret values
- Cryptographic support for one time password and other non-PKI based authentication mechanisms
- Random number generation.

These functions may also be used to support TSP system functions to create electronic seals and electronic timestamps. From the perspective of this security target, specific cryptographic purposes such as electronic signatures and electronic seals are not distinguished: they both consist of a series of cryptographic functions (such as creating message digests, or encrypting data) using specific keys[13].

Note that only algorithms and algorithm parameters (e.g. key length) approved for the identified purpose shall be used by the TOE to carry out cryptographic operations. Supported algorithms and key sizes have been selected to be consistent with a subset of options permitted in [TS 119 312] or [SOG-IS-Crypto].

[13] Some cryptographic operations, such as creating message digests, do not require keys.

Where the HSM is deployed in order to meet a national requirement, the end-user should only use algorithms compliant to local regulations as applicable – e.g. as may apply to a TSP within a specific region as part of future legislation.

1.4.8 Key management

The TOE supports the secure management of cryptographic keys necessary for its implemented cryptographic functions, including:

- Key establishment (including key generation)
- Protection of keys held within the TOE and held externally (for use by the TOE)
- Control of access and use of keys by the cryptographic functions within the TOE
- Deletion of keys within the TOE.

The TOE supports the following techniques for establishing keys:

- Generation of cryptographic keys using a random number generator and implementing the key generation algorithms depending on the intended use of the keys
- Import of cryptographic keys in encrypted form or cryptographic key components using split-knowledge procedures
- Key agreement protocols establishing common secrets with external entities
- Derivation of keys from shared knowledge.

Secret keys are associated with attributes that determine their use, and the correct association between the key and its attributes is protected against unauthorized modification. The attributes maintained by the TOE include[14]:

- The identifier of the key (this enables it to be linked by an application to a particular owner)
- The type of the key (e.g. whether the key is a secret key of a symmetric cryptographic algorithm or the secret (commonly called private) key of an asymmetric cryptographic algorithm)
- Authorization data that enables access to the key (required only for secret keys)
- Re-authorization conditions such as determining a time period or number of uses of a key that are enabled by a single presentation of the correct authorization data for the key, after which the authorization will have to be re-presented in order to authorize any further uses of the key (required only for secret keys)
- Key usage constraints that determine which cryptographic functions can use the key (e.g. encryption or signature)
- Whether the key is allowed to be exported
- Whether the key is an Assigned Key
- Integrity protection data that protects the integrity of the key value, the values of the key attributes, and the binding of the key to its attributes.

[14] These attributes are sufficient to allow a secret key to be identified as one that is used to produce qualified electronic signatures and qualified electronic seals that meet the requirements of [Regulation]

Authorization to change the attributes of a key is, in general, distinct from authorization to use the key for cryptographic functions. For example, a signature key may need to require that some or all of its attributes cannot be changed after initial definition (e.g. because such changes might enable subjects beyond the signatory alone to access the key, or might allow the permitted use of the key to be changed) – this is supported by the definition of an 'Assigned Key' which cannot be imported or exported, for which the re-authorization conditions and key usage cannot be changed, and for which the authorization data can only be changed on successful validation of the current authorization data.

Keys can leave the TOE in the following situations:

> **External storage of keys**

  The TOE allows external storage of keys for later use by the TOE (or another instance of the TOE within the same authorized security infrastructure operated by a TSP). This reflects the fact that when dealing with large numbers of keys, a cryptographic module may not have sufficient internal storage to hold them all internally. Keys stored in this way correspond to 'external stored TOE data' in Figure 1-3, and the form in which the key is stored is sufficient to ensure the confidentiality (for at least secret keys) and integrity of the key and the binding of the key to its attributes.

> **Export of keys**

  The TOE allows export of keys for use by authorized client applications, provided that they are not Assigned Keys, that other key attributes do not prohibit export, and that the correct authorization data for the key has been supplied. Although the TOE checks key attributes to determine whether to allow export, the appropriate values to use for the key attributes will depend on the application context in which the key is used, and the security measures (technical, physical and procedural) that apply to that context. Keys that are exported are not included in the 'external stored TOE data' in Figure 1-3.

  Keys might be imported or exported as part of providing general cryptographic functions (e.g. in support of client applications that use the TOE to support their own authentication mechanisms), but the TOE also allows individual secret keys to be identified as non-exportable. Assigned keys cannot be imported or exported, and represent a more strongly controlled type of key that is intended to be used only within the TOE for operations such as electronic signature or electronic seal generation.

1.4.9 Self-protection

The cryptographic module ensures it always remains in a secure state, even when under attack or facing abnormal conditions. For that purpose, the module implements:

> Self-tests to demonstrate the correct operation of the Random Number Generator (RNG) and of the cryptographic algorithms, as well as the verification of the firmware integrity and authenticity

> Protection against physical attacks through the monitoring of voltage and temperature (and abortion of any operation if voltage or temperature are outside the expected range)

> Protection against physical attacks by means of a passive epoxy-coated shield (for K7 card) or electrically wired active shield (for K7+ card)

> Protection against power loss.

1.4.10 Audit

The cryptographic module is assumed to be part of a larger system that manages audit data for the system as a whole (integrating audit records from a number of individual components). The TOE therefore logs audit

records for its own actions, and typically exports these audit data to a separate audit server which is assumed to be monitored and controlled by the System Auditor.

An auditor role (Audit User) is defined for the TOE as a subset of the 'administrator', whose responsibility is limited to configuring the audit system of the TOE and not for managing the complete log which is assumed to be the role of System Auditor in the larger system.

1.4.11 Usage and major security features of the TOE

The TOE is a separate component with its own hardware and software, communicating via a well-defined physical and logical interface with the client application. The main TOE physical interface is the PCI-E bus with multiple logical interfaces supported on-top of the physical layer.

Several instances of a TOE may be combined in a single domain under a common infrastructure, but the nature of this combination and common infrastructure is beyond the scope of this Security Target.

The threat environment the TOE is designed for is one of high threat of network compromise, and low threat of physical compromise (for example, a Certification Authority facility with a high degree of physical protection, but an operational requirement to be connected to an untrusted network such as the internet).

The environment is assumed to prevent prolonged unauthorized physical access to the TOE (including theft). The TOE provides physical protection mechanisms to deter undetected compromise of its security functions by low attack potential individuals that do have physical access to the TOE (for example disgruntled employees with legitimate access to the TOE).

The TOE is responsible for protecting the keys against logical attacks that would result in disclosure, compromise and unauthorized modification, and for ensuring that the TOE services are only used in an authorized way.

Client applications request cryptographic functions from the TOE, typically using a key managed by the TOE[15], once the appropriate authorization has been provided.

[15] All cryptographic operations in the scope of this Protection Profile are carried out using keys managed by the TOE, and therefore any use of other keys is outside the scope of the Security Target.

2 Conformance Claim

Common Criteria version: This ST conforms to CC Version 3.1 Release 5 [CC1] [CC2] [CC3].

Conformance to CC part 2 and 3:

- This ST is CC part 2 extended with FCS_RNG.1 (Generation of Random Numbers) and FPT_TST_EXT.1 (Basic TSF Self Testing). All the other SFRs have been drawn from the catalogue of requirements in CC part 2 [CC2].
- This ST is CC part 3 conformant. It means that all SARs in that ST are based only upon assurance components in CC part 3 [CC3].

Assurance package conformance: EAL4 augmented (EAL4+)

This ST conforms to the assurance package EAL4 augmented by ALC_FLR.2 and AVA_VAN.5.

Protection Profile (PP) conformance claim:

This ST claims strict conformance to the [CEN EN 419221-5] protection profile.

3 Security Problem Definition

3.1 Assets

The assets that need to be protected by the TOE are identified below:

- **R.SecretKey**: secret keys used in symmetric cryptographic functions and private keys used in asymmetric cryptographic functions, managed and used by the TOE in support of the cryptographic services that it offers. This includes user keys, owned and used by specific users, and support keys used in the implementation and operation of the TOE. The asset also includes copies of such keys made for external storage and/or backup purposes. The confidentiality and integrity of these keys shall be protected.
- **R.PubKey**: public keys managed and used by the TOE in support of the cryptographic services that it offers (including user keys and support keys). This asset includes copies of keys made for external storage and/or backup purposes. The integrity of these keys shall be protected.
- **R.ClientData**: data supplied by a client for use in a cryptographic function. Depending on the context, this data may require confidentiality and/or integrity protection.
- **R.RAD**: reference data held by the TOE that is used to authenticate an administrator (hence to control access to privileged administrator functions such as TOE backup, export of audit data) or to authorize a user for access to secret and private keys (R.SecretKey). This asset includes copies of authentication/authorization data made for external storage and/or backup purposes. The integrity of the RAD shall be protected; its confidentiality shall also be protected unless the authentication method used means that the RAD is public data (such as a public key).

3.2 Subjects

The types of subjects identified in this ST are:

- **S.Application**: a client application, or process acting on behalf of a client application and that communicates with the TOE over a local or external interface. Client applications will in some situations be acting directly on behalf of end users (see S.User).
- **S.User**: an end user of the TOE who can be associated with secret keys and authentication/authorization data held by the TOE. An end user communicates with the TOE by using a client application (S.Application).
- **S.Admin**: an administrator of the TOE. Administrators are responsible for performing the TOE initialization, TOE configuration and other TOE administrative functions.

Each type of subject may include many individual members, for example a single TOE will generally have many users who are all included as members of the type S.User.

3.3 Threats

The following threats are defined for the TOE. The attacker (i.e. the 'threat agent') described in each of the threats is a subject who is not authorized for the relevant action, but who may present themselves as either a completely unknown user, or as one of the subjects in section 3.2 (but in this case the attacker will not have access to the authentication or authorization data for the subject).

T.KeyDisclose Unauthorised disclosure of secret/private key

An attacker obtains unauthorized access to the plaintext form of a secret key (R.SecretKey), enabling either direct reading of the key or other copying into a form that can be used by the attacker as though the key were their own. This access may be gained during generation, storage, import/export, use of the key, or backup if supported by the TOE.

T.KeyDerive Derivation of secret/private key

An attacker derives a secret key (R.SecretKey) from publicly known data, such as the corresponding public key or results of cryptographic functions using the key or any other data that is generally available outside the TOE.

T.KeyMod Unauthorised modification of a key

An attacker makes an unauthorized modification to a secret or public key (R.SecretKey or R.PubKey) while it is stored in, or under the control of, the TOE, including export and backups if supported. This includes replacement of a key as well as making changes to the value of a key, or changing its attributes such as required authorization, usage constraints or identifier (changing the identifier to the identifier used for another key would allow unauthorized substitution of the original key with a key known to the attacker). The threat therefore includes the case where an attacker is able to break the binding between a key and its critical attributes[16].

T.KeyMisuse Misuse of a key

An attacker uses the TOE to make unauthorized use of a secret key (R.SecretKey) that is managed by the TOE (including the unauthorized use of a secret key for a cryptographic function that is not permitted for that key[17]), without necessarily obtaining access to the value of the key.

T.KeyOveruse Overuse of a key

An attacker uses a key (R.SecretKey) that has been authorized for a specific use (e.g. to make a single signature) in other cryptographic functions that have not been authorized.

[16] See OT.KeyIntegrity in section 4.1 for further discussion of critical attributes of a key.
[17] This therefore means that the threat includes unauthorized use of a cryptographic function that makes use of a key.

T.DataDisclose Disclosure of sensitive client application data

An attacker gains access to data that requires protection of confidentiality (R.ClientData, and possibly R.RAD) supplied by a client application during transmission to or from the TOE or during transmission between physically separate parts of the TOE.

T.DataMod Unauthorized modification of client application data

An attacker modifies data (R.ClientData such as DTBS/R, authentication/authorization data, or a public key (R.PubKey) supplied by a client application during transmission to the TOE or during transmission between physically separate parts of the TOE, so that the result returned by the TOE (such as a signature or public key certificate) does not match the data intended by the originator of the request.

T.Malfunction Malfunction of TOE hardware or software

The TOE may develop a fault that causes some other security property to be weakened or to fail. This may affect any of the assets and could result in any of the other threats being realized. Particular causes of faults to be considered are:

- Environmental conditions (including temperature and power)
- Failures of critical TOE hardware components (including the RNG)
- Corruption of TOE software.

3.4 Organisational Security Policies

P.Algorithms Use of approved cryptographic algorithms

The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognized authorities as appropriate for use by TSPs.

Application Note 1

The relevant authorities and endorsements are determined by the context of the client applications that use the TOE. For digital signatures within the European Union this is as indicated in [Regulation] and an exemplary list of algorithms and parameters is given in [TS 119 312] or [SOG-IS-Crypto] (see also section 1.4.7).

P.KeyControl Support for control of keys

The life cycle of the TOE and any secret keys that it manages (where such keys are associated with specific entities, such as the signature creation data associated with a signatory or the seal creation data associated with a seal creator[18]), shall be implemented in such a way that the secret keys can be reliably protected by the legitimate owner against use by others, and in such a way that the use of the secret keys by the TOE can be confined to a set of authorized cryptographic functions.

[18] A seal creator may be a *legal person* (see [Regulation]) rather than a *natural person*, and seal creation data may therefore be authorized for use by a number of natural persons, depending on the nature and requirements of the trust service provided.

Application Note 2

This policy is intended to ensure that the TOE can be used for qualified electronic seals and qualified electronic signatures as in [Regulation], but recognizes that not all keys are used for such purposes. Therefore, although the TOE needs to be able to support the necessary strong controls over keys in order to create such seals and signatures, not all keys need the same level and type of control.

P.RNG Random Number Generation

The TOE is required to generate random numbers that meet a specified quality metric, for use by client applications. These random numbers shall be suitable for use as keys, authentication/authorization data, or seed data for another random number generator that is used for these purposes.

P.Audit Audit trail generation

The TOE is required to generate an audit trail of security-relevant events, recording the event details and the subject associated with the event.

Application Note 3

The cryptographic module TOE is assumed to be part of a larger system that manages audit data. The TOE therefore logs audit records, and it is assumed that these are collected, maintained and reviewed in the larger system. Hence there is no separate auditor role within the cryptographic module TOE, but the role of System Auditor is assumed to exist in the larger system – cf. A.AuditSupport in section 3.5.

3.5 Assumptions

A.ExternalData Protection of data outside TOE control

Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities shall provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment.

In particular, any backups of the TOE and its data are maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data does not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data requires at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

A.Env Protected operating environment

The TOE operates in a protected environment that limits physical access to the TOE to authorized Administrators. The TOE software and hardware environment (including client applications) is installed maintained by Administrators in a secure state that mitigates against the specific risks applicable to the deployment environment.

A.DataContext Appropriate use of TOE functions

Any client application using the cryptographic functions of the TOE will ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and

confidentiality). For example, when creating a digital signature over a DTBS the client application will ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and will correctly and securely manage the signature received from the TOE; and when certifying a public key the client application will ensure that necessary checks are made to prove possession of the corresponding private key. The client application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) performs a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS.

Client applications are also responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events.

Similar requirements apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys.

Appropriate procedures are defined for the initial creation of data and continuing operation of the TOE according to the specific risks applicable to the deployment environment and the ways in which the TOE is used.

A.UAuth Authentication of application users

Any client application using the cryptographic services of the TOE will correctly and securely gather identification and authentication/authorization data from its users and securely transfer it to the TOE (protecting the confidentiality of the authentication/authorization data as required) when required to authorize the use of TOE assets and services.

A.AuditSupport Audit data review

The audit trail generated by the TOE will be collected, maintained and reviewed by a System Auditor according to a defined audit procedure for the TSP.

Application Note 4

As noted for P.Audit in section 3.4, the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.

A.AppSupport Application security support

Procedures to ensure the ongoing security of client applications and their data will be defined and followed in the environment, and reflected in use of the appropriate TOE cryptographic functions and parameters, and appropriate management and administration actions on the TOE. This includes, for example, any relevant policies on algorithms, key generation methods, key lengths, key access, key import/export, key usage limitations, key activation, cryptoperiods and key renewal, and key/certificate revocation.

4 Security Objectives

This section identifies and defines the security objectives for the TOE and its operational environment.

Security objectives reflect the stated intent and counter the identified threats, as well as comply with the identified organisational security policies and assumptions.

4.1 Security Objectives for the TOE

The following security objectives describe security functions to be provided by the TOE.

OT.PlainKeyConf Protection of confidentiality of plaintext secret keys

The plaintext value of secret keys is not made available outside the TOE (except where the key has been exported securely in the manner of OT.ImportExport). This includes protection of the keys during generation, storage (including external storage), and use in cryptographic functions, and means that even authorized users of the keys and administrators of the TOE cannot directly access the plaintext value of a secret key.

OT.Algorithms Use of approved cryptographic algorithms

The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognized authorities as appropriate for use by TSPs. This ensures that the algorithms used do not enable publicly known data to be used to derive secret keys.

Application Note 5

See note under P.Algorithms (section 3.4) on relevant references for digital signatures within the European Union.

OT.KeyIntegrity Protection of integrity of keys

The value and critical attributes of keys (secret or public) have their integrity protected by the TOE against unauthorized modification (unauthorized modifications include making unauthorized copies of a key such that the attributes of the copy can be changed without the same authorization as for the original key). Critical attributes in this context are defined to be those implementation-level attributes of a key that could be used by an attacker to cause the equivalent of a modification to the key value by other means (e.g. including changing the cryptographic functions for which a key can be used, the users with access to the key, or the identifier of the key). This objective includes protection of the keys during generation, storage (including external storage), and use.

OT.Auth Authorization for use of TOE functions and data

The TOE carries out an authentication/authorization check on all subjects before allowing them to use the TOE. The following types of entity are distinguished for the purposes of authorization (i.e. each type has a distinct method of authorization):

> Administrators of the TOE

> Users of TOE cryptographic functions (client applications using secure channels)

> Users of secret keys.

In particular, the TOE always requires authorization before using a secret key.

Application Note 6

Local client applications within a suitable security environment (such as client applications that are connected to the TOE by a channel such as a PCIe bus within the same hardware appliance) do not require authentication to communicate with the TOE, as noted in section 1.4.3. However, use of a secret key always requires prior authorization.

OT.KeyUseConstraint Constraints on use of keys

Any key (secret or public) has an unambiguous definition of the purposes for which it can be used, in terms of the cryptographic functions or operations (e.g. encryption or signature) that it is permitted to be used for. The TOE rejects any attempt to use the key for a purpose that is not permitted. The TOE also has an unambiguous definition of the subjects that are permitted to access the key (and the purposes for which this access can be used) and allows this to be set to the granularity of an individual subject – these access constraints apply to *use* of the key even where the key value is not accessible. This objective means that the TOE also prevents unauthorized use of any cryptographic functions that use a key.

OT.KeyUseScope Defined scope for use of a key after authorization

The TOE is required to define and apply clearly stated limits on when authorization and re-authorization are required in order for a secret key to be used[19]. For example the TOE may allow secret keys to be used for a specified time period or number of uses after initial authorization, or for may allow the key to be used until authorization is explicitly rescinded. As another example, the TOE may implement a policy that requires re-authorization before every use of a secret key.

Application Note 7

Such limits on the use of a key after initial authorisation are termed "re-authorisation conditions" in this ST. A wide range of policies and re-authorisation conditions are allowed, and different policies may be applied to different types of secret key, but the re-authorisation conditions for all types of secret key shall be unambiguously defined in the Security Target. The decision to use supported re-authentication conditions is made on the basis of the application context. Making appropriate use of re-authorisation conditions supports client applications in meeting their requirements for OE.DataContext and OE.AppSupport.

OT.DataConf Protection of confidentiality of sensitive client application data

The TOE provides secure channels to client applications that can be used to protect the confidentiality of sensitive data (such as authentication/authorization data) during transmission between the client application

[19] Any attempt to use the key in cryptographic functions that are not permitted for that key is addressed by OT.KeyUseConstraint.

and the TOE, or during transmission between separate parts of the TOE where that transmission passes through an insecure environment.

Application Note 8

Protection of secret keys (as a specific type of sensitive data) is also subject to additional protection specified in other TOE objectives. Any requirements for secure storage and control of access to other types of client application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport. For example, if a client application uses the TOE to perform cryptographic functions on data that represent a passphrase value and the passphrase value is to be stored on the TOE, then the client application would need to use an appropriate encryption function before storing the data on the TOE.

OT.DataMod Protection of integrity of client application data

The TOE provides secure channels to client applications that can be used to protect the integrity of sensitive data (such as data to be signed, authentication/authorization data or public key certificates) during transmission between the client application and the TOE.

Application Note 9

Any requirements for integrity protection of client application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport.

OT.ImportExport Secure import and export of keys

The TOE allows import and export of secret keys only by using a secure method that protects the confidentiality and integrity of the data during transmission – in particular, secret keys shall be exported only in encrypted form (it is not sufficient to rely on properties of a secure channel to provide the protection: the key itself shall be encrypted). The TOE also allows individual secret keys under its control to be identified as non-exportable, in which case any attempt to export them will be rejected automatically. Public keys may be imported and exported in a manner that protects the integrity of the data during transmission.

Assigned keys cannot be imported or exported.

OT.Backup Secure backup of user data

Any method provided by the TOE for backing up user data, including secret keys, preserves the security of the data and is controlled by authorized Administrators. The secure backup process preserves the confidentiality and integrity of the data during creation, transmission, storage and restoration of the backup data. Backups also preserve the integrity of the attributes of keys.

OT.RNG Random number quality

Random numbers generated and provided to client applications for use as keys, authentication/authorization data, or seed data for another random number generator that is used for these purposes shall meet a defined quality metric in order to ensure that random numbers are not predictable and have sufficient entropy.

OT.TamperDetect Tamper Detection

The TOE shall provide features to protect its security functions against tampering. In particular the TOE shall make any physical manipulation within the scope of the intended environment (adhering to OE.Env) detectable for the administrators of the TOE.

OT.FailureDetect Detection of TOE hardware or software failures

The TOE detects faults that would cause some other security property to be weakened or to fail, including:

> Environmental conditions outside normal operating range (including temperature and power)

> Failures of critical TOE hardware components (including the RNG)

> Corruption of TOE software.

On detection of a fault, the TOE takes action to maintain its security and the security of the data that it contains and controls.

OT.Audit Generation of audit trail

The TOE creates audit records for security-relevant events, recording the event details and the subject associated with the event. The TOE ensures that the audit records are protected against accidental or malicious deletion or modification of records by providing tamper protection (either prevention or detection) for the audit log.

4.2 Security Objectives for the Operational Environment

The following security objectives relate to the TOE environment. This includes client applications as well as the procedure for the secure operation of the TOE.

OE.ExternalData Protection of data outside TOE control

Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities shall provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment. This includes protection of data that is exported from, or imported to, the TOE (such as audit data and encrypted keys).

In particular, any backups of the TOE and its data shall be maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data shall not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data shall require at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

OE.Env Protected operating environment

The TOE shall operate in a protected environment that limits physical access to the TOE to authorized Administrators. The TOE software and hardware environment (including client applications) shall be installed and maintained by Administrators in a secure state that mitigates against the specific risks applicable to the deployment environment, including (where applicable):

- Protection against loss or theft of the TOE or any of its externally stored assets
- Inspections to deter and detect tampering (including attempts to access side-channels, or to access connections between physically separate parts of the TOE, or parts of the hardware appliance)
- Protection against the possibility of attacks based on emanations from the TOE (e.g. electromagnetic emanations) according to risks assessed for the operating environment
- Protection against unauthorized software and configuration changes on the TOE and the hardware appliance
- Protection to an equivalent level of all instances of the TOE holding the same assets (e.g. where a key is present as a backup in more than one instance of the TOE).

OE.DataContext Appropriate use of TOE functions

Any client application using the cryptographic functions of the TOE shall ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and confidentiality). For example, when creating a digital signature over a DTBS the client application shall ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and shall correctly and securely manage the signature received from the TOE; and when certifying a public key the client application shall ensure that necessary checks are made to prove possession of the corresponding private key. The client application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) shall perform a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS.

Client applications shall be responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events.

Similar requirements shall apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys.

Appropriate procedures shall be defined for the initial creation of data and continuing operation of the TOE according to the specific risks applicable to the deployment environment and the ways in which the TOE is used.

OE.Uauth Authentication of application users

Any client application using the cryptographic services of the TOE shall correctly and securely gather identification and authentication/authorization data from its users and securely transfer it to the TOE (protecting the confidentiality of the authentication/authorization data as required) when required to authorize the use of TOE assets and services.

OE.AuditSupport Audit data review

The audit trail generated by the TOE will be collected, maintained and reviewed by a System Auditor according to a defined audit procedure for the TSP.

Application Note 10

As noted for P.Audit in section 3.4, the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.

OE.AppSupport Application security support

Procedures to ensure the ongoing security of client applications and their data shall be defined and followed in the environment, and reflected in use of the appropriate TOE cryptographic functions and parameters, and appropriate management and administration actions on the TOE. This includes, for example, any relevant policies on algorithms, key generation methods, key lengths, key access, key import/export, key usage limitations, key activation, cryptoperiods and key renewal, and key/certificate revocation.

5 Extended Components Definition

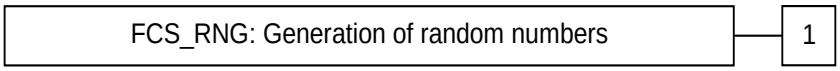
5.1 Generation of random numbers (FCS_RNG)

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

Family behavior:

This family defines quality requirements for the generation of random numbers which are intended to be used for cryptographic purposes.

Component levelling:



Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1	*Generation of random numbers*
Hierarchical to:	No other components.
Dependencies:	No dependencies.

FCS_RNG.1.1	The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].
FCS_RNG.1.2	The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet [assignment: *a defined quality metric*].

Application Note 11

A physical random number generator (RNG) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (key strokes, mouse movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs where a hybrid physical RNG produces at least the amount of entropy the RNG output may contain and the internal state of a hybrid deterministic RNG output contains fresh entropy but less than the output of RNG may contain.

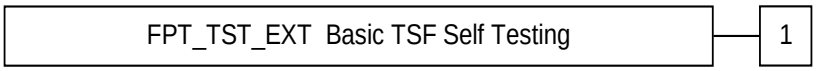
5.2 Basic TSF Self Testing (FPT_TST_EXT.1)

The extended component defined here is a simplified version of FPT_TST.1 in [CC2]

Family behavior:

Components in this family address the requirements for self-testing the TSF for selected correct operation.

Component levelling:



Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

> Indication that TSF self-test was completed.

FPT_TST_EXT.1	**Basic TSF Self Testing**
Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_TST_EXT.1.1	The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self-tests should occur]*] to demonstrate the correct operation of the TSF: [assignment: *list of self-tests run by the TSF*].

6 Security Requirements

This chapter gives the security functional requirements (SFR) and the security assurance requirements (SAR) for the TOE and the environment.

6.1 Typographical Conventions

The following conventions are used in the definitions of the SFRs:

- Refinements are denoted in one of two ways, depending on whether they add detail to an SFR ('explanatory refinements') or update the text of an SFR element ('element refinements'). Explanatory refinements follow the SFR that they update and are marked by the word "**Refinement**" in bold followed by text describing the refinement. Element refinements are indicated by **bold text** within an SFR element, with the original text indicated in a footnote.
- Selections and assignments that have already been made in the [CEN EN 419221-5] Protection Profile are *italicized*, and the original text on which the selection or assignment has been made is not reminded.
- Selections and assignments made in this ST are underlined, and the PP original text on which the selection or assignment has been made is indicated in a footnote.
- Iteration operations on SFR components are denoted by showing a slash "/" and the iteration indicator after the SFR component identifier.

6.2 SFR Architecture

6.2.1 SFR Relationships

Figure 6-1 and Figure 6-2 give a graphical presentation of the connections between the Security Functional Requirements (SFRs) from section 6.3 below and the underlying functional areas and operations that the TOE provides. The diagrams provide a context for SFRs that relates to their use in the TOE, whereas section 6.3 defines the SFRs grouped by the abstract class and family groupings in [CC2].

Figure 6-1: Architecture of Key Protection SFRs

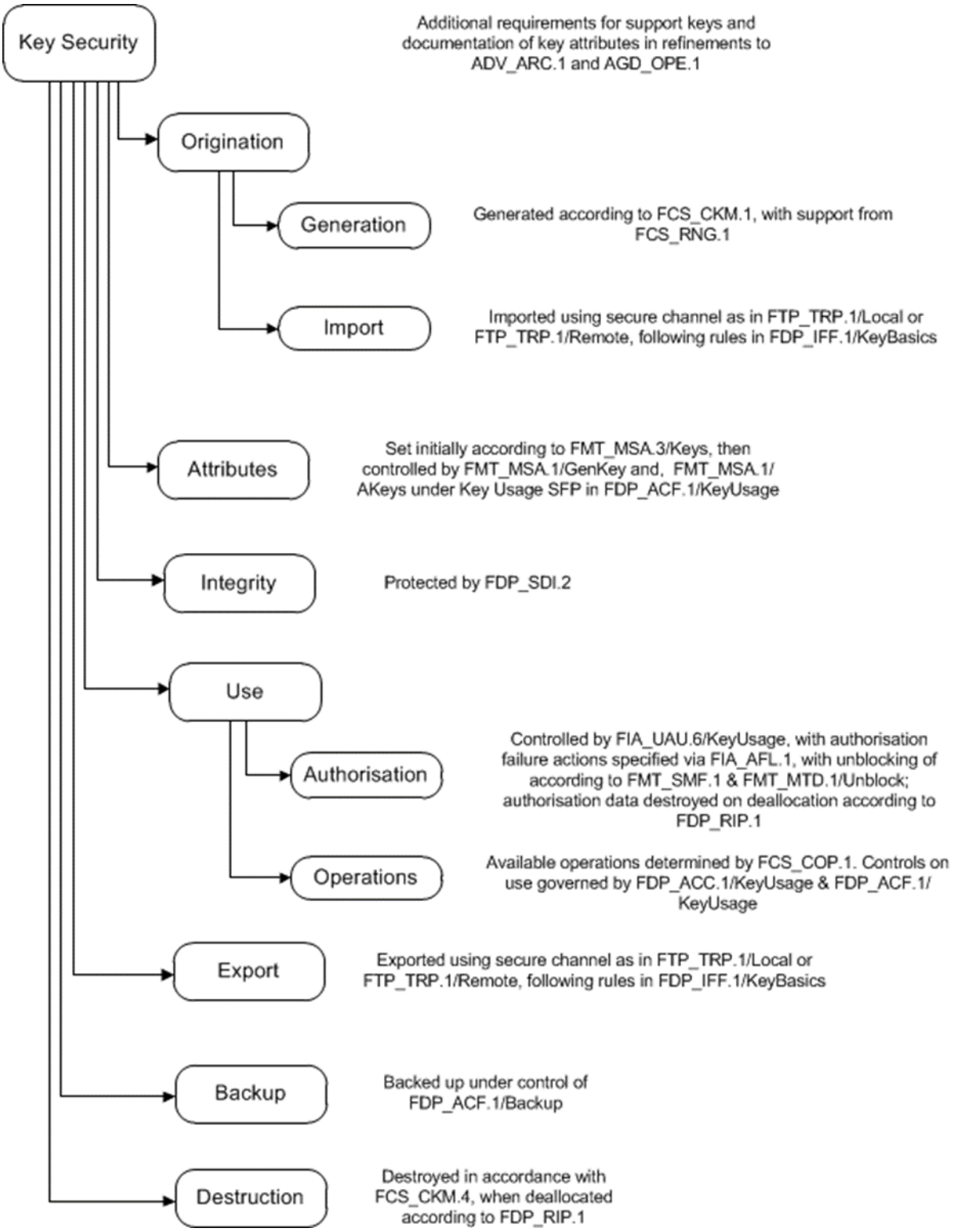
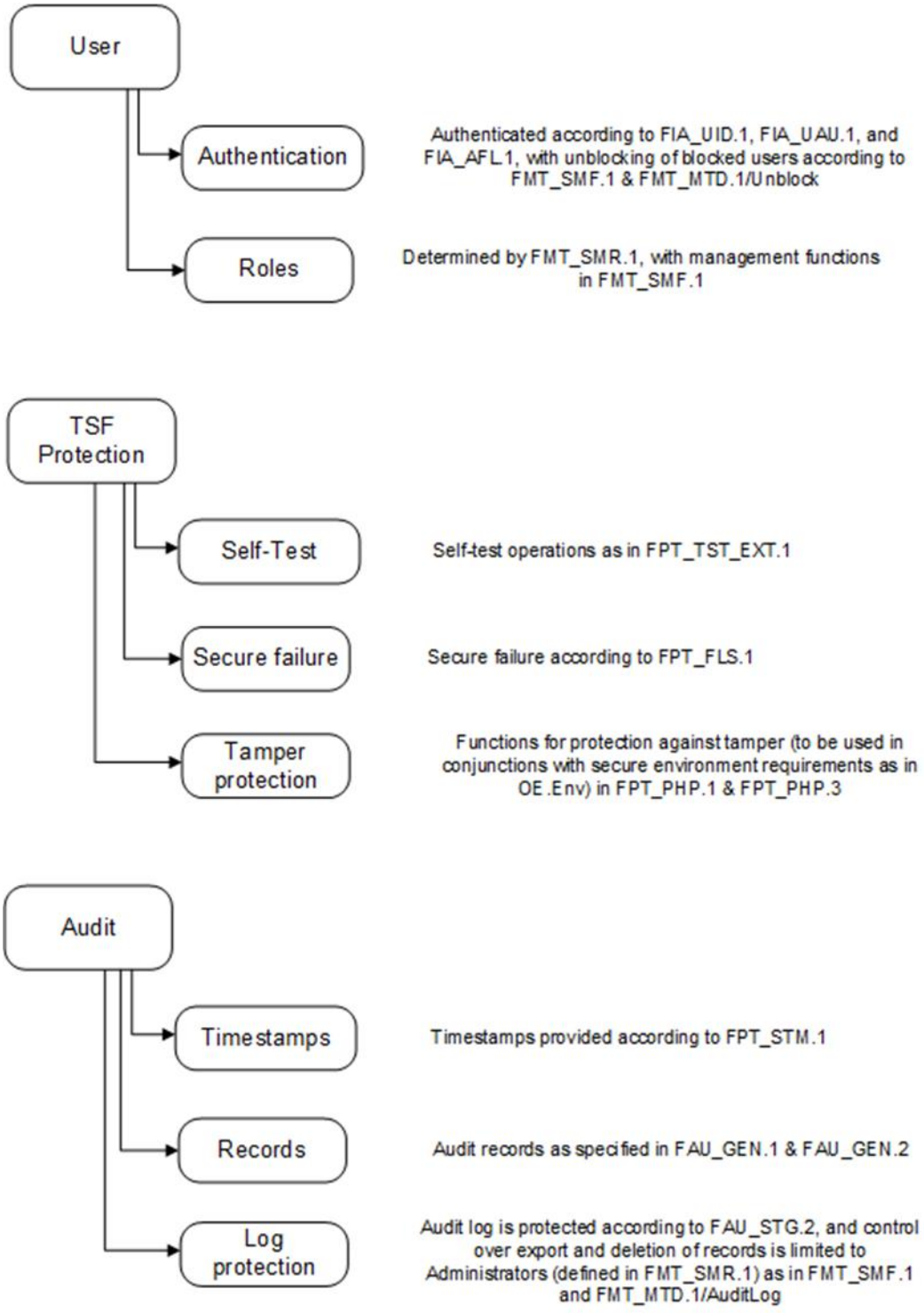


Figure 6-2: Architecture of User, TSF Protection & Audit SFRs

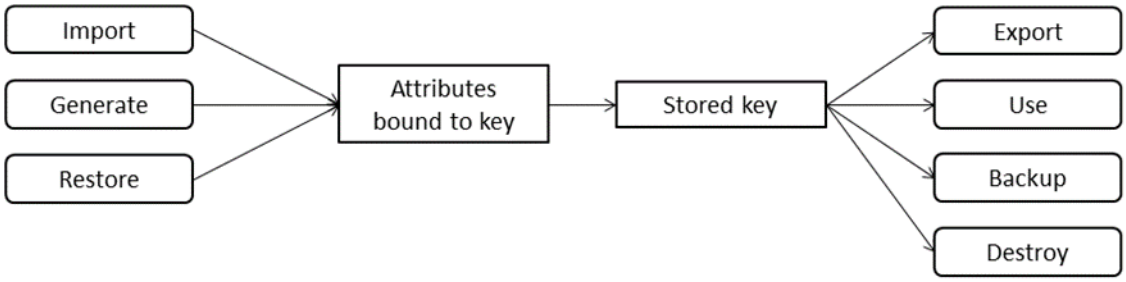


6.2.2 SFRs and the Key Lifecycle

The generic lifecycle for a key is illustrated in Figure 6-3. This shows the methods by which a key may arrive in the TOE (import, generation or restore from backup), resulting in binding of a set of attributes to the key

and storage of the key, and finally the ways in which a stored key may then be processed (export, use in a cryptographic function, backup, or destruction). The SFRs related to each of these aspects are then described below Figure 6-3.

Figure 6-3: Generic Key Lifecycle and Related SFRs



Import:

- FDP_IFF.1/KeyBasics requires a secure channel (FTP_TRP.1) and import in encrypted form or by using at least two components
- FAU_GEN.1 requires audit of import

Generate:

- FCS_CKM.1 requires approved algorithms
- FCS_RNG.1 defines requirements on random number generation
- FMT_MSA.3/Keys defines requirements on key attribute initialization
- FAU_GEN.1 requires audit of generation (and of failure of RNG)

Restore:

- FDP_ACF.1/Backup requires only an Administrator can restore from a backup, all backups shall preserve confidentiality and integrity of keys (as appropriate to key type) and their attributes, and any restore shall be under dual person control
- FAU_GEN.1 requires auditing of a restore (or of any integrity failure during a restore attempt)

Attributes bound to key:

- FMT_MSA.3/Keys defines requirements on key attribute initialization
- FDP_ACF.1/KeyUsage, FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys define requirements on key attribute modification
- FAU_GEN.1 requires audit of changes to key attributes

Stored key:

- FDP_IFF.1/KeyBasics requires no plaintext access
- FDP_SDI.2 requires protection of the integrity of keys and their attributes
- FAU_GEN.1 requires audit of integrity errors detected

Export:

- FDP_IFF.1/KeyBasics requires a secure channel (FTP_TRP.1), authorization before export, no export of Assigned Keys, export controlled by the export flag attribute, and export in encrypted form
- FAU_GEN.1 requires audit of export

Use:

- FIA_AFL.1 requires blocking of access to a key on reaching an authorization failure threshold (FDP_IFF.1/KeyBasics and FMT_MTD.1/Unblock define requirements on unblocking)
- FDP_ACF.1/KeyUsage requires authorization before use of a key and that the key can only be used as identified in its Key Usage attribute
- FIA_UAU.6/KeyAuth requires authorization before initial use of a key and describes any additional requirements for re-authorization conditions such as expiry of a time period or number of uses of a key (or when the authorization period has been explicitly ended)
- FDP_RIP.1 requires protection of authorization data on deallocation
- FDP_IFF.1/KeyBasics requires no access to intermediate values in any operation using a secret key
- FCS_COP.1 requires the use of approved algorithms
- FAU_GEN.1 requires audit of authorization failure (and blocking or unblocking)

Backup:

- FDP_ACF.1/Backup requires only Administrator can make a backup; all backups must preserve confidentiality and integrity of keys (as appropriate to key type) and their attributes
- FAU_GEN.1 requires auditing of a backup

Destroy:

- FDP_RIP.1 requires key to be protected on deallocation
- FCS_CKM.4 requires key zeroisation on deallocation
- FAU_GEN.1 requires audit of key destruction

6.3 Security Functional Requirements

The individual security functional requirements are specified in the sections below.

6.3.1 Cryptographic Support (FCS)

FCS_CKM.1	*Cryptographic key generation*
Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm Key Generation Support Table and

Table 6-2: Key Derivation Support Table[20] and specified cryptographic key sizes Key Generation Support Table and

Table 6-2: Key Derivation Support Table[21] that meet the following: Key Generation Support Table and

Table 6-2: Key Derivation Support Table[22].

Table 6-1: Key Generation Support Table

Key Generation Algorithm	***Key Sizes***	***Applicable Standards***
RSA Key Generation	*Modulus length 2048, 3072, 4096*	*FIPS Pub 186-4 [FIPS 186-4] Appendix B.3.3 and B.3.6 with primality tests from C.3.*
ECC Key Generation	*NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B409, B-571* *Brainpool curves:* *brainpoolP160r1, brainpoolP160t1, brainpoolP192r1, brainpoolP192t1, brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP320r1, brainpoolP320t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1*	*FIPS Pub 186-4 [FIPS 186-4] Appendix B.4.1 and B.4.2, Appendix D* *RFC- 5639 [IETF RFC Brainpool]*
DSA Domain Parameter Generation	*Modulus length 2048 and 3072*	*FIPS Pub 186-4 [FIPS 186-4] chapter 2.1 and Appendix A.1.1.2*
DSA Key-Pair Generation	*Modulus length 2048 and 3072*	*FIPS Pub 186-4 [FIPS 186-4] Section A.2.1*
Diffie-Hellman (DH) Domain Parameter Generation	*Modulus length 2048, 3072 and 4096 bits*	*FIPS Pub 186-4 [FIPS 186-4] Appendix A.1*
Diffie-Hellman (DH) Key-Pair Generation	*Modulus length 2048,3072 and 4096 bits*	*FIPS Pub 186-4 [FIPS 186-4] Appendix A.2.1*

[20] [assignment: *cryptographic key generation algorithm*]
[21] [assignment*: cryptographic key sizes*]
[22] [assignment*: list of standards*]

Key Generation Algorithm	Key Sizes	Applicable Standards
AES	*128,192 and 256 bits*	*FIPS Pub 197 [FIPS 197] chapters 3.1 and 6*
Generic Secret	*128 – 4096 bits (in increments of 8 bits)*	*N/A*

Table 6-2: Key Derivation Support Table

Key Derivation Algorithm	Key Sizes	Supported PRF / Hashing Function / Cipher	Applicable Standards
Counter Mode KDF	*128,192 and 256 bits when AES is cipher.* *128 - 4096 bits when HMAC PRF is used*	*AES-CMAC, HMAC-SHA1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA512.*	*Counter Mode KDF from FIPS SP 800-108 [SP800-108] chapter 4 and 5.1 with FIPS Pub 197 [FIPS 197] for supported cipher.*
Single-step KDF	*None*	*SHA-512*	*[SP800-56C] chapter 4*
EC Diffie-Hellman Key Agreement	*Curve P-224*	*SHA-224, SHA-256, SHA-384, SHA-512*.	*ECC - Ephemeral Unified, One Pass DH from NIST Special Pub 800-56A [SP800-56A]*
EC Diffie-Hellman Key Agreement	*Curve P-256*	*SHA-256, SHA-384, SHA-512.*	*ECC - Ephemeral Unified, One Pass DH from NIST Special Pub 800-56A [SP800-56A]*
EC Diffie-Hellman Key Agreement	*Curve P-384*	*SHA-384, SHA-512.*	*ECC - Ephemeral Unified, One Pass DH from NIST Special Pub 800-56A [SP800-56A]*
EC Diffie-Hellman Key Agreement	*Curve P521*	*SHA-512.*	*ECC - Ephemeral Unified, One Pass DH and Full Unified from NIST Special Pub 800-56A [SP800-56A]*

Key Derivation Algorithm	***Key Sizes***	***Supported PRF / Hashing Function / Cipher***	***Applicable Standards***
EC Diffie-Hellman Key Agreement	*brainpoolP160r1, brainpoolP160t1, brainpoolP192r1, brainpoolP192t1, brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP320r1, brainpoolP320t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1*	*SHA-224, SHA-256, SHA-384, SHA-512.*	*ECC - Ephemeral Unified, One Pass DH from NIST Special Pub 800-56A [SP800-56A]* *RFC- 5639 [IETF RFC Brainpool]*
Diffie-Hellman Key Agreement	*Modulus 2048, 3072 and 4096 bits*	*SHA-224, SHA-256, SHA-384, SHA-512*.	*FFC - dhHybrid1, dhEphem, dhHybrid1Flow and dhOneFlow from NIST Special Pub 800-56A [SP800-56A]*

Application Note 12

> *Key generation is linked to the setting of security attributes of a key (including the link to a subject who owns the key, via the setting of authorization data) as in FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys.*

> *The internal RNG of the TOE (see FCS_RNG.1/DRG.4) is used in the key generation process*

FCS_CKM.4 *Cryptographic key destruction*

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *zeroisation* that meets the following: action taken according to the key zeroisation table below[23].

[23] [assignment*: list of standards*]

Table 6-3: Key Zeroisation Table

Key deletion method (KDM)	***Action taken***	***Context in which the KDM is used***
KDM1	*Overwrite of memory*	> *Applies to user keys stored in HSM partitions, which can be erased by means of ICD commands.* > *Applies to some HSM support keys as well, that can be erased by means of ICD commands.*
KDM2	*RAM reset*	*Resetting of RAM causes the erasure of all RAM resident keys*
KDM3	*Erasure of entire memory sectors*	*HSM wipe-out, which can be called from the bootloader, or triggered by the active tamper detection on K7+ TOE. All user keys and HSM support keys are erased.*
KDM4	*Erasure of the encrypting keys*	> *Any user key stored in a HSM partition is encrypted with a chain of HSM support keys. Erasure of any of these support keys is equivalent to erasing the user key.* > *The same applies to user keys stored externally, which are encrypted with a HSM support key.* > *The same applies to many 'intermediate' HSM support keys which are encrypted by other HSM support keys.*
KDM5	*Erasure of HSE-BBRAM in response to a tamper /decommission event*	*Decommission signal (on K7 and K7+ TOEs) or active tamper detection (on K7+ TOE) trigger the erasure of the HSE-BBRAM, which contains the 'top-level' HSM support keys that encrypt the other HSM support keys.*

Application Note 13

All user keys and all HSM support keys are covered by one or several destruction methods from the above table. As required by the ADV_ARC refinement in section 6.4.1, the ADV_ARC document will include a table listing all the HSM support keys; this table will indicate which destruction method(s) (among the five) is/are available for each HSM support key.

FCS_COP.1/SigGen_Main *Cryptographic operation – Digital Signature Generation (Main)*

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SigGen_Main The TSF shall perform digital signature generation[24] in accordance with a specified cryptographic algorithm as shown in the Digital Signature Generation Algorithm Support Table[25] and cryptographic key sizes as shown in the Digital Signature Generation Algorithm Support Table[26] that meet the following: standards as shown in the Digital Signature Generation Algorithm Support Table[27].

Table 6-4: Digital Signature Generation Algorithm Support Table

Signature Generation Algorithm	***Key Sizes***	***Hash Algorithm***	***Applicable Standards***
RSA	*Modulus length 2048 and 3072 bits*	*SHA-224, SHA-256, SHA-384, SHA-512*, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*RSASSA-PSS and RSASSA-PKCS1-v1_5 from PKCS #1 [PKCS#1] chapters 8.1.1 and 8.2.1*
RSA	*Modulus length 4096 bits*	*SHA-224, SHA-256, SHA-384, SHA-512,* SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*RSASSA-PSS and RSASSA-PKCS1-v1_5 from PKCS #1 [PKCS#1]* *chapters 8.1.1 and 8.2.1*
RSA	*Modulus length 2048 and 3072 bits*	*SHA-224, SHA-256, SHA-384, SHA-512,* SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*ANSI X9.31 Signature Generation as covered under FIPS Pub 186-4 [FIPS 186-4] chapter 5.4*
RSA	*Modulus length 4096 bits*	*SHA-224, SHA-256, SHA-384, SHA-512,* SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*ANSI X9.31 Signature Generation as covered under FIPS Pub 186-4 [FIPS 186-4] chapter 5.4*
DSA	*Modulus length 2048 and 3072 bits*	*SHA-224, SHA-256, SHA-384,* SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*FIPS Pub 186-4 [FIPS 186-4] chapter 4.6*

[24] [assignment: *list of cryptographic operations*]
[25] [assignment*: cryptographic algorithm*]
[26] [assignment*: cryptographic key sizes*]
[27] [assignment: *list of standards*].

Signature Generation Algorithm	***Key Sizes***	***Hash Algorithm***	***Applicable Standards***
ECDSA	*NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B409, B-571* *Brainpool curves: brainpoolP160r1, brainpoolP160t1, brainpoolP192r1, brainpoolP192t1, brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP320r1, brainpoolP320t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1*	*SHA-224, SHA-256, SHA-384, SHA-512,* SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*FIPS Pub 186-4 [FIPS 186-4] chapter 6.4 and Appendix D* *RFC- 5639 [IETF RFC Brainpool]*

FCS_COP.1/SigVer_Main *Cryptographic operation – Digital Signature Verification (Main)*

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SigVer_Main The TSF shall perform digital signature verification[28] in accordance with a specified cryptographic algorithm as shown in the Digital Signature Verification Algorithm Support Table[29] and cryptographic key sizes as shown in the Digital Signature Verification Algorithm Support Table[30] that meet the following: standards as shown in the Digital Signature Verification Algorithm Support Table[31].

Table 6-5: Digital Signature Verification Algorithm Support Table

Signature Verification Algorithm	***Key Sizes***	***Hash Algorithm***	***Applicable Standards***
RSA	*Modulus length 1024, 2048, 3072 and 4096 bits*	*SHA-1, SHA-224, SHA-256, SHA-384, SHA-512,* SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*RSASSA-PSS and RSASSA-PKCS1-v1_5 from PKCS #1 [PKCS#1] chapters 8.1.2 and 8.2.2*

[28] [assignment: *list of cryptographic operations*]
[29] [assignment*: cryptographic algorithm*]
[30] [assignment*: cryptographic key sizes*]
[31] [assignment: *list of standards*].

Signature Verification Algorithm	***Key Sizes***	***Hash Algorithm***	***Applicable Standards***
RSA	*Modulus length 1024, 2048, 3072 and 4096 bits*	*SHA-1, SHA-224, SHA-256, SHA-384, SHA-512,* SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*ANSI X9.31 Signature Generation as covered under FIPS Pub 186-4 [FIPS 186-4] chapter 5.4*
DSA	*Modulus length 1024, 2048 and 3072 bits*	*SHA-1, SHA-224, SHA-256, SHA-384,* SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*FIPS Pub 186-4 [FIPS 186-4] chapter 4.7*
ECDSA	*NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B409, B-571* *Brainpool curves: brainpoolP160r1, brainpoolP160t1, brainpoolP192r1, brainpoolP192t1, brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP320r1, brainpoolP320t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1*	*SHA-1, SHA-224, SHA-256, SHA-384, SHA-512*, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	*FIPS Pub 186-4 [FIPS 186-4] chapter 6.4 and Appendix D* *RFC- 5639 [IETF RFC Brainpool]*

FCS_COP.1/SigVer_Bootloader *Cryptographic operation – Signature Verification (Bootloader)*

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SigVer_Bootloader The TSF shall perform digital signature verification[32] in accordance with a specified cryptographic algorithm RSA[33] and cryptographic key sizes 4096 bits with SHA-384[34] that meet the following: RSASSA-PKCS1-v1_5 from PKCS #1 [PKCS#1] chapter 8.2.2[35].

[32] [assignment: *list of cryptographic operations*]
[33] [assignment*: cryptographic algorithm*]
[34] [assignment*: cryptographic key sizes*]
[35] [assignment: *list of standards*].

Application Note 14

This iteration covers the signature verification algorithm contained within the Bootloader used to verify the authenticity (including integrity) of the main cryptographic module firmware on power-on in support of FPT_TST_EXT.1.

This function is used to verify both the signature on the stored firmware ahead of execution alongside to validate the associated certificate chain back to the Thales root certificate.

FCS_COP.1/Digest_Main	*Cryptographic operation – Digest (Main)*

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/Digest_Main The TSF shall perform message digest[36] in accordance with a specified cryptographic algorithm as shown in the Message Digest Algorithm Support Table[37] and cryptographic key sizes as shown in the Message Digest Algorithm Support Table[38] that meet the following: standards as shown in Message Digest Algorithm Support Table[39].

Table 6-6: Message Digest Algorithm Support Table

Message Digest Algorithm	***Key Sizes***	***Applicable Standards***
SHA-224	*None*	*FIPS Pub 180-4 [FIPS 180-4] chapter 6.3*
SHA-256	*None*	*FIPS Pub 180-4 [FIPS 180-4] chapter 6.2*
SHA-384	*None*	*FIPS Pub 180-4 [FIPS 180-4] chapter 6.5*
SHA-512	*None*	*FIPS Pub 180-4 [FIPS 180-4] chapter 6.4*
SHA3-224	*None*	*FIPS Pub 202 FIPS 202, chapter 5.2 and 6.1*
SHA3-256	*None*	*FIPS Pub 202 FIPS 202, chapter 5.2 and 6.1*
SHA3-384	*None*	*FIPS Pub 202 FIPS 202, chapter 5.2 and 6.1*
SHA3-512	*None*	*FIPS Pub 202 FIPS 202, chapter 5.2 and 6.1*

[36] [assignment: *list of cryptographic operations*]
[37] [assignment*: cryptographic algorithm*]
[38] [assignment*: cryptographic key sizes*]
[39] [assignment: *list of standards*]

FCS_COP.1/Sym_Enc_Dec *Cryptographic operation – Symmetric Encrypt/Decrypt*

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/Sym_Enc_Dec The TSF shall perform symmetric encryption and decryption[40] in accordance with a specified cryptographic algorithm as shown in the Symmetric Algorithm Support Table[41] and cryptographic key sizes as shown in the Symmetric Algorithm Support Table[42] that meet the following: standards as shown in the Symmetric Algorithm Support Table[43].

Table 6-7: Symmetric Algorithm Support Table

Symmetric Algorithm	***Key Sizes***	***Supported Mode***	***Applicable Standards***
AES	*128, 192 and 256 bits*	*ECB, CBC, OFB, CFB8, CFB128, CTR*	*FIPS Pub 197 [FIPS 197] chapter 5 and NIST SP 800-38A [SP800-38A] chapter 6*
AES	*128, 192 and 256 bits*	*GCM with 128-bit tag*	*FIPS Pub 197 [FIPS 197] chapter 5 and NIST SP 800-38D [SP800-38D] chapter 7*
AES	*128 and 256 bits*	*XTS-AES*	*FIPS Pub 197 [FIPS 197] chapter 5 and NIST SP 800-38E [SP800-38E]*
AES	*128, 192 and 256 bits*	*AES-KW and AES-KWP*	*FIPS Pub 197 [FIPS 197] chapter 5 and NIST SP 800-38F [SP800-38F] chapter 6*

FCS_COP.1/ASym_Enc_Dec *Cryptographic operation – Asymmetric Encrypt/Decrypt*

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

[40] [assignment: *list of cryptographic operations*]
[41] [assignment*: cryptographic algorithm*]
[42] [assignment*: cryptographic key sizes*]
[43] [assignment: *list of standards*]

FCS_COP.1.1/ASym_Enc_Dec The TSF shall perform *asymmetric encryption and decryption*[44] in accordance with a specified cryptographic algorithm RSA[45] and cryptographic key sizes 2048-4096[46] that meet the following: RSAES-OAEP from PKCS #1 v2.1 and KTS-OAEP-basic from NIST SP800-56B [SP800-56B] chapter 9.2 and KAS-1 basic from NIST SP800-56B [SP800-56B] chapter 8.2[47].

FCS_COP.1/MAC *Cryptographic operation – Message Authentication Code (MAC)*

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/MAC The TSF shall perform message authentication code generation and verification[48] in accordance with a specified cryptographic algorithm as shown in the Message Authentication Code Algorithm Support Table[49] and cryptographic key sizes as shown in the Message Authentication Code Algorithm Support Table [50] that meet the following: standards as shown in Message Authentication Code Algorithm Support Table[51].

Table 6-8: Message Authentication Code Algorithm Support Table

MAC Algorithm	Key Sizes	Supported PRF / Hashing Function	Applicable Standards
HMAC	128 – 4096 bits (in increments of 8 bits)	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	FIPS Pub 198-1 [FIPS 198-1]
AES CMAC	128, 192 and 256 bits	N/A	NIST SP 800-38B [SP800-38B] chapter 6
AES GMAC	128, 192 and 256 bits	N/A	NIST SP 800-38D [SP800-38D] chapter 7

FCS_RNG.1/PTG.2 *Generation of random numbers (PTG.2)*

Hierarchical to: No other components.

[44] [assignment: *list of cryptographic operations*]
[45] [assignment*: cryptographic algorithm*]
[46] [assignment*: cryptographic key sizes*]
[47] [assignment: *list of standards*].
[48] [assignment: *list of cryptographic operations*]
[49] [assignment*: cryptographic algorithm*]
[50] [assignment*: cryptographic key sizes*]
[51] [assignment: *list of standards*]

Dependencies: No dependencies

FCS_RNG.1.1/PTG.2 The TSF shall provide a physical[52] random number generator that implements [AIS31] PTG.2 security capabilities:

(PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.

(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source.

(PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.

(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.

(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered continuously. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.[53]

FCS_RNG.1.2/PTG.2 The TSF shall provide octets of bits[54] that meet [AIS31] PTG.2 quality metric:

(PTG.2.6) Test procedure A and none does not distinguish the internal random numbers from output sequences of an ideal RNG.

(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997.[55].

FCS_RNG.1/DRG.4 *Generation of random numbers (DRG.4)*

Hierarchical to: No other components.

Dependencies: No dependencies

FCS_RNG.1.1/DRG.4 The TSF shall provide a hybrid deterministic[56] random number generator that implements [AIS31] DRG.4 security capabilities:

(DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2 as random source.

(DRG.4.2) The RNG provides forward secrecy.

(DRG.4.3) The RNG provides backward secrecy even if the current internal state is known.

(DRG.4.4) The RNG provides enhanced forward secrecy:

- On demand,
- On condition: after 2^32 generate requests or 2^32 bits generated, whichever comes first
- After 10 seconds.

[52] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]
[53] [assignment: *list of security capabilities*]
[54] [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*]
[55] [assignment: *a defined quality metric*]
[56] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]

(DRG.4.5) The internal state of the RNG is seeded by a PTRNG of class PTG.2.[57].

FCS_RNG.1.2/DRG.4 The TSF shall provide octets of bits[58] that meet [AIS31] DRG.4 quality metric:

(DRG.4.6) The RNG generates output for which k >2^{34} strings of bit length 128 are mutually different with probability 1−ε, with ε < 2^{-16}.

(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A and NIST SP 800-22 test suite[59].

6.3.2 Identification and authentication (FIA)

FIA_UID.1/STC_User *Timing of identification*

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_UID.1.1/STC_User The TSF shall allow

1. *Self-test according to FPT_TST_EXT.1*
2. The following list of additional actions:
   a. Submit bootloader commands for:
      i. Diagnostics and status information
      ii. HSM wipe-out
      iii. Completing the boot process by passing the execution control to the Firmware
      iv. Instructing the Firmware to delete FM applications stored in the TOE's flash memory
   b. query HSM status, authenticated identity of the HSM, configuration and licenses
   c. query container configuration
   d. PED configuration and communication requests
   e. query log status and submit external log messages for addition to secure audit log
   f. STC management operations (request public key, activate channel, open and close channel)
   g. request state of HSM roles
   h. Submit public requests to embedded FM applications [60]

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

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

[57] [assignment: *list of security capabilities*]
[58] [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*]
[59] [assignment: *a defined quality metric*]
[60] [assignment: list of additional TSF-mediated actions]

Application Note 15

'HSM wipe-out' means that all HSM content (user information, user keys, HSM keys and certificates, firmware) are erased. Note that the bootloader still remains.

FIA_UAU.1/STC_User *Timing of authentication*

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1/STC_User The TSF shall allow

1. *Self-test according to FPT_TST_EXT.1,*
2. *Identification of the user by means of TSF required by FIA_UID.1 / STC_User*
3. The following list of additional actions:
   a. Submit bootloader commands for:
      i. Diagnostics and status information
      ii. HSM wipe-out
      iii. Completing the boot process by passing the execution control to the Firmware
      iv. Instructing the Firmware to delete FM applications stored in the TOE's flash memory
   b. query HSM status, authenticated identity of the HSM, configuration and licenses
   c. query container configuration
   d. PED configuration and communication requests
   e. query log status and submit external log messages for addition to secure audit log
   f. STC management operations (request public key, activate channel, open and close channel)
   g. request state of HSM roles
   h. Submit public requests to embedded FM applications [61]

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

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

Application Note 16

- *Identification and authentication are done at the same time, which explains that the list of allowed actions is the same for FIA_UID.1/STC_User and FIA_UAU.1/STC_User.*
- *'HSM wipe-out' means that all HSM content (user information, user keys, HSM keys and certificates, firmware) are erased. Note that the bootloader still remains.*

[61] [assignment: *list of additional TSF-mediated actions*]

FIA_UID.1/HSM_Roles *Timing of identification*

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_UID.1.1/HSM_Roles The TSF shall allow

1. *Self-test according to FPT_TST_EXT.1*
2. The following list of additional actions:
   a. Submit bootloader commands for:
      i. Diagnostics and status information
      ii. HSM wipe-out
      iii. Completing the boot process by passing the execution control to the Firmware
      iv. Instructing the Firmware to delete FM applications stored in the TOE's flash memory
   b. query HSM status, authenticated identity of the HSM, configuration and licenses
   c. query container configuration
   d. PED configuration and communication requests
   e. query log status and submit external log messages for addition to secure audit log
   f. request state of HSM roles
   g. query container object identify (from known OUID or object handle)
   h. session management functions (i.e. open, close, close all, clean access, get session info)
   i. Login requests
   j. HSM deactivation
   k. Zeroize the HSM
   l. Request new initialization of HSM
   m. create, modify, destroy and get attributes of public partition objects
   n. Submit public requests to embedded FM applications[62]

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

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

Application Note 17

- *This SFR applies to the HSM explicit roles (i.e. HSM SO, Partition SO, Administrator, Audit User, Partition CO, Partition Limited CO and Partition CU).*
- *'Public partition objects' mentioned in this SFR are limited to Data Objects (i.e. Class=CKO_DATA) stored in the partition areas.*

[62] [assignment: list of additional TSF-mediated actions]

- *'HSM zeroisation' means that all user information and user key material are erased. The bootloader, firmware and HSM own data (HSM keys and certificates) are not erased.*
- *'HSM wipe-out' means that all HSM content (user information, user keys, HSM keys and certificates, firmware) are erased. Note that the bootloader still remains.*

FIA_UAU.1/HSM_Roles	*Timing of authentication*

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1/HSM_Roles The TSF shall allow

1. *Self-test according to FPT_TST_EXT.1,*
2. *Identification of the user by means of TSF required by FIA_UID.1 / HSM_Roles*
3. The following list of additional actions:
   a. Submit bootloader commands for:
      i. Diagnostics and status information
      ii. HSM wipe-out
      iii. Completing the boot process by passing the execution control to the Firmware
      iv. Instructing the Firmware to delete FM applications stored in the TOE's flash memory
   b. query HSM status, authenticated identity of the HSM, configuration and licenses
   c. query container configuration
   d. PED configuration and communication requests
   e. query log status and submit external log messages for addition to secure audit log
   f. request state of HSM roles
   g. query container object identify (from known OUID or object handle)
   h. session management functions (i.e. open, close, close all, clean access, get session info)
   i. Login requests
   j. HSM deactivation
   k. Zeroize the HSM
   l. Request new initialization of HSM
   m. create, modify, destroy and get attributes of public partition objects
   n. Submit public requests to embedded FM applications[63]

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

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

[63] [assignment: *list of additional TSF-mediated actions*]

Application Note 18

- *This SFR applies to the HSM explicit roles (i.e. HSM SO, Partition SO, Administrator, Audit User, Partition CO, Partition Limited CO and Partition CU). Moreover, Identification and authentication are done at the same time, which explains that the list of allowed actions is the same for FIA_UID.1/HSM_Roles and FIA_UAU.1/HSM_Roles.*
- *'Public partition objects' mentioned in this SFR are limited to Data Objects (i.e. Class=CKO_DATA) stored in the partition areas.*
- *'HSM zeroisation' means that all user information and user key material are erased. The bootloader, firmware and HSM own data (HSM keys and certificates) are not erased.*
- *'HSM wipe-out' means that all HSM content (user information, user keys, HSM keys and certificates, firmware) are erased. Note that the bootloader still remains.*

FIA_UID.1/Key_Owner *Timing of identification*

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_UID.1.1/Key_Owner The TSF shall allow

1. *Self-test according to FPT_TST_EXT.1*
2. Any operation except for:
   a. Cryptographic operation using an Assigned Key or a General Key
   b. Export operation of a General Key
   c. Modification of the authorization data of an Assigned Key or a General Key[64]

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

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

FIA_UAU.1/Key_Owner *Timing of authentication*

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1/Key_Owner The TSF shall allow

1. *Self-test according to FPT_TST_EXT.1,*
2. *Identification of the user by means of TSF required by FIA_UID.1 / Key_Owner*
3. Any operation except for:
   a. Cryptographic operation using an Assigned Key or a General Key
   b. Export operation of a General Key

[64] [assignment: list of additional TSF-mediated actions]

c. Modification of the authorization data of an Assigned Key or a General Key[65]

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

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

Application Note 19

Identification and authentication are done at the same time, which explains that the list of allowed actions is the same for FIA_UID.1/Key_Owner and FIA_UAU.1/Key_Owner.

FIA_AFL.1 *Authentication failure handling*

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when the number as specified in Table 6-9 [66] unsuccessful authentication **or authorization** attempts occur related *to consecutive failed authentication or authorization attempts*.

FIA_AFL.1.2 When the defined number of unsuccessful authentication **or authorization** attempts has been met[67], the TSF shall *block access to* the functionality specified in Table 6-9 [68] *until* the unblocking condition as specified in table 6-9 is met [69].

Table 6-9: Handling of authentication/authorization failures

Role	***Number of consecutive authentication/authorization failures (as considered in FIA_AFL.1.1)***	***Functionality being blocked (as considered in FIA_AFL.1.2)***	***Unblocking condition (as considered in FIA_AFL.1.2)***
HSM SO	*A positive integer within the range [1 to 3], configurable by the HSM SO*	*All HSM functionalities except HSM re-initialization.*	*None. HSM is totally zeroized.*
Partition SO	*A positive integer within the range [1 to 10], configurable by the Partition SO*	*All the functionalities of the related partition*	*None. Partition is totally zeroized*

[65] [assignment: list of additional TSF-mediated actions]
[66] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[67] [selection: met, surpassed]
[68] [assignment: description of the relevant functionality]
[69] [selection: unblocked by [assignment: identification of the authorized subject or role], a time period [assignment: time period] has elapsed]

Role	***Number of consecutive authentication/authorization failures (as considered in FIA_AFL.1.1)***	***Functionality being blocked (as considered in FIA_AFL.1.2)***	***Unblocking condition (as considered in FIA_AFL.1.2)***
Administrator	*Same number as for the HSM SO*	*Administrator role is locked out.*	*If the module policy "Partition SO can reset PIN" is enabled, the HSM SO can unlock the Administrator role by resetting its authentication data.* *Otherwise, there is no unblocking capability, and the Administrator role must be re-initialized.*
Audit User	*Same number as for the HSM SO.*	*Audit user login and related capabilities (role is locked out)*	*After a 60 seconds time period has elapsed*
Partition CO	*Same number as for the partition SO*	*Partition CO, Partition Limited CO and Partition CU roles are locked out.*	*If the module policy "Partition SO can reset PIN" is enabled, the Partition SO can unlock the Partition CO role by resetting its authentication data.* *Otherwise, there is no unblocking capability. All user keys contained in the partition are lost and the partition CO role must be re-initialized.*
Partition Limited CO	*Same number as for the partition SO*	*Partition Limited CO role is locked out. Partition CO and Partition CU roles are still functional.*	*The Partition CO can unlock the Partition Limited CO role by resetting its credentials.*
Partition CU	*Same number as for the partition SO*	*Partition CU role is locked out. Partition CO and Partition Limited CO roles are still functional.*	*The Partition CO can unlock the Partition CU role by resetting its credentials.*

Role	*Number of consecutive authentication/authorization failures (as considered in FIA_AFL.1.1)*	*Functionality being blocked (as considered in FIA_AFL.1.2)*	*Unblocking condition (as considered in FIA_AFL.1.2)*
Key Owner	*3*	*The related key is blocked: all operations on that key or using that key are forbidden (cryptographic operations, export, change of authorization data)*	*The Partition CO can unblock the key by setting the number of failed authorizations to any integer value in the range [0...2], or by resetting the key authorization data for General Keys.*
STC User	*1*	*Blocking of STC authentication mechanism for the identified STC client*	*After a 30 second time period has elapsed*

Application Note 20

'Zeroisation' (applied to the HSM or to a partition) means that all user information and user key material are erased.

FIA_UAU.6/KeyAuth *Re-authenticating*

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_UAU.6.1/KeyAuth The TSF shall **authorize and re-authorize**[70] the user **for access to a secret key** under the conditions

1. *Authorization in order to be granted initial access to the key; and*
2. Re-authorization of both General Keys and Assigned keys under the following conditions: after explicit rescinding of previous authorization for access to the secret key [71]

[70] re-authenticate

[71] [selection:
- Re-authorization of [assignment: identification of secret keys that are subject to re-authorization conditions below] both General Keys and Assigned keys under the following conditions: [selection:
  - after expiry of the time period (as specified in the secret key's attributes) for which the secret key was last authorized;
  - after the number of uses of the secret key (as specified in the secret key's attributes) for which the secret key was last authorized has already been made;
  - after explicit rescinding of previous authorization for access to the secret key];
- [assignment: list of other conditions under which authorization and re-authorization for access to secret keys is required]
- Authorization on every subsequent access to the key]

Application Note 21

It is the responsibility of the client application to make appropriate use of any re-authentication conditions according to the application context (cf. OE.DataContext and OE.AppSupport).

6.3.3 User data protection (FDP)

FDP_IFC.1/KeyBasics *Subset information flow control*

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/KeyBasics The TSF shall enforce the *Key Basics SFP* on

1. *subjects: all*
2. *information: keys*
3. *operations: all*

FDP_IFF.1/KeyBasics *Simple security attributes*

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

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

1. *whether a key is a secret or a public key*
2. *whether a secret key is an Assigned Key*
3. *whether channels selected to export keys are secure*
4. *the value of the Export Flag of a key*

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

1. *Export of secret keys shall only be allowed provided that the secret key is not an Assigned Key, that the secret key is encrypted, and that a secure channel (providing authentication and integrity protection) is used for the export*
2. *Public keys shall always be exported with integrity protection of their key value and attributes*
3. *Keys shall only be imported over a secure channel (providing authentication and integrity protection)*
4. *A secret key can only be imported if it is a non-Assigned key*
5. *Secret keys shall only be imported in encrypted form or using split-knowledge procedures requiring at least two key components to reconstruct the key, with key components supplied by at least two separately authenticated users*
6. *Unblocking access to a key shall not allow any subject other than those authorized to access the key at the time when it was blocked.*

FDP_IFF.1.3/KeyBasics The TSF shall enforce the following additional information flow control rules: *none*

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

FDP_IFF.1.5/KeyBasics The TSF shall explicitly deny an information flow based on the following rules:

1. *No subject shall be allowed to access the plaintext value of any secret key directly.*
2. *No subject shall be allowed to export a secret key in plaintext.*
3. *No subject shall be allowed to export an Assigned Key.*
4. *No subject shall be allowed to export a secret key without submitting the correct authorization data for the key*
5. *No subject shall be allowed to access intermediate values in any operation that uses a secret key*
6. *A key with an Export Flag value marking it as non-exportable shall not be exported*

Application Note 22

A secure channel for export of keys in FDP_IFF.1.2/KeyBasics (1) or for import of keys in FDP_IFF.1.2/KeyBasics (3) is one that meets the requirements of FTP_TRP.1/Local or FTP_TRP.1/External.

The encrypted form required for keys imported or exported over a secure channel requires encryption of the key itself, in addition to any encryption provided by the secure channel.

Unblocking a key as in FDP_IFF.1.2/KeyBasics (6) is intended only to restore the ability of subjects to authorize for access to a key by presenting the correct authorization data. As noted for FMT_MTD.1/Unblock, the subject who unblocks the key shall not be able also to use the key as a result of the unblocking (unless of course they are able to supply the correct authorization data). This is a part of ensuring that sole control of secret keys can be achieved.

Application Note 23

This SFR is supported by the following FCS_COP iterations:

- Encryption of keys for External Key Storage: FCS_COP.1/Sym_Enc_Dec with AES-256-GCM.
- Encryption of keys for Key Export/Import: FCS_COP.1/Sym_Enc_Dec with AES 128, 192, 256 bits, (ECB, CBC, CTR, GCM, KW, KWP) and FCS_COP.1/ASym_Enc_Dec (all algorithms and parameters)

FDP_ACC.1/KeyUsage *Subset access control*

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/KeyUsage The TSF shall enforce the *Key Usage SFP* on

1. *subjects: all*
2. *objects: keys*
3. *operations: all*

FDP_ACF.1/KeyUsage *Security attribute based access control*

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

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

1. *whether the subject is currently authorized to use the secret key*
2. *whether the subject is currently authorized to change the attributes of the secret key*
3. *the cryptographic function that is attempting to use the secret key.*

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

1. *Attributes of a key shall only be changed by an authorized subject, and only as permitted in the Key Attributes Modification Table*
2. *Only subjects with current authorization for a specific secret key shall be allowed to carry out operations using the plaintext value of that key*
3. *Only cryptographic functions permitted by the secret key's Key Usage attribute shall be carried out using the secret key*

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

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

Application Note 24

Whether a subject is currently authorized for access to a secret key is determined by whether the subject has submitted the correct authorization data for the key, and whether this authorization is yet subject to one or more of the re-authorization conditions in FIA_UAU.6/KeyAuth.

Whether a subject is currently authorized to change the attributes of a secret key is determined by the iterations of FMT_MSA.1

Application Note 25

FDP_ACF.1.2/KeyUsage (1) refers to controls over changing attributes that are specified in more detail in the iterations of FMT_MSA.1.

FDP_ACF.1.2/KeyUsage (2) requires that a key can only be used when the relevant subject has been authorized either by presenting the correct authorization data for the key as part of the request for the operation or else the authorization has previously been presented by the subject and the current use of the key does not yet require re-authorization according to FIA_UAU.6/KeyAuth (meaning that the current usage is therefore within the usage constraints for time and number of uses since the last authorization of use of the key). The reference to use of the plaintext value of the key does not imply that a subject has access to that value, only that it can be used to carry out operations within the TOE – reference to operations of this sort are thus distinguished from operations that may use an encrypted form of a secret key (e.g. for external storage of keys) and that are not necessarily restricted in this way.

Application Note 26

The requirements of FDP_ACF.1/KeyUsage apply regardless of how the key is stored by the TOE, including when the key is externally stored.

FDP_ACC.1/Backup *Subset access control*

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

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

1. *subjects: all*
2. *objects: keys*
3. *operations: backup, restore*

Application Note 27

This SFR is trivially met as no backup facility is provided by the TOE.

FDP_ACF.1/Backup *Security attribute based access control*

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

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

1. *whether the subject is an administrator*

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

1. *Only authorized administrators shall be able to perform any backup operation provided by the TSF to create backups of the TSF state or to restore the TSF state from a backup*
2. *Any restore of the TSF shall only be possible under at least dual person control, with each person being an administrator*
3. *Any backup and restore shall preserve the confidentiality and integrity of the secret keys, and the integrity of public keys*
4. *Any backup and restore operations shall preserve the integrity of the key attributes, and the binding of each set of attributes to its key*

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

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

Application Note 28

This SFR is trivially met as no backup facility is provided by the TOE.

FDP_SDI.2	*Stored data integrity monitoring and action*

Hierarchical to: FDP_SDI.1 Stored data integrity monitoring

Dependencies: No dependencies

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for *integrity errors* on *all* ***keys (including security attributes)***[72], based on the following attributes: *integrity protection data.*

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall

1. *prohibit the use of the altered data*
2. *notify the error to the user*

Application Note 29

This SFR is supported by FCS_COP.1/Digest_Main with SHA-256 algorithm.

FDP_RIP.1	**Subset residual information protection**

Hierarchical to: No other components

Dependencies: No dependencies

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

- *authorization data*
- *secret keys*

6.3.4 Trusted path/channels (FTP)

FTP_TRP.1/Local/Embedded	*Trusted Path*

Hierarchical to: No other components

Dependencies: No dependencies

FTP_TRP.1.1/Local/Embedded The TSF shall provide a communication path between itself and *local* **embedded FM applications**[73] that is logically distinct from other communication paths and provides assured **authentication**[74] of its end points and protection of the communicated data from *modification and disclosure*.

[72] objects
[73] Refinement: 'local client applications' is refined to 'local embedded FM applications' to distinguish them from local client applications stored within the appliance.
[74] identification

FTP_TRP.1.2/Local/Embedded The TSF shall permit local **embedded FM** applications[75] to initiate communication via the trusted path.

FTP_TRP.1.3/Local/Embedded The TSF shall require the use of the trusted path for all services provided by the TOE to the local embedded FM applications[76].

Application Note 30

Local embedded FM applications are located within the TOE physical boundary (i.e. inside the PCI-e card). As stated in application note 29 of the PP, the local trusted path is mapped to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment).

FTP_TRP.1/Local/Appliance *Trusted Path*

Hierarchical to: No other components

Dependencies: No dependencies

FTP_TRP.1.1/Local/Appliance The TSF shall provide a communication path between itself and *local **client applications***[77] that is logically distinct from other communication paths and provides assured **authentication**[78] of its end points and protection of the communicated data from *modification and disclosure*.

FTP_TRP.1.2/Local/Appliance The TSF shall permit local client applications[79] to initiate communication via the trusted path.

FTP_TRP.1.3/Local/Appliance The TSF shall require the use of the trusted path for all security-related services offered to local client applications stored within the appliance, including:

- Cryptographic operations
- Operations on keys
- Authentication operations[80].

Application Note 31

Local client applications are located within the physical boundary of the same hardware appliance (either the generic server in which the TOE is inserted, or the Thales Luna Network HSM). Therefore, as stated in application note 29 of the PP, the local trusted path is mapped to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment).

[75] [Selection: the TSF, local client applications]. Moreover 'local client applications' is refined to 'local embedded FM applications' to distinguish them from local client applications stored within the appliance.
[76] [assignment: services for which trusted path is required]
[77] users
[78] identification
[79] [selection: the TSF, local client applications]
[80] [assignment: services for which trusted path is required]

FTP_TRP.1/External *Trusted Path*

Hierarchical to: No other components

Dependencies: No dependencies

FTP_TRP.1.1/External The TSF shall provide a communication path between itself and *remote **external client applications***[81] that is logically distinct from other communication paths and provides assured **authentication**[82] of its end points and protection of the communicated data from *modification and disclosure*.

FTP_TRP.1.2/External The TSF shall permit remote external client applications[83] to initiate communication via the trusted path.

FTP_TRP.1.3/External The TSF shall require the use of the trusted path for all security-related services offered to remote external client applications, including:

- Cryptographic operations
- Operations on keys
- Authentication operations[84]

Application Note 32

This SFR is supported by the following FCS_COP iterations:

FCS_CKM.1 (ECDH P-521)

FCS_COP.1/Sym_Enc_Dec (AES-256 GCM or CTR)

FCS_COP.1./MAC (HMAC-SHA-512)

6.3.5 Protection of the TSF (FPT)

FPT_STM.1 *Reliable time stamps*

Hierarchical to: No other components

Dependencies: No dependencies

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

Application Note 33

The TOE must provide timestamps suitable for supporting the time in an audit record for FAU_GEN.1.

[81] users
[82] identification
[83] [selection: the TSF, remote external client applications]
[84] [assignment: services for which trusted path is required]

FPT_TST_EXT.1 *Basic TSF Self Testing*

Hierarchical to: No other components

Dependencies: No dependencies

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests *during initial start-up (or power-on),* periodically during normal operation and on demand[85] to demonstrate the correct operation of the TSF:

> *At initial start-up (or power-on):*

- *Software/firmware integrity test*
- *Cryptographic algorithm tests*
- *Random number generator tests*

> Periodically during normal operation: Random number generator tests

> On demand:

- Cryptographic algorithm tests
- Software/firmware integrity test
- Random number generator tests][86].

FPT_PHP.1 Passive detection of physical attack

Hierarchical to: No other components

Dependencies: No dependencies

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.

FPT_PHP.3/K7 Resistance to physical attack

Hierarchical to: No other components

Dependencies: No dependencies

FPT_PHP.3.1/K7 The TSF shall resist the physical tampering scenarios listed in Table 6-10[87] to the TSF elements listed in Table 6-10 [88] by responding automatically such that the SFRs are always enforced.

[85] [selection: periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self-tests should occur]]
[86] [assignment: list of additional self-tests run by the TSF]
[87] [assignment: physical tampering scenarios]
[88] [assignment: list of TSF devices/elements]

Table 6-10: List of physical tampering scenarios for K7 hardware variants

Physical tampering scenario	*TSF element*
Modification of the voltage and temperature outside the normal operating range	*PCB*

Application Note 34

This SFR applies specifically to the K7 hardware variants whose PCB is protected by a passive epoxy coating shield: 808-000048-002, 808-000073-001 and 808-000066-001.

FPT_PHP.3/K7+ *Resistance to physical attack*

Hierarchical to: No other components

Dependencies: No dependencies

FPT_PHP.3.1/K7+ The TSF shall resist the physical tampering scenarios listed in Table 6-11[89] to the TSF elements listed in Table 6-11 [90] by responding automatically such that the SFRs are always enforced.

Table 6-11: List of physical tampering scenarios for K7+ hardware variants

Physical tampering scenario	*TSF element*
Modification of the voltage and temperature outside the normal operating range	*PCB*
Unauthorized physical access to the PCB components protected by the active shield. In that scenario, the objective of the attacker is to perform direct physical probing on the security chip(s) to get unauthorized access to internal signals.	*All components under the active shield*
Unauthorized physical access to the PCB components protected by the active shield. In that scenario, the objective of the attacker is to modify the underlying circuitry in order to cause TSF malfunctions.	*All components under the active shield*

Application Note 35

This SFR applies specifically to the two K7+ hardware variants whose PCB is protected by an active shield: 808-000069-001 and 808-000070-001.

FPT_FLS.1 *Failure with preservation of secure state*

Hierarchical to: No other components

Dependencies: No dependencies

[89] [assignment: physical tampering scenarios]
[90] [assignment: list of TSF devices/elements]

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

1. *Self-test according to FPT_TST_EXT.1 fails*
2. *Environmental conditions are outside normal operating range (including temperature and power)*
3. *Failures of critical TOE hardware components (including the RNG) occur*
4. *Corruption of TOE software occurs*
5. None[91]*.*

6.3.6 Security management (FMT)

For the purposes of specifying a minimum set security attributes of keys, and the constraints on initialisation and modification of these attributes in FMT_MSA.1 and FMT_MSA.3, two separate types of keys are defined: Assigned Keys (defined and recognized by having their 'Assigned Flag' attribute set to 'assigned'), and general keys (keys that have their 'Assigned Flag' attribute set to 'non-assigned').

Assigned Keys represent a type of key that can be more easily mapped to requirements for sole control because changes to some of their attributes are more tightly controlled (see FMT_MSA.1/AKeys, and the description of attributes below) and, since they are intended for use within the TOE, because they cannot be imported or exported[92]. In particular, an Administrator cannot avoid the need to provide the current authorization data in order to use such a key, nor can an Administrator change the authorization data (which would then allow use of the key by the Administrator). This enables a key to be generated and then to be made an Assigned Key at the point where it is assigned to an individual signatory or, in the case of a key used for the creation of electronic seals, to a group of key users[93].

In the FMT_MSA SFRs specified for keys below, the permitted values of assignments have been restricted to identify a minimum set of attributes that shall be mapped to their implementation in a TOE, and to specify a minimum set of constraints on their initialization and subsequent modification. Additional notes regarding these attributes are as follows:

- Key identifier: this must be sufficient to uniquely identify the key within the system of which the TOE is a part
- Key type: this identifies at a minimum whether the key is a secret key of a symmetric cryptographic algorithm or the secret (commonly called private) key of an asymmetric cryptographic algorithm
- Authorization data: value of data that allows the key to be used for cryptographic operations according to the rules in other SFRs such as FDP_IFF.1/KeyBasics, FDP_ACF.1/KeyUsage, and FDP_ACF.1/Backup. Authorization data is required only for secret keys
- Re-authorization conditions: the constraints on uses of the key that can be made before re-authorization is required according to FIA_UAU.6/KeyAuth, and which determines whether a subject is currently authorized to use a key as in FDP_ACF.1/KeyUsage. The types of secret key to which re-authorization

[91] [assignment: list of other types of failures in the TSF]

[92] Assigned Keys may be stored externally in a form that protects the confidentiality and integrity of the key and the binding of the key to its attributes (in particular the requirements of the SFRs FDP_IFF.1/KeyBasics and FDP_SDI.1 apply to externally stored keys), as discussed in section 1.

[93] Secure operating procedures will be needed in order to ensure that the process from generation to assignment is suitable for maintaining any requirements for non-repudiation that may apply to the application context for use of the key (cf. OE.DataContext and the refinement to AGD_OPE.1 in section 6.4.1).

conditions apply, and the details of the re-authorization conditions for a specific TOE are described in FIA_UAU.6/KeyAuth in section 6.3.2

- Key usage: the cryptographic functions that are allowed to use the key as in FDP_ACF.1/KeyUsage
- Export flag: indicates whether the key is allowed to be exported (cf. FDP_IFF.1/KeyBasics); allowed values are referred to in the PP as 'true' (meaning export is allowed) and 'false' (meaning export is not allowed) but may be mapped to other suitable binary values in TOE implementations
- Assigned flag: indicates whether the key has currently been assigned. Once a key has been assigned by an Administrator then its authorization data can only be changed on successful validation of the current authorization data – it cannot be changed or reset by an Administrator – and the re-authorization conditions and key usage attributes cannot be changed; allowed values are referred to in the PP as 'assigned' and 'non-assigned' but may be mapped to other suitable binary values in TOE implementations.

FMT_SMR.1 *Security roles*

Hierarchical to: No other components

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles **HSM SO, Partition SO, Administrator, Audit User, Partition CO**[94], **STC User**[95]*,* **Key Owner**[96]*,* Partition Limited CO, Partition CU[97].

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

Application Note 36

All the roles mentioned in this SFR are described in section 1.4.5.

FMT_SMF.1 *Security management functions*

Hierarchical to: No other components

Dependencies: No dependencies

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

1. *Unblock of access due to authentication or authorization failures*
2. *Modifying attributes of keys*

[94] Refinement: HSM SO, Partition SO, Administrator, Audit User and Partition CO replace the generic wording 'Administrator' mentioned in the protection profile. These are the actual HSM administrative roles, as described in section 1.4.5.
[95] Refinement: as described in section 1.4.5, STC_User is the implicit role used to authenticate local and external client applications, both of which are supported by the TOE. Note that authentication through the STC is mandatory for external client applications, and optional for local client applications, as mentioned in section 7.5 "Trusted path".
[96] Refinement: as described in section 1.4.5, Key Owner is the implicit role that corresponds to the Key User role mentioned in the protection profile.
[97] [assignment: list of additional authorized identified roles]

3. *Export and deletion of the audit data, which can take place only under the control of the* **Audit User**[98] *role*
4. *No backup and restore functions*[99]
5. *Key import function*[100]
6. *Key export function*[101]
7. **Firmware updates**[102]
8. **Loading of embedded FM applications**[103]

FMT_MTD.1/Unblock	*Management of TSF data*

Hierarchical to:	No other components
Dependencies:	FMT_SMR.1 Security roles
	FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/Unblock The TSF shall restrict the ability to *unblock* the TSF data listed in table 6-12[104] to the administrators listed in Table 6-12[105].

Table 6-12: Unblocking of TSF data by administrators

Blocked TSF data	Administrator which can unblock the TSF data
Administrator account	HSM SO provided that the module policy "Partition SO can reset PIN" is enabled. Otherwise, there is no unblocking capability.
Partition CO account	Partition SO provided that the module policy "Partition SO can reset PIN" is enabled. Otherwise, there is no unblocking capability.
Partition Limited CO account	Partition CO
Partition CU account	Partition CO
Any User Key (whether General or Assigned)	Case of a User Partition: Partition CO Case of the Admin Partition: HSM SO

98 Refinement of the 'Administrator' initial wording, as Audit User is the only HSM administrative role to control the export and deletion of audit data.
99 [selection: backup and restore functions, no backup and restore functions]
100 [selection: key import function , no key import function]
101 [selection: key export function , no key export function]
102 Refinement to the PP (to add the firmware update capability)
103 Refinement to the PP (to add the FM loading capability)
104 [assignment: list of TSF data]
105 [assignment: the authorized identified administrative roles]

Application Note 37

*There is a distinction between administrators authorised to unblock a key and users authorised to use the key. When unblocking a secret key, the unblocking process shall not allow a subject to use the key other than a subject who is authorised by presentation of the current authorisation data. For example, an administrator who is able to unblock the key cannot then **use** the key as a result of the unblocking (so the unblocking process does not itself allow the key to be used, nor does it enable the authorisation data to be changed without proving knowledge of the previous authorisation data). This is a part of ensuring that sole control of secret keys can be achieved.*

FMT_MTD.1/AuditLog *Management of TSF data*

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/AuditLog The TSF shall restrict the ability to *control export and deletion of* the *audit log records* to the **Audit User**[106] *role*.

Application Note 38

The control of export and deletion of the audit log records helps to ensure their protection against accidental or malicious deletion (deletion should normally occur only after the records have been exported and preserved outside the TOE). Note that this does not require the Audit User to carry out these export or delete operations manually as long as the actions are controlled by the Audit User.

FMT_MTD.1/FW_update *Management of TSF data*

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/FW_update The TSF shall restrict the ability to update[107] the HSM firmware[108] to the HSM SO[109] role.

Application Note 39

This SFR iteration is an addition to [CEN EN 419221-5] to support the firmware update capability mentioned as a refinement in FMT_SMF.1.

[106] Refinement of the 'Administrator' initial wording, as Audit User is the only HSM administrative role to control the export and deletion of audit data.
[107] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[108] [assignment: list of TSF data]
[109] [assignment: the authorized identified roles].

FMT_MTD.1/FM_loading *Management of TSF data*

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/FM_loading The TSF shall restrict the ability to load[110] embedded FM applications[111] to the HSM SO[112] role.

Application Note 40

This SFR iteration is an addition to *[CEN EN 419221-5]* to *support the FM loading capability mentioned as a refinement in FMT_SMF.1.*

FMT_MSA.1/GenKeys *Management of security attributes (General Keys)*

Hierarchical to: No other components

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/GenKeys The TSF shall enforce the *Key Usage SFP* to restrict the ability to *modify* the security attributes listed in Table 6-13[113] to the conditions and actors specified in Table 6-13[114].

Table 6-13: Key Attributes Modification Table, instantiated for K7 General Keys

Key attribute	***Mapping to PP category / naming***	***Modification conditions***
CKA_OUID	*Key ID*	*Cannot be modified*
CKA_CLASS	*Key type*	*Cannot be modified*
CKA_KEY_TYPE	*Key type*	*Cannot be modified*

[110] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[111] [assignment: list of TSF data]
[112] [assignment: the authorized identified roles].
[113] [assignment: list of security attributes, to include attributes as specified in the Key Attributes Modification Table]
[114] [assignment: list of subjects, objects, and operations among subjects and General Keys, to include at least the constraints specified in the Key Attributes Modification Table]

Key attribute	***Mapping to PP category / naming***	***Modification conditions***
CKA_AUTH_DATA	*Authorization Data*	*Case of a User Partition: the attribute can be modified by Key Owner only when modification operation includes successful validation of current (pre-modification) authorization data, or by Partition CO.* *Case of the Admin Partition: the attribute can be modified by Key Owner only when modification operation includes successful validation of current (pre-modification) authorization data, or by the HSM SO.*
CKA_ENCRYPT	*Key Usage*	*Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).* *Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).*
CKA_DECRYPT	*Key Usage*	*Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).* *Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).*
CKA_WRAP	*Key Usage*	*Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).* *Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).*

Key attribute	***Mapping to PP category / naming***	***Modification conditions***
CKA_UNWRAP	*Key Usage*	*Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).* *Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).*
CKA_SIGN	*Key Usage*	*Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).* *Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).*
CKA_VERIFY	*Key Usage*	*Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).* *Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).*
CKA_DERIVE	*Key Usage*	*Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).* *Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).*

Key attribute	***Mapping to PP category / naming***	***Modification conditions***
CKA_EXTRACTABLE	*Export Flag*	*Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).* *Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).*
CKA_ASSIGNED	*Assigned Flag*	*Case of a User Partition: the attribute can be modified only by Partition CO, and only to change from non-assigned to assigned* *Case of the Admin Partition: the attribute can be modified only by the HSM SO, and only to change from non-assigned to assigned*
CKA_MODIFIABLE	-	*Case of a User Partition: The attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well, and only to change from modifiable to non-modifiable.* *Case of the Admin Partition: The attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well, and only to change from modifiable to non-modifiable.*
CKA_FAILED_KEY_ AUTH_COUNT	-	*Case of a User Partition: the attribute can be modified only by Partition CO* *Case of the Admin Partition: the attribute can be modified only by the HSM SO or by the Administrator.*

Application Note 41

'Integrity protection data' is maintained automatically by the TSF on key objects (as stated in FDP_SDI.2). However, it is not implemented as a security attribute, which is the reason why it is not listed in table 6-12.

The same applies to 're-authorization conditions'. According to FIA_UAU.6, re-authorization is required after explicit rescinding of previous authorization. As there is no other option, no dedicated security attribute is needed here.

FMT_MSA.1/AKeys	*Management of security attributes (Assigned Keys)*

Hierarchical to: No other components

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/AKeys The TSF shall enforce the *Key Usage SFP* to restrict the ability to *modify* the security attributes listed in Table 6-14[115] to the conditions and actors specified in Table 6-14[116].

Table 6-14: Key Attributes Modification Table, instantiated for K7 Assigned Keys

Key attribute	***Mapping to PP category / naming***	***Modification conditions***
CKA_OUID	*Key ID*	*Cannot be modified*
CKA_CLASS	*Key type*	*Cannot be modified*
CKA_KEY_TYPE	*Key type*	*Cannot be modified*
CKA_AUTH_DATA	*Authorization Data*	*Can be modified by Key Owner only when modification operation includes successful validation of current (pre-modification) authorization data.*
CKA_ENCRYPT	*Key Usage*	*Cannot be modified*
CKA_DECRYPT	*Key Usage*	*Cannot be modified*
CKA_WRAP	*Key Usage*	*Cannot be modified*
CKA_UNWRAP	*Key Usage*	*Cannot be modified*
CKA_SIGN	*Key Usage*	*Cannot be modified*
CKA_VERIFY	*Key Usage*	*Cannot be modified*
CKA_DERIVE	*Key Usage*	*Cannot be modified*

[115] [assignment: list of security attributes, to include attributes as specified in the Key Attributes Modification Table]
[116] [assignment: list of subjects, objects, and operations among subjects and Assigned Keys to include at least the constraints specified in the Key Attributes Modification Table]

Key attribute	***Mapping to PP category / naming***	***Modification conditions***
CKA_EXTRACTABLE	*Export Flag*	*Cannot be modified*
CKA_ASSIGNED	*Assigned Flag*	*Cannot be modified*
CKA_MODIFIABLE	-	*Cannot be modified*
CKA_FAILED_KEY_ AUTH_COUNT	-	*Case of a User Partition: the attribute can be modified only by Partition CO.* *Case of the Admin Partition: the attribute can be modified only by the HSM SO.*

Application Note 42

- 'Integrity protection data' is maintained automatically by the TSF on key objects (as stated in FDP_SDI.2). However, it is not implemented as a security attribute, which is the reason why it is not listed in table 6-13.
- The same applies to 're-authorization conditions'. According to FIA_UAU.6, re-authorization is required after explicit rescinding of previous authorization. As there is no other option, no dedicated security attribute is needed here.

FMT_MSA.3/Keys *Static attribute initialisation*

Hierarchical to: No other components

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1/Keys The TSF shall enforce the *Key Usage SFP* to provide restrictive[117] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Keys The TSF shall allow the authorized identified roles, according to Table 6-15[118] to specify alternative initial values to override the default values when an object or information is created.

[117] [selection, choose one of: restrictive, permissive, [assignment: other property]]
[118] [assignment: the authorized identified roles, according to the constraints in the Key Attributes Initialisation Table]

Table 6-15: Key Attributes Initialisation Table, instantiated for K7 Keys

Key Attribute	*Initialization rules (Assigned Key)*	*Initialization rules (General Key)*
CKA_OUID	*Initialized by generation process.*	
CKA_CLASS	*Initialized by generation process.*	
CKA_KEY_TYPE	*Initialized by generation process.*	
CKA_AUTH_DATA	*Case of a User Partition: Initialized by Key Owner or by Partition CO during generation (no default value).* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation (no default value).*	
CKA_ENCRYPT	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.*	
CKA_DECRYPT	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.*	
CKA_WRAP	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.*	
CKA_UNWRAP	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.*	
CKA_SIGN	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.*	
CKA_VERIFY	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.*	
CKA_DERIVE	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.*	

Key Attribute	*Initialization rules (Assigned Key)*	*Initialization rules (General Key)*
CKA_EXTRACTABLE	*Set to false.*	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.*
CKA_ASSIGNED	*Set to true.*	*Set to false.*
CKA_MODIFIABLE	*Set to false.*	*Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is true.* *Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is true.*
CKA_FAILED_KEY_ AUTH_COUNT	*Set to 0.*	*Set to 0.*

Application Note 43

Regarding key import, The TOE will assign the default values specified above to imported general keys, for the attributes that are missing from the imported key (if any).

6.3.7 Security audit data generation (FAU)

FAU_GEN.1 *Audit data generation*

Hierarchical to: No other components

Dependencies: FPT_STM.1 Reliable time stamps

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 *not specified* level of audit; and
- **c.** *Startup of the TOE;*
- **d.** *Shutdown of the TOE*
- **e.** *Cryptographic key generation (FCS_CKM.1);*
- **f.** *Cryptographic key destruction (FCS_CKM.4);*

g. *Failure of the random number generator (FCS_RNG.1);*

h. *Authentication and authorization failure handling (FIA_AFL.1): all unsuccessful authentication or authorization attempts, the reaching of the threshold for the unsuccessful authentication or authorization attempts and the blocking actions taken;*

i. *All attempts to import or export keys (FDP_IFF.1/KeyBasics);*

j. *All modifications to attributes of keys (FDP_ACF.1/KeyUsage, FMT_MSA.1/GenKeys and* FMT_MSA.1/AKeys*);*

k. *Backup and restore (FDP_ACF.1/Backup): use of any backup function, use of any restore function, unsuccessful restore because of detection of modification of the backup data;*

l. *Integrity errors detected for keys (FDP_SDI.2);*

m. *Failures to establish secure channels (FTP_TRP.1/Local, FTP_TRP.1/External);*

n. *Self-test completion (FPT_TST_EXT.1);*

o. *Failures detected by the TOE (FPT_FLS.1);*

p. *All administrative actions (FMT_SMF.1, FMT_MSA.1 (all iterations), FMT_MSA.3/Keys,);*

q. *Unblocking of access (FMT_MTD.1/Unblock);*

r. *Modifications to audit parameters (affecting the content of the audit log) (FAU_GEN.1)*

s. None[119].

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: None[120].

FAU_GEN.2 User identity association

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_STG.2 Guarantees of audit data availability

Hierarchical to: FAU_STG.1 Protected audit trail storage

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.2.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

[119] [Assignment: other specifically defined auditable events]
[120] [Assignment: other audit relevant information]

FAU_STG.2.2 The TSF shall be able to detect[121] unauthorized modifications to the stored audit records in the audit trail.

FAU_STG.2.3 The TSF shall ensure that *all* stored audit records will be maintained when the following conditions occur: *audit storage exhaustion.*

[121] [selection, choose one of: prevent, detect]

6.4 TOE Security Assurance Requirements

The security assurance requirement level is **EAL4** augmented with **ALC_FLR.2 and AVA_VAN.5.** The assurance components are identified in the table below (with augmentations in bold). It is noted that due to the physically protected environment in which the TOE operates (as expressed in OE.Env), the scope for physical attacks is limited.

Table 6-16: Security Assurance Requirements

Assurance Class	Assurance Components
Security Target (ASE)	ST introduction (ASE_INT.1)
	Conformance claims (ASE_CCL.1)
	Security problem definition (ASE_SPD.1)
	Security objectives (ASE_OBJ.2)
	Extended components definition (ASE_ECD.1)
	Derived security requirements (ASE_REQ.2)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Security architecture description (ADV_ARC.1)
	Complete functional specification (ADV_FSP.4)
	Basic modular design (ADV_TDS.3)
	Implementation representation of the TSF (ADV_IMP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Production support, acceptance procedures and automation (ALC_CMC.4)
	Problem tracking CM coverage (ALC_CMS.4)
	Delivery procedures (ALC_DEL.1)
	Identification of security measures (ALC_DVS.1)
	Developer defined life-cycle model (ALC_LCD.1)
	Well-defined development tools (ALC_TAT.1)
	Flaw Reporting Procedures (ALC_FLR.2)
Tests (ATE)	Functional testing (ATE_FUN.1)
	Analysis of coverage (ATE_COV.2)

	Testing: basic design (ATE_DPT.1)
	Independent testing – sample (ATE_IND.2)
Vulnerability assessment (AVA)	**Advanced methodical vulnerability analysis (AVA_VAN.5)**

6.4.1 Refinements of Security Assurance Requirements

The following refinements are made to selected assurance requirements in Table 6-16:

ADV_ARC.1 *Security architecture description*

Refinement:

The following specific topics shall be addressed as part of ADV_ARC.1 for this Protection Profile. It is acceptable for references to deliverables supplied for other assurance families, such as ADV_FSP, to be used to meet these requirements, provided that the relationship of the relevant interface specifications to the concepts in the Protection Profile is clear. Note that in some cases, the requirement for description of these particular aspects under ADV_ARC is intended to make clear any differences between the full capabilities of the product and the scope of the Security Target.

1. In general cryptographic modules will make use of 'support keys' as part of their implementation of protection mechanisms, where these keys are generally not held on behalf of specific users[122] or client applications, but are used by the TOE to carry out its normal operations and as part of the implementation mechanism other SFRs and to protect the TSF itself. These support keys may be used for a variety of purposes (including aspects such as authentication, authorization, secure channels, security of external storage, or internal data protection), For the purposes of this PP, support keys used by the TOE are treated as TSF data, and require a specific security rationale to be included as part of the ADV_ARC.1 deliverables. This rationale shall include a description of the key architecture, identifying all support keys used by the TOE (at least in its evaluated configuration), their method of generation and storage, their purpose in TOE operation, and the ways in which they are protected so as to support the requirements of FDP_IFF.1/KeyBasics and FDP_ACF.1/KeyUsage (noting that the mechanisms used for support keys may differ from those used for user keys). Examples would be keys used for wrapping user keys in order to allow secure storage of the user keys, keys used to implement secure channels, and keys used to protect backups. The description shall demonstrate that sufficient entropy has been used in the generation of each support key, and the source of that entropy. The rationale shall demonstrate that these support keys cannot be exported/imported in a way that threatens the secure operation of the TOE. The evaluator shall include the description of the support keys in their analysis of the protection of user data (e.g. to confirm that it does not introduce vulnerabilities in the implementation of the SFRs).
2. If updates to the TOE software or firmware are supported then the ADV_ARC.1 deliverables must describe how the TOE is protected against unauthorized updates, by using digital signatures. This shall be confirmed by evaluator testing (if updates are supported) to confirm that updates with invalid signatures are rejected without being executed. The digital signature algorithms used to protect updates shall be included in the scope of FCS_COP.1 signature SFR(s).

[122] Some support keys may be seen as being held on behalf of administrators, but the main intention of distinguishing support keys and user keys is for the ADV_ARC.1 deliverables to describe all the different types of key available, their properties, and their relationship to the SFRs in this Protection Profile.

3. The ADV_ARC.1 deliverables shall in particular describe:
    a. Any use that the TOE makes of an audit server
    b. The locations used for any externally stored keys and the structure and format of the externally stored keys including the cryptographic structures that protect the keys in their externally stored form, and that bind them to their attributes (support keys are separately addressed by the description required in item 1 above)
    c. All key import and/or export functions and the secure channels that they use
    d. The secure channels supported by the TOE and the authentication mechanisms that they use (cf. FTP_TRP.1/Local & FTP_TRP.1/External)
    e. All local and external interfaces used for communications with users, client applications, audit data, and stored TOE data (cf. Figure 1-3)
    f. The specific key attributes supported, their method of representation (e.g. the relevant data structures and permitted values) and the method by which they are bound to the corresponding key value (cf. FMT_MSA.1). This also includes identifying the types of keys (if any) that support re-authorization conditions described in FIA_UAU.6/KeyAuth
    g. The user types and roles supported, the interfaces by which they interact with the TOE (e.g. a local administrator console or an externally available API), the authentication methods used (cf. FIA_UAU.1), and any privileges available to the user type/role
    h. All of the cryptographic functions provided and whether any non-endorsed cryptographic algorithms and/or cryptographic functions are available (cf. FCS_COP.1)
    i. The authorization methods used for keys (cf. FIA_UAU.6/KeyAuth & FDP_ACC.1/KeyUsage)
    j. Description of the way in which the TOE ensures that it only holds authorization data for the minimum time possible before deallocating it according to FDP_RIP.1
    k. If the TOE provides backup operations then the ADV_ARC deliverables shall describe the use of support keys by the backup and restore processes (cf. FDP_ACF.1/Backup), and in particular shall describe the ways in which confidentiality and integrity of the backup are provided, and the way in which the TOE rejects an attempt to carry out a restore process using backup data that has been modified
    l. Any mechanisms that the TOE uses to support dual person control (cf. FDP_ACF.1/Backup).

AGD_OPE.1 *Operational user guidance*

Refinement:

The following specific topics shall be addressed as part of the Operational Guidance for the TOE:

1. The specific ways in which the TOE needs to be configured and used in order to provide qualified electronic signatures and qualified electronic seals that meet the requirements of [Regulation]. This includes ways in which the TOE can ensure that the signatory can, with high level of confidence, have sole control over the use of the secret key that acts as his/her signature creation data. Thus, for example, it may be necessary for client applications to use TOE interfaces according to certain guidance in order to correctly implement the requirements on attributes of keys as described in this ST. It may be necessary for the TOE to define ways in which secret keys to be used for signing purposes can be created in a way that does not allow subsequent modification of some or all of their attributes, e.g. by an administrator, before they are assigned to the signatory (cf. FMT_MSA.1/AKeys). The intention of this aspect of the operational user guidance documentation is

to identify the configuration and secure use required for a particular TOE, and how it is necessary to connect this with other aspects such as procedural controls and client applications in the operational environment.

The evaluators shall test the identified ways of using the TOE for qualified electronic signatures and qualified electronic seals to demonstrate that the description in the Operational Guidance is suitably complete, and that the keys produced by following the Operational Guidance do indeed meet the requirements of requirements of [Regulation, Annex II & Annex III] for qualified electronic signatures and qualified electronic seals.

2. The use of trusted channels (cf. FTP_TRP.1/Local & FTP_TRP.1/External).
3. The available key attributes, their possible values, and the meaning of each of these values (cf. FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys, including their use to constrain the period and number of uses that are enabled by authorization of a key (cf. FIA_UAU.6/KeyAuth).
4. Identification of any non-endorsed cryptographic algorithms and/or cryptographic functions that are available (cf. FCS_COP.1).
5. Identification of any other cryptographic algorithms and operations that are not included in the scope of the Security Target.
6. Possible errors from the self-test process and the actions that should be taken in response to each (cf. FPT_TST_EXT.1).
7. Specific failures detected by the TOE (cf. FPT_FLS.1).
8. Audit functions and their configuration (including specification of the available audit records), along with any other actions that are associated with audit functions (e.g. archiving or viewing audit records, or use of an external audit server) (cf. FAU_GEN.1, FAU_STG.2, FMT_MTD.1/AuditLog).
9. Any configuration and operation requirements for dual-control operations (cf. FDP_ACF.1/Backup).
10. If backup is provided by the TOE (cf. FDP_ACF.1/Backup), then the Operational Guidance shall describe the backup and restore functions, and the administrator roles that are required to carry them out.
11. If key import is provided by the TOE, then the Operational Guidance shall describe how attributes are defined for any imported keys (cf. FMT_MSA.3/Keys). The evaluators shall test the import process to demonstrate that the description in the Operational Guidance is suitably complete, and that the keys imported have attributes appropriately defined. Similarly if key export is provided by the TOE then the Operational Guidance shall describe whether attributes are exported with keys (and if so, then how the attributes are represented and associated with the exported key), and the evaluators shall test the export process to demonstrate that the description in the Operational Guidance is suitably complete, and that the handling of attributes is as described.

ATE_IND.2 *Independent testing – sample*

Refinement:

The following specific topics shall be addressed as part of the independent testing of the TOE:

1. The evaluator shall execute the electronic signature and electronic seal operations provided by the TOE and shall confirm that the signatures and seals returned by the TOE correspond to the correct DTBS.
2. If software and/or firmware updates are supported by the TOE then the evaluator shall carry out tests to ensure that only updates with valid digital signatures can be installed on the TOE.

AVA_VAN.5 *Advanced methodical vulnerability analysis*

Refinement:

Regarding the protection of the TOE against physical attacks: because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 and FPT_PHP.3 for this TOE is equivalent to the level of assessment in section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 [ISO 19790] for Security Level 3.

7 TOE Summary Specification

7.1 Initialization, partitions, sessions and roles

7.1.1 Initialization

Before the module can be used to perform any cryptographic or key management functions, it must first be initialized. Initialisation causes the HSM to be zeroized[123] and creates the HSM Security Officer role (HSM SO) in the Admin Partition. The HSM SO must then set the configurable policies at the cryptographic module level and can create User Partitions with associated roles, to make the cryptographic module ready for use.

7.1.2 Admin and User Partitions

Cryptographic keys are stored and managed inside containers called partitions. The TOE supports two types of partitions:

> The Admin partition is mandatory. Its primary purpose is to support administrative tasks at the HSM level. Specific roles are defined in the Admin partition and there can be only one Admin partition inside the TOE.

> User partitions are optional. The primary purpose of a User partition (also called Application partition) is to group all keys belonging to a same entity or application in a dedicated container isolated from the other partitions. Specific roles are supported at the user partition level, which are different from the roles defined in the Admin partition.

Any type of partition (Admin partition or User partition) can contain cryptographic keys. For a given partition, the management and usage of the related key material is restricted to the roles assigned to that partition, therefore enforcing a strict isolation between the different partitions managed inside the TOE.

7.1.3 Sessions

Any user must access the module through a session. Sessions are initially opened as Public sessions and may remain Public or become Private (authenticated) following a successful user authentication. Session states are kept separate based on the user authentication state stored by the module. The module allows multiple user identities to be authenticated at a time. Once authenticated, a session becomes bound to the user identity and has access to all cryptographic operations appropriate to the user's role and may access private objects generated on behalf of the user in previous sessions. Although there may be many users authenticated to the cryptographic module, there is effectively only one thread of execution within the module and, therefore, only one command being executed from request through to response at any given time.

[123] 'HSM zeroisation' means that all user information and user key material are erased. The bootloader, firmware and HSM own data (HSM keys and certificates) are not erased.

7.1.4 Roles

The following roles are explicitly or implicitly enforced by the module:

Figure 7-1: Module roles

Role	Description
HSM Security Officer (HSM SO)	– Admin partition role – Authorized to install and configure the TOE, set and maintain global HSM security policies, create and delete application partitions. The HSM SO can also create the Administrator role and reset the Administrator password (configuration dependent). The HSM SO can also perform key management tasks and cryptographic operations for the Admin partition. Note that the HSM SO can only use key objects if he/she can also successfully authenticate as the Key Owner. The TOE can have only one HSM SO.
Administrator	– Admin partition role – Optional Admin partition role, performs key management tasks and cryptographic operations for the Admin partition, without having privileges to unblock keys / assign keys / reset key authentication data. Note that the Administrator can only use key objects if he/she can also successfully authenticate as the Key Owner.
Audit User	– Admin partition role – Initializes the Audit container containing the secret key used to generate Message Authentication Code (MAC) for secure audit messages alongside configuring logging levels for the HSM.
Application Partition Security Officer (Partition SO)	– User partition role – Creates partition level roles, activates partition, sets and changes partition-level policies, option to reset Crypto Officer password (configuration dependent).
STC User	Implicit role whose main purpose is to authenticate remote client connections from outside the TOE hardware appliance boundary[124].
Application Partition Crypto Officer (Partition CO)	– User partition role – Partition role authorized to create, use, destroy and transfer key objects for a given partition. Can optionally create the Partition Limited CO and Partition CU, and perform initial assignment of key authorization data. Note that the Partition CO can only use key objects if he/she can also successfully authenticate as the Key Owner.

[124] Usage of the STC is mandatory to authenticate remote client applications. Note that local client applications can be authenticated through the STC as well, although this is not mandatory - as explained in section 7.5 “Trusted Channel”.

Role	Description
Application Partition Limited Crypto Officer (Partition Limited CO)	– User partition role – Optional partition role authorized to create, use and delete key objects, and perform initial assignment of key authorization data without having privileges to create Partition Limited CO or CU / reset login credentials. Note that the Partition Limited CO can only use and delete key objects if he/she can also successfully authenticate as the Key Owner.
Crypto User (Partition CU)	– User partition role – Partition role authorized to use the key objects within the partition (e.g., sign, encrypt/decrypt). Note that the Partition CU can only use key objects if he/she can also successfully authenticate as the Key Owner.
Key Owner	Implicit role used to authenticate the owner of a key through verification of the related key authorization data.

HSM SO, Administrator, Partition SO, Audit User, Partition CO, Partition Limited CO and Partition CU are explicit roles supported by the HSM. These roles are enforced at the session level.

STC User is an implicit role used to authenticate local and external client applications, both of which are supported by the TOE. Note that authentication through the STC is mandatory for external client applications, and optional for local client applications, as mentioned in section 7.5 “Trusted path”. When the STC is used, the prior authentication of the client application and the subsequent establishment of the STC secure channel are mandatory steps before the application can send cryptographic operations or key management commands to the module.

Key Owner is also an implicit role used to authenticate the owner of a key (through verification of the related key authorization data).

The following table indicates (at a high-level) which activities can be performed by each role. It also provides a link to the subjects defined in the Protection Profile.

Figure 7-2: Module roles mapped to function.

Function		HSM Role(s)	Related PP subject
Initialisation, configuration		HSM SO, Partition SO	S.Admin
Audit Log Configuration		Audit User	S.Admin
Key Management		Partition CO (for User partitions) HSM SO, Administrator (for Admin partition)	S.Admin
Client Application Authentication		STC User	S.Application
Application Key Use	Session authentication	Partition CO, Partition Limited CO, Partition CU (for User partitions, and depending on configuration of TOE and deployment goals) HSM SO, Administrator (for Admin partition)	S.User
	Key usage	Key Owner	

Note: as defined in [CEN EN 419221-5], the subject S.Application represents *"a client application, or process acting on behalf of a client application and that communicates with the TOE over a local or external interface".* In the above table, it can be directly mapped to the HSM implicit role "STC User" when the Secure Trusted Channel (STC) is enforced between the client application and the TOE. As mentioned previously, usage of the STC is mandatory for remote client applications and optional for local ones. In the latter case, when STC is not used, S.Application is not mapped to any specific role supported by the TOE, and the first authentication step occurs at the session level.

7.2 Authentication

7.2.1 Authentication methods

STC User (e.g. remote client application) is authenticated through the verification of a digital certificate during the establishment of the STC secure channel between the application and the module.

Two models for authentication are supported for all the roles supported at the session level:

- **Token based authentication** - where authentication data is provided by means of a trusted PIN Entry Device (PED). The PED can be connected locally through a separate port to the TOE (local PED) or remotely (remote PED). The PED and software required to operate it are the responsibility of the IT Environment.
- **Password based authentication** – where authentication is based on a configurable length password.

The model of authentication to be used is determined during HSM manufacturing and this model is used for all roles.

Key Owner is authenticated through the verification of the key authorization data associated to each key (whatever it is a General Key or an Assigned Key). Authorization as the Key Owner is always separately required before a key can be actually used in a cryptographic function (or exported), regardless of any other authorization that may have been established.

7.2.2 Allowed operations before authentication

For all roles, identification and authentication are done simultaneously. The operations/commands that can be executed prior authentication depend on the considered role as listed below.

Table 7-1 Allowed operations before authentication by role.

Role	Operations/commands allowed before authentication
STC User	> HSM self-tests > Submit bootloader commands for: • Diagnostics and status information • HSM wipe-out • Completing the boot process by passing the execution control to the Firmware • Instructing the Firmware to delete FM applications stored in the TOE's flash memory > query HSM status, authenticated identity of the HSM, configuration and licenses > query container configuration > PED configuration and communication requests > query log status and submit external log messages for addition to secure audit log > STC management operations (request public key, activate channel, open and close channel) > request state of HSM roles > Submit public requests to embedded FM applications

Role	Operations/commands allowed before authentication
All session roles (HSM SO, Partition SO, Administrator, Audit User, Partition CO, Partition Limited CO and Partition CU)	> HSM self-tests • Submit bootloader commands for: • Diagnostics and status information • HSM wipe-out • Completing the boot process by passing the execution control to the Firmware • Instructing the Firmware to delete FM applications stored in the TOE's flash memory > query HSM status, authenticated identity of the HSM, configuration and licenses > query container configuration > PED configuration and communication requests > query log status and submit external log messages for addition to secure audit log > request state of HSM roles > query container object identify (from known OUID or object handle) > session management functions (i.e. open, close, close all, clean access, get session info) > Login requests > HSM deactivation > Zeroize the HSM > Request new initialization of HSM > create, modify, destroy and get attributes of public partition objects > Submit public requests to embedded FM applications
> Key Owner	> HSM Self-tests > Any operation except for: • Cryptographic operation using an Assigned Key or a General Key • Export operation of a General Key • Modification of the authorization data of an Assigned Key or a General Key

Any other operation/command requires authentication of the considered role.

Note: 'HSM zeroisation' means that all user information and user key material are erased. The bootloader, firmware and HSM own data (HSM keys and certificates) are not erased. 'HSM wipe-out' means that all HSM content (user information, user keys, HSM keys and certificates, firmware) are erased; the bootloader still remains.

7.2.3 Authentication failure handling

For all roles, after a defined number of consecutive unsuccessful authentication attempts is met, the module blocks some functionalities until an unblocking condition is satisfied. The functionalities being blocked, as well as the unblocking condition, depend on the considered role according to the table below:

Table 7-2: Summary table for authentication failure handling.

Role	Number of consecutive authentication/authorization failures (as considered in FIA_AFL.1.1)	Functionality being blocked (as considered in FIA_AFL.1.2)	Unblocking condition (as considered in FIA_AFL.1.2)
HSM SO	A positive integer within the range [1 to 3], configurable by the HSM SO. Note: the default value is 3 and can be changed by the HSM SO once authenticated.	All HSM functionalities except HSM re-initialization.	None. HSM is totally zeroized.
Partition SO	A positive integer within the range [1 to 10], configurable by the Partition SO. Note: the default value is 10 and can be changed by the Partition SO once authenticated	All the functionalities of the related partition	None. Partition is totally zeroized
Administrator	Same number as for the HSM SO	Administrator role is locked out.	If the module policy "Partition SO can reset PIN" is enabled, the HSM SO can unlock the Administrator role by resetting its authentication data. Otherwise, there is no unblocking capability, and the Administrator role must be re-initialized.

Role	Number of consecutive authentication/authorization failures (as considered in FIA_AFL.1.1)	Functionality being blocked (as considered in FIA_AFL.1.2)	Unblocking condition (as considered in FIA_AFL.1.2)
Audit User	Same number as for the HSM SO.	Audit user login and related capabilities (role is locked out)	After a 60 seconds time period has elapsed
Partition CO	Same number as for the partition SO	Partition CO, Partition Limited CO and Partition CU roles are locked out.	If the module policy "Partition SO can reset PIN" is enabled, the Partition SO can unlock the Partition CO role by resetting its authentication data. Otherwise, there is no unblocking capability. All user keys contained in the partition are lost and the partition CO role must be re-initialized.
Partition Limited CO	Same number as for the partition SO	Partition Limited CO role is locked out. Partition CO and Partition CU roles are still functional.	The Partition CO can unlock the Partition Limited CO role by resetting its credentials.
Partition CU	Same number as for the partition SO	Partition CU role is locked out. Partition CO and Partition Limited CO roles are still functional.	The Partition CO can unlock the Partition CU role by resetting its credentials.
Key Owner	3	The related key is blocked: all operations on that key or using that key are forbidden (cryptographic operations, export, change of authorization data)	The Partition CO can unblock the key by setting the number of failed authorizations to any integer value in the range [0...2], or by resetting the key authorization data for General Keys.
STC User	1	Locking of STC authentication mechanism	After a 30 second time period has elapsed

Note: 'Zeroisation' (applied to the HSM or to a partition) means that all user information and user key material are erased.

7.2.4 Re-authentication conditions

As mentioned in section 7.2.1, the Key Owner is granted access to a secret key after presentation of the correct key authorization data. The authentication remains valid until an explicit rescinding of previous authorization for access to the secret key.

7.3 Cryptography

7.3.1 Cryptographic key generation

The following Key Generation and Key Derivation algorithms are supported by the module and considered within the present security evaluation:

Table 7-3: Key generation methods.

Key Generation Algorithm	Key Sizes	Applicable Standards
RSA Key Generation	Modulus length 2048, 3072, 4096	FIPS Pub 186-4 [FIPS 186-4] Appendix B.3.3 and B.3.6 with primality tests from C.3
ECC Key Generation	NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B409, B-571 Brainpool curves: brainpoolP160r1, brainpoolP160t1, brainpoolP192r1, brainpoolP192t1, brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP320r1, brainpoolP320t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1	FIPS Pub 186-4 [FIPS 186-4] Appendix B.4.1 and B.4.2, Appendix D RFC- 5639 [IETF RFC Brainpool]
DSA Domain Parameter Generation	Modulus length 2048 and 3072	FIPS Pub 186-4 [FIPS 186-4] chapter 2.1 and Appendix A.1.1.2
DSA Key-Pair Generation	Modulus length 2048 and 3072	FIPS Pub 186-4 [FIPS 186-4] Section A.2.1
Diffie-Hellman (DH) Domain Parameter Generation	Modulus length 2048, 3072 and 4096 bits	FIPS Pub 186-4 [FIPS 186-4] Appendix A.1
Diffie-Hellman (DH) Key-Pair Generation	Modulus length 2048,3072 and 4096 bits	FIPS Pub 186-4 [FIPS 186-4] Appendix A.2.1
AES	128,192 and 256 bits	FIPS Pub 197 [FIPS 197] chapters 3.1 and 6
Generic Secret	128 – 4096 bits (in increments of 8 bits)	N/A

Table 7-4 Key derivation methods.

Key Derivation Algorithm	Key Sizes	Supported PRF / Hashing Function / Cipher	Applicable Standards
Counter Mode KDF	128,192 and 256 bits when AES is cipher. 128 - 4096 bits when HMAC PRF is used	AES-CMAC, HMAC-SHA1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512..	Counter Mode KDF from FIPS SP 800-108 [SP800-108] chapter 4 and 5.1 with FIPS Pub 197 [FIPS 197] for supported cipher.
Single-step KDF	None	SHA-512	[SP800-56C] chapter 4
EC Diffie-Hellman Key Agreement	Curve P-224	SHA-224, SHA-256, SHA-384, SHA-512.	ECC - Ephemeral Unified, One Pass DH from NIST Special Pub 800-56A [SP800-56A]
EC Diffie-Hellman Key Agreement	Curve P-256	SHA-256, SHA-384, SHA-512.	ECC - Ephemeral Unified, One Pass DH from NIST Special Pub 800-56A [SP800-56A]
EC Diffie-Hellman Key Agreement	Curve P-384	SHA-384, SHA-512.	ECC - Ephemeral Unified, One Pass DH from NIST Special Pub 800-56A [SP800-56A]
EC Diffie-Hellman Key Agreement	Curve P521	SHA-512.	ECC - Ephemeral Unified, One Pass DH and Full Unified from NIST Special Pub 800-56A [SP800-56A]
EC Diffie-Hellman Key Agreement	brainpoolP160r1, brainpoolP160t1, brainpoolP192r1, brainpoolP192t1, brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP320r1, brainpoolP320t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1	SHA-224, SHA-256, SHA-384, SHA-512.	ECC - Ephemeral Unified, One Pass DH from NIST Special Pub 800-56A [SP800-56A] RFC- 5639 [IETF RFC Brainpool]
Diffie-Hellman Key Agreement	Modulus 2048, 3072 and 4096 bits	SHA-224, SHA-256, SHA-384, SHA-512.	FFC - dhHybrid1, dhEphem, dhHybrid1Flow and dhOneFlow from NIST Special Pub 800-56A [SP800-56A]

7.3.2 Cryptographic operations

The following cryptographic algorithms are supported by the module and considered within the present security evaluation.

Table 7-5: Digital signature generation

Signature Generation Algorithm	Key Sizes	Hash Algorithm	Applicable Standards
RSA	Modulus length 2048 and 3072 bits	SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	RSASSA-PSS and RSASSA-PKCS1-v1_5 from PKCS #1 [PKCS#1] chapters 8.1.1 and 8.2.1
RSA	Modulus length 4096 bits	SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	RSASSA-PSS and RSASSA-PKCS1-v1_5 from PKCS #1 [PKCS#1] chapters 8.1.1 and 8.2.1
RSA	Modulus length 2048 and 3072 bits	SHA-224, SHA-256, SHA-384, SHA-512	ANSI X9.31 Signature Generation as covered under FIPS Pub 186-4 [FIPS 186-4] chapter 5.4
RSA	Modulus length 4096 bits	SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	ANSI X9.31 Signature Generation as covered under FIPS Pub 186-4 [FIPS 186-4] chapter 5.4
DSA	Modulus length 2048 and 3072 bits	SHA-224, SHA-256, SHA-384, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	FIPS Pub 186-4 [FIPS 186-4] chapter 4.6
ECDSA	NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B409, B-571 Brainpool curves: brainpoolP160r1, brainpoolP160t1, brainpoolP192r1, brainpoolP192t1, brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP320r1, brainpoolP320t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1	SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	FIPS Pub 186-4 [FIPS 186-4] chapter 6.4 and Appendix D RFC- 5639 [IETF RFC Brainpool]

Table 7-6: Digital signature verification

Signature Verification Algorithm	Key Sizes	Hash Algorithm	Applicable Standards
RSA	Modulus length 1024, 2048, 3072 and 4096 bits	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	RSASSA-PSS and RSASSA-PKCS1-v1_5 from PKCS #1 [PKCS#1] chapters 8.1.2 and 8.2.2
RSA	Modulus length 1024, 2048, 3072 and 4096 bits	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	ANSI X9.31 Signature Generation as covered under FIPS Pub 186-4 [FIPS 186-4] chapter 5.4
DSA	Modulus length 1024, 2048 and 3072 bits	SHA-1, SHA-224, SHA-256, SHA-384, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	FIPS Pub 186-4 [FIPS 186-4] chapter 4.7
ECDSA	NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B409, B-571 Brainpool curves: brainpoolP160r1, brainpoolP160t1, brainpoolP192r1, brainpoolP192t1, brainpoolP224r1, brainpoolP224t1, brainpoolP256r1, brainpoolP256t1, brainpoolP320r1, brainpoolP320t1, brainpoolP384r1, brainpoolP384t1, brainpoolP512r1, brainpoolP512t1	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	FIPS Pub 186-4 [FIPS 186-4] chapter 6.4 and Appendix D RFC- 5639 [IETF RFC Brainpool]

Table 7-7: Hash algorithms

Message Digest Algorithm	Key Sizes	Applicable Standards
SHA-224	None	FIPS Pub 180-4 [FIPS 180-4] chapter 6.3
SHA-256	None	FIPS Pub 180-4 [FIPS 180-4] chapter 6.2
SHA-384	None	FIPS Pub 180-4 [FIPS 180-4] chapter 6.5

Message Digest Algorithm	Key Sizes	Applicable Standards
SHA-512	None	FIPS Pub 180-4 [FIPS 180-4] chapter 6.4
SHA3-224	None	FIPS Pub 202 FIPS 202, chapter 5.2 and 6.1
SHA3-256	None	FIPS Pub 202 FIPS 202, chapter 5.2 and 6.1
SHA3-384	None	FIPS Pub 202 FIPS 202, chapter 5.2 and 6.1
SHA3-512	None	FIPS Pub 202 FIPS 202, chapter 5.2 and 6.1

Table 7-8: Symmetric encrypt/decrypt

Symmetric Algorithm	Key Sizes	Supported Mode	Applicable Standards
AES	128, 192 and 256 bits	ECB, CBC, OFB, CFB8, CFB128, CTR	FIPS Pub 197 [FIPS 197] chapter 5 and NIST SP 800-38A [SP800-38A] chapter 6
AES	128, 192 and 256 bits	GCM with 128-bit tag	FIPS Pub 197 [FIPS 197] chapter 5 and NIST SP 800-38D [SP800-38D] chapter 7
AES	128 and 256 bits	XTS-AES	FIPS Pub 197 [FIPS 197] chapter 5 and NIST SP 800-38E [SP800-38E]
AES	128, 192 and 256 bits	AES-KW and AES-KWP	FIPS Pub 197 [FIPS 197] chapter 5 and NIST SP 800-38F [SP800-38F] chapter 6

Table 7-9: Asymmetric encrypt/decrypt

Asymmetric Algorithm	Key Sizes	Applicable Standards
RSA	Modulus length 2048 to 4096 bits	RSAES-OAEP from PKCS #1 v2.1 KTS-OAEP-basic from NIST SP800-56B [SP800-56B] chapter 9.2 KAS1-basic from NIST SP800-56B [SP800-56B] chapter 8.2

Table 7-10: Message Authentication Code (MAC)

MAC Algorithm	Key Sizes	Supported PRF / Hashing Function	Applicable Standards
HMAC	128 – 4096 bits (in increments of 8 bits)	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	FIPS Pub 198-1 [FIPS 198-1]
AES CMAC	128, 192 and 256 bits	N/A	NIST SP 800-38B [SP800-38B] chapter 6
AES GMAC	128, 192 and 256 bits	N/A	NIST SP 800-38D [SP800-38D] chapter 7

7.3.3 Random number generation

The module provides a physical random number generator (PTRNG) that meets [AIS31] PTG.2 requirements.

RNG characteristics consistent with PTG.2 are:

- A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers are output.
- If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source.
- The online test detects non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF does not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.
- The online test procedure is effective to detect non-tolerable weaknesses of the random numbers soon.
- The online test procedure checks the quality of the raw random number sequence. It is triggered continuously. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.
- Test procedure A (as defined in the testing procedures for [AIS31]) does not distinguish the internal random numbers from output sequences of an ideal RNG.
- The average Shannon entropy per internal random bit exceeds 0.997.

The module also provides a hybrid deterministic random number generator (RNG) that meets [AIS31] DRG.4 requirements. RNG characteristics are:

- The internal state of the RNG uses the PTRNG of class PTG.2 as random source.
- The RNG provides forward secrecy.
- The RNG provides backward secrecy even if the current internal state is known.

> The RNG provides enhanced forward secrecy:
  - On demand,
  - On condition: after 2^32 generate requests or 2^32 bits generated, whichever comes first
  - After 10 seconds.

> The internal state of the RNG is seeded by the PTRNG of class PTG.2.

> The RNG generates output for which k >2^34 strings of bit length 128 are mutually different with probability 1-ε, with ε < 2^(-16).

> Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers pass test procedure A (as defined in the testing procedures for [AIS31]) and NIST SP 800-22 test suite.

7.4 User data protection

7.4.1 Flow control policy

The following flow control rules are enforced by the module:

> Export of secret keys is only allowed provided that the secret key is not an Assigned Key, that the secret key is encrypted, and that a secure channel (providing authentication and integrity protection) is used for the export

> Public keys are always exported with integrity protection of their key value and attributes

> Keys are only imported over a secure channel (providing authentication and integrity protection)

> A secret key can only be imported if it is a non-Assigned key

> Keys are only imported in encrypted form

> Unblocking access to a key does not allow any subject other than those authorized to access the key at the time when it was blocked.

> No subject is allowed to access the plaintext value of any secret key directly.

> No subject is allowed to export a secret key in plaintext.

> No subject is allowed to export an Assigned Key

> No subject is allowed to export a secret key without submitting the correct authorization data for the key

> No subject is allowed to access intermediate values in any operation that uses a secret key

> A key with an Export Flag value marking it as non-exportable shall not be exported

7.4.2 Access control policy

The following access control rules are enforced by the module:

> Attributes of a key can only be changed by an authorized subject, and only as permitted in the Key Attributes Modification Table

- Only subjects with current authorization for a specific secret key are allowed to carry out operations using the plaintext value of that key
- Only cryptographic functions permitted by the secret key's Key Usage attribute can be carried out using the secret key

7.4.3 Stored data integrity protection

The module enforces integrity protection on all keys. The integrity mechanism protects both the key value and its security attributes. It consists of a MAC value calculated over the whole key object (key value + security attributes), which is verified prior allowing any operation on (or with) the key. Should the MAC verification fail, the module prohibits the targeted operation and returns an error message. The event is also notified through a record in the audit log.

7.4.4 Handling of residual data

The module ensures that any previous information content of a resource is made unavailable upon the deallocation of the resource from the following objects:

- Authorization data
- Secret keys

7.5 Trusted Channel

The module implements a secure channel (called STC) between itself and external (remote) client applications. The STC enforces authentication of its end points and protects all information communicated over the channel from unauthorized modification and disclosure. It is required to use the STC for all security-related services, including:

- Cryptographic operations
- Operations on keys
- Authentication operations

Note that the communication with local client applications (within the same local server or appliance boundary) doesn't require the STC, since the local environment provides sufficient protection. However, the STC can be used for such communication as well.

The inter-process communication channel between the TOE and the embedded FM applications (stored within the PCI-e card) is considered as a trusted path as well: as for the local client applications, the local environment provides sufficient protection.

7.6 Key Management

7.6.1 Key security attributes

The following security attributes are associated to any General or Assigned Key to enforce the access control and flow control policies:

Table 7-11 Summary key attributes used by the module to support claims in this ST.

Key attribute	Mapping to PP category / naming
CKA_OUID	Key ID
CKA_CLASS	Key type
CKA_KEY_TYPE	Key type
CKA_AUTH_DATA	Authorization Data
CKA_ENCRYPT	Key Usage
CKA_DECRYPT	Key Usage
CKA_WRAP	Key Usage
CKA_UNWRAP	Key Usage
CKA_SIGN	Key Usage
CKA_VERIFY	Key Usage
CKA_DERIVE	Key Usage
CKA_EXTRACTABLE	Export Flag
CKA_ASSIGNED	Assigned Flag
CKA_MODIFIABLE	-
CKA_FAILED_KEY_ AUTH_COUNT	-

The TSF enforces the following modifications rules on security attributes of General Keys:

Table 7-12: Summary of key attribute modification rules for General Keys.

Key attribute	Mapping to PP category / naming	Modification conditions
CKA_OUID	Key ID	Cannot be modified
CKA_CLASS	Key type	Cannot be modified
CKA_KEY_TYPE	Key type	Cannot be modified

Key attribute	Mapping to PP category / naming	Modification conditions
CKA_AUTH_DATA	Authorization Data	Case of a User Partition: the attribute can be modified by Key Owner only when modification operation includes successful validation of current (pre-modification) authorization data, or by Partition CO. Case of the Admin Partition: the attribute can be modified by Key Owner only when modification operation includes successful validation of current (pre-modification) authorization data, or by the HSM SO.
CKA_ENCRYPT	Key Usage	Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false). Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).
CKA_DECRYPT	Key Usage	Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false). Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).
CKA_WRAP	Key Usage	Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false). Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).
CKA_UNWRAP	Key Usage	Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false). Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).

Key attribute	Mapping to PP category / naming	Modification conditions
CKA_SIGN	Key Usage	Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false). Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).
CKA_VERIFY	Key Usage	Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false). Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).
CKA_DERIVE	Key Usage	Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false). Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).
CKA_EXTRACTABLE	Export Flag	Case of a User Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false). Case of the Admin Partition: If (CKA_MODIFIABLE == true), the attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well. Cannot be modified if (CKA_MODIFIABLE == false).
CKA_ASSIGNED	Assigned Flag	Case of a User Partition: the attribute can be modified only by Partition CO, and only to change from non-assigned to assigned Case of the Admin Partition: the attribute can be modified only by the HSM SO, and only to change from non-assigned to assigned

Key attribute	Mapping to PP category / naming	Modification conditions
CKA_MODIFIABLE	-	Case of a User Partition: The attribute is modifiable by Partition CO or Partition Limited CO provided he/she is the Key Owner as well, and only to change from modifiable to non-modifiable. Case of the Admin Partition: The attribute is modifiable by HSM SO or by Administrator provided he/she is the Key Owner as well, and only to change from modifiable to non-modifiable.
CKA_FAILED_KEY_ AUTH_COUNT	-	Case of a User Partition: the attribute can be modified only by Partition CO Case of the Admin Partition: the attribute can be modified only by the HSM SO or by the Administrator.

The TSF enforces the following modifications rules on security attributes of Assigned Keys:

Table 7-13: Summary of key attribute modification rules for Assigned Keys.

Key attribute	Mapping to PP category / naming	Modification conditions
CKA_OUID	Key ID	Cannot be modified
CKA_CLASS	Key type	Cannot be modified
CKA_KEY_TYPE	Key type	Cannot be modified
CKA_AUTH_DATA	Authorization Data	Can be modified by Key Owner only when modification operation includes successful validation of current (pre-modification) authorization data
CKA_ENCRYPT	Key Usage	Cannot be modified
CKA_DECRYPT	Key Usage	Cannot be modified
CKA_WRAP	Key Usage	Cannot be modified
CKA_UNWRAP	Key Usage	Cannot be modified
CKA_SIGN	Key Usage	Cannot be modified
CKA_VERIFY	Key Usage	Cannot be modified
CKA_DERIVE	Key Usage	Cannot be modified
CKA_EXTRACTABLE	Export Flag	Cannot be modified
CKA_ASSIGNED	Assigned Flag	Cannot be modified

Key attribute	Mapping to PP category / naming	Modification conditions
CKA_MODIFIABLE	-	Cannot be modified
CKA_FAILED_KEY_ AUTH_COUNT	-	Case of a User Partition: the attribute can be modified only by Partition CO Case of the Admin Partition: the attribute can be modified only by the HSM SO.

The TSF enforces the following initialization rules on the attributes:

Key Attribute	Initialization rules (Assigned Key)	Initialization rules (General Key)
CKA_OUID	Initialized by generation process	
CKA_CLASS	Initialized by generation process	
CKA_KEY_TYPE	Initialized by generation process	
CKA_AUTH_DATA	Case of a User Partition: Initialized by Key Owner or by Partition CO during generation (no default value). Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation (no default value).	
CKA_ENCRYPT	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.	
CKA_DECRYPT	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.	
CKA_WRAP	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.	
CKA_UNWRAP	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.	
CKA_SIGN	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.	

Key Attribute	Initialization rules (Assigned Key)	Initialization rules (General Key)
CKA_VERIFY	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.	
CKA_DERIVE	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.	
CKA_EXTRACTABLE	Set to false.	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is false. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is false.
CKA_ASSIGNED	Set to true.	Set to false
CKA_MODIFIABLE	Set to false.	Case of a User Partition: Initialized by Key Owner or by Partition CO or by Partition Limited CO during generation. Default value is true. Case of the Admin Partition: Initialized by Key Owner or by the HSM SO or by the Administrator during generation. Default value is true.
CKA_FAILED_KEY_ AUTH_COUNT	Set to 0	Set to 0

Note related to key import: The TSF will assign the default values specified above to imported general keys, for the attributes that are missing from the imported key (if any).

7.6.2 Key destruction

The module performs zeroisation to destroy cryptographic keys. Zeroisation is achieved through the following means:

Table 7-14: Summary key destruction methods.

Key deletion method (KDM)	Action taken	Context in which the KDM is used
KDM1	Overwrite of memory	Applies to user keys stored in HSM partitions, which can be erased by means of ICD commands. Applies to some HSM support keys as well, that can be erased by means of ICD commands.
KDM2	RAM reset	Resetting of RAM causes the erasure of all RAM resident keys
KDM3	Erasure of entire memory sectors	HSM wipe-out, which can be called from the bootloader, or triggered by the active tamper detection on K7+ TOE. All user keys and HSM support keys are erased.
KDM4	Erasure of the encrypting keys	Any user key stored in a HSM partition is encrypted with a chain of HSM support keys. Erasure of any of these support keys is equivalent to erasing the user key. The same applies to user keys stored externally, which are encrypted with a HSM support key. The same applies to many 'intermediate' HSM support keys which are encrypted by other HSM support keys.
KDM5	Erasure of HSE-BBRAM in response to a tamper/decommission event	Decommission signal (on K7 and K7+ TOEs) or active tamper detection (on K7+ TOE) trigger the erasure of the HSE-BBRAM, which contains the 'top-level' HSM support keys that encrypt the other HSM support keys.

7.6.3 External key storage

User keys (i.e. General Keys and Assigned Keys) can be stored within the module, or externally. This latter case may be necessary when too many user keys have to be handled, making it impossible to store them all inside the module's memories. Key objects are externally stored in the form of "key blobs" including both the key value (in encrypted form) and the key security attributes. Integrity is ensured on the entire key objects (value and attributes) through the mechanism described in section 7.4.3.

7.7 Self-protection

7.7.1 Self-tests

The module runs a suite of the following self-tests during initial start-up (or power-on), periodically during normal operation and on demand to demonstrate correct operation:

> **At initial start-up (or power-on):**

- Software/firmware integrity test

- Cryptographic algorithm tests
- Random number generator tests

> **Periodically during normal operation:** Random number generator tests

> **On demand:**

- Cryptographic algorithm tests
- Software/firmware integrity test
- Random number generator test

The module preserves itself in a secure state in the event of failures detected during the self-tests.

Note: the software/firmware integrity test is done through the verification of the firmware cryptographic signature (which establishes firmware authenticity as well). This verification is done during self-tests and also prior to the installation of any new version of the firmware.

Involved algorithms for firmware signature verification are RSA 4096 bits with SHA-384 hash.

7.7.2 Protection against physical attacks (K7 card)

The K7 printed circuit board (PCB) is coated with an epoxy shield that enables passive detection of any attempt to physically compromise the underlying electronic circuits and components. The physical compromise can be detected by a visual inspection of the K7 PCB. Note that any attempt to remove the epoxy coating has a very high probability to destroy the underlying components, thus making the module inoperable.

The K7 PCB also embeds voltage and temperature sensors which cause the module to reject any command while outside of the expected voltage and temperature range.

7.7.3 Protection against physical attacks (K7+ card)

The K7+ printed circuit board (PCB) is coated with an electrically wired shield that enables both active and passive detection of any attempt to physically compromise the underlying electronic circuits and components, and triggers the erasure of all support and user keys.

The K7+ PCB also embeds voltage and temperature sensors which cause the module to reject any command while outside of the expected voltage and temperature range. Moreover, the module triggers the erasure of all support and user keys if the voltage or temperature are below or above critical thresholds.

7.7.4 Power loss

If power is lost to the module for whatever reason, permanent objects (private keys, etc.) are preserved and remain cryptographically protected; session objects are cleared from the module. The module can be placed back into operation without compromise of its functionality or permanently stored data. In case of power failure in the host IT environment, host system restart or other circumstances that do not affect the module's operational capability, the module will ensure continued protection of sensitive material and will permit recovery from the last logged in state.

7.8 Audit

The module generates an audit record of the following auditable events:

- Start-up and shutdown of the audit functions
- Startup of the TOE;
- Shutdown of the TOE
- Cryptographic key generation
- Cryptographic key destruction
- Failure of the random number generator
- Authentication and authorization failure handling: all unsuccessful authentication or authorization attempts, the reaching of the threshold for the unsuccessful authentication or authorization attempts and the blocking actions taken;
- All attempts to import or export keys
- All modifications to attributes of keys;
- Integrity errors detected for keys;
- Failures to establish secure channels
- Self-test completion;
- Failures detected by the TOE;
- All administrative actions;
- Unblocking of access;
- Modifications to audit parameters (affecting the content of the audit log).

The module records within each audit record the following information:

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

For audit events resulting from actions of identified users, the module associates each auditable event with the identity of the user that caused the event.

The module provides reliable time stamps to support the audit capability.

Audit records are automatically exported by the module to an audit server in the IT environment. The path to the audit server has to be configured by the Audit User role. If the link to the audit server is lost, the module stores the audit records internally until the connection to the audit server is back or the module memory is full. In this latter case (memory is full), the module rejects any command that would have triggered an additional audit record.

The module implements an integrity protection mechanism on audit records. Once exported to the audit server, the integrity of the records can be verified by the module through a request sent by the System Auditor.

7.9 Firmware updates

The module supports firmware updates. The HSM SO is the only role authorized to load a new version of the firmware within the TOE. The new firmware must be signed with a Thales support key, and the TOE performs a verification of that signature to allow or reject the loading.

The firmware signature (and related verification) is done with RSA-4096 algorithm. The signature is applied to a SHA-384 hash of the firmware.

7.10 Embedded FM application loading

The module supports the loading of FM applications. The HSM SO is the only role authorized to load FM applications on top of the TOE. Any new FM application must be hashed with SHA-512 algorithm and signed with RSA-2048, 3072 or 4096 algorithm. The TOE performs a verification of that signature to allow or reject the loading.

8 Rationales

8.1 Security Objectives Rationale

8.1.1 Security Objectives Coverage

The table below shows the mapping of Threats, Organizational Security Policies and Assumptions to Security Objectives for the TOE and for the TOE Environment.

Table 8-1: Security Problem Definition mapping to Security Objectives

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit	OE.ExternalData	OE.Env	OE.DataContext	OE.AppSupport	OE.Uauth	OE.AuditSupport
T.KeyDisclose	X		X				X		X	X		X			X	X				
T.KeyDerive		X									X									
T.KeyMod			X						X	X		X								
T.KeyMisuse				X	X															
T.KeyOveruse						X														
T.DataDisclose							X										X	X		
T.DataMod								X									X	X		
T.Malfunction													X							
P.Algorithms		X																		
P.KeyControl	X	X		X	X	X			X	X										
P.RNG											X									
P.Audit														X						
A.ExternalData															X					
A.Env																X				
A.DataContext																	X			
A.AppSupport																		X		
A.UAuth																			X	
A.AuditSupport																				X

8.1.2 Security Objectives Sufficiency

The following paragraphs describe the rationale for the sufficiency of the Security Objectives relative to the Threats, OSPs and Assumptions.

8.1.2.1 Threats

T.KeyDisclose is addressed by the requirement in OT.PlainKeyConf to keep plaintext secret keys unavailable, and this is supported in terms of controls over key attributes (which might threaten the confidentiality of the key if modified) in OT.KeyIntegrity. The confidentiality of secret keys that are exported is protected partly by the use of a secure channel as described in OT.DataConf and the requirements for import and export in OT.ImportExport (including the requirement to export secret keys only in encrypted form, or to be able to exclude the export of a key entirely). Physical tamper protection of the keys is provided by OT.TamperDetect (supported by an appropriate inspection procedure as required in OE.Env). Protection of secret key confidentiality during backup is ensured by OT.Backup. The environment also contributes to maintaining secret key confidentiality by protecting any versions of a secret key that may exist outside the TOE, as in OE.ExternalData, and by protecting the operation of the TOE itself by providing a secure environment, as in OE.Env.

T.KeyDerive is addressed by the choice of algorithms that have been endorsed for the appropriate purposes, and this is described in OT.Algorithms. Where keys are generated by the TOE then the use of a suitable random number generator is required by OT.RNG in order to mitigate the risk that an attacker can guess or deduce the key value.

T.KeyMod is addressed by requiring integrity protection of secret and public keys, and their critical attributes in OT.KeyIntegrity, and by requiring use of secure channels that protect integrity if a key is imported or exported (OT.ImportExport). Protection of key integrity during backup is ensured by OT.Backup. Physical tamper protection of the keys is provided by OT.TamperDetect (supported by an appropriate inspection procedure as required in OE.Env).

T.KeyMisuse raises the possibility of a secret key being used for an unintended and unauthorized purpose, and is addressed by the requirement in OT.Auth for the TOE to carry out an authorization check before using a secret key. OT.KeyUseConstraint expands on this to set out requirements for the granularity of authorization.

T.KeyOveruse is concerned with the possibility that more uses may be made of an authorized key than were intended, and this is addressed by the requirements of OT.KeyUseScope which requires controls to be specified and enforced for any re-authorization conditions that the TOE allows a user to define.

T.DataDisclose is concerned with the transmission of data between client applications and the TOE, or between separate parts of the TOE where the transmission passes through an insecure environment. This is addressed by OT.DataConf, which requires the TOE to provide secure channels to protect such communications. The appropriate use of such channels is a requirement for the environment as expressed in OE.DataContext, as is the use of appropriate procedures in OE.AppSupport.

T.DataMod is concerned with the possibility of unauthorized modification of data transmitted between a client application and the TOE, and this is addressed by OT.DataMod which requires that the TOE provides secure channels that can be used to protect the integrity of data that they carry. As with T.DataDisclose, the appropriate use of such channels is a requirement for the environment as expressed in OE.DataContext, as is the use of appropriate procedures in OE.AppSupport.

T.Malfunction is addressed by the requirement in OT.FailureDetect for the TOE to detect certain types of fault.

8.1.2.2 Organisational Security Policies

P.Algorithms requires the use of key generation and other cryptographic functions that are endorsed by appropriate authorities, and this is addressed by OT.Algorithms.

P.KeyControl requires that the TOE can provide controls and support a key lifecycle to ensure that secret keys can be reliably protected against use by those other than the owner of the key, and that the keys can be confined to use for certain cryptographic functions. This is addressed by a combination of TOE objectives as follows:

- OT.PlainKeyConf protects the value of the secret key to prevent the possibility of it being used by unauthorized subjects
- OT.Algorithms ensures that endorsed algorithms that employ and support suitable properties and procedures are provided by the TOE
- OT.Auth, OT.KeyUseConstraint and OT.KeyUseScope ensure that the TOE can provide well-defined limits on the use of a key when it is authorized (as described above for T.KeyMisuse and T.KeyOveruse)
- OT.ImportExport and OT.Backup ensure protection of keys when they are transmitted outside the TOE to client applications or for backup purposes, including the prevention of export of Assigned Keys.

P.Audit requires the TOE to provide an audit trail and this is addressed directly by OT.Audit (which includes protection of the audit records).

8.1.2.3 Assumptions

Each of the Assumptions in section 3.5 is directly matched by a security objective for the operational environment in section 4.2. The wording of each objective for the operational environment includes the wording of each assumption, and no further rationale is therefore given here.

8.2 Security Requirements Rationale

8.2.1 Security Requirements Coverage

The table below summarizes the mapping of Security Objectives for the TOE to SFRs.

Table 8-2: TOE Security Objectives mapping to SFRs.

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FCS_CKM.1		X												
FCS_CKM.4	X													
FCS_COP.1		X												
FCS_RNG.1											X			
FIA_UID.1				X										
FIA_UAU.1				X										
FIA_AFL.1				X										
FIA_UAU.6/KeyAuth				X		X								
FDP_IFC.1/KeyBasics	X				X				X					
FDP_IFF.1/KeyBasics	X		X		X				X					
FDP_ACC.1/KeyUsage					X	X								
FDP_ACF.1/KeyUsage					X	X								
FDP_ACC.1/Backup										X				
FDP_ACF.1/Backup										X				
FDP_SDI.2			X											
FDP_RIP.1	X				X									
FTP_TRP.1/Local			X	X			X	X	X					
FTP_TRP.1/External			X	X			X	X	X					
FPT_STM.1														X
FPT_TST_EXT.1													X	
FPT_PHP.1												X		
FPT_PHP.3												X		

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FPT_FLS.1													X	
FMT_SMR.1				X										X
FMT_SMF.1				X										X
FMT_MTD.1/Unblock				X										
FMT_MTD.1/AuditLog														X
FMT_MSA.1/GenKeys					X									
FMT_MSA.1/AKeys					X									
FMT_MSA.3/Keys					X									
FAU_GEN.1														X
FAU_GEN.2														X
FAU_STG.2														X

OT.PlainKeyConf is addressed by the requirements in the Key Basics SFP defined in FDP_IFC.1/KeyBasics and FDP_IFF.1/KeyBasics (especially FDP_IFF.1.5/KeyBasics). Secure destruction of keys according to FCS_CKM.4 protects the key value at the end of its lifetime. FDP_RIP.1 protects secret keys from being accessed after they have been deallocated.

OT.Algorithms is addressed by the need to use endorsed standards for FCS_COP.1 (cf. Application Note 14) and the use of an appropriate random number generator in FCS_CKM.1. Note that the refinements to assurance components in section 6.4.1 also specify requirements that ensure clear documentation of endorsed and non-endorsed algorithms and functions provided by the TOE.

OT.KeyIntegrity is addressed primarily by FDP_SDI.2 which requires integrity protection of keys and their attributes by the TOE. FDP_IFF.1/KeyBasics requires that any importing or exporting of keys requires the use of secure channels and integrity protection (cf. the requirement for an integrity-protected channel as part of FTP_TRP.1/Local and FTP_TRP.1/External, which is linked to the Key Basics SFP by Application Note 20 under FDP_IFF.1/KeyBasics).

OT.Auth is addressed by FIA_UID.1, FIA_UAU.1 and FIA_AFL.1 for administrator authentication (with FMT_MTD.1/Unblock and its dependencies on FMT_SMR.1 and FMT_SMF.1 ensuring that appropriate roles and unblocking for authorization and authentication failures are also provided). Authorization for external client applications is provided by the requirements for authentication of endpoints in FTP_TRP.1/Local and FTP_TRP.1/External. Authorization for the use of secret keys is addressed by FIA_UAU.6/KeyAuth.

OT.KeyUseConstraint is addressed by the requirements for well-defined (and securely initialized) key attributes in FMT_MSA.1/GenKeys, FMT_MSA.1/AKeys, and FMT_MSA.3/Keys, and the application

of the attributes to operate constraints on the use of keys in FDP_IFC.1/KeyBasics, FDP_IFF.1/KeyBasics, FDP_ACC.1/KeyUsage and FDP_ACF.1/KeyUsage. FDP_RIP.1 protects authorization data (which enables a key to be used) from being accessed after it has been deallocated.

OT.KeyUseScope is addressed by the Key Usage SFP in FDP_ACC.1/KeyUsage and FDP_ACF.1/KeyUsage and by the re-authorization conditions for use of a secret key specified in FIA_UAU.6/KeyAuth.

OT.DataConf is addressed by the authentication and confidentiality requirements for secure channels in FTP_TRP.1/Local and FTP_TRP.1/External.

OT.DataMod is addressed by the authentication and integrity requirements for secure channels in FTP_TRP.1/Local and FTP_TRP.1/External.

OT.ImportExport is addressed by the requirements for the use of secure import/export through a secure channel and restrictions on how keys are imported and exported to protect confidentiality and integrity in the Key Basics SFP in FDP_IFC.1/KeyBasics and FDP_IFF.1/KeyBasics, and by the requirements on the secure channels themselves in FTP_TRP.1/Local and FTP_TRP.1/External.

OT.Backup separates out the requirements for any backup and restore properties that the TOE may provide and is addressed directly by the Backup SFP in FDP_ACC.1/Backup and FDP_ACF.1/Backup.

OT.RNG is addressed by the requirement in FCS_RNG.1 for a random number generator of an appropriate type, which meets appropriate randomness metrics.

OT.TamperDetect is addressed by the requirement for passive tamper detection in FPT_PHP.1 and the tamper response mechanisms in FPT_PHP.3.

OT.FailureDetect is addressed by the self-test requirements of FPT_TST_EXT.1 and secure failure requirements of FPT_FLS.1.

OT.Audit is addressed in terms of basic creation of audit records by the requirements for audit record generation in FAU_GEN.1 and FAU_GEN.2 and provision of timestamps for use in audit records in FPT_STM.1. Protection of the audit trail is ensured by FAU_STG.2, FMT_MTD.1/AuditLog and FMT_SMF.1. Support for the Administrator role that controls export and deletion of audit records from the TOE is required by FMT_SMR.1.

8.2.2 SFR Dependencies

The dependencies between SFRs are addressed as shown in Table 8-3. Where a dependency is not met in the manner defined in [CC2] then a rationale is provided for why the dependency is unnecessary or else met in some other way.

Table 8-3: SFR Dependencies Rationale

Requirement	Dependencies	Fulfilled by
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1 FCS_CKM.4

Requirement	Dependencies	Fulfilled by
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1 See also note below on key attributes during import or export.
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1 FCS_CKM.4 See also note below on key attributes during import or export.
FCS_RNG.1	No dependencies	-
FIA_UID.1	No dependencies	-
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_UAU.6/KeyAuth	No dependencies	-
FDP_IFC.1/KeyBasics	FDP_IFF.1	FDP_IFF.1/KeyBasics
FDP_IFF.1/KeyBasics	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/KeyBasics FMT_MSA.3/Keys
FDP_ACC.1/KeyUsage	FDP_ACF.1	FDP_ACF.1/KeyUsage
FDP_ACF.1/KeyUsage	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/KeyUsage FMT_MSA.3/Keys
FDP_ACC.1/Backup	FDP_ACF.1	FDP_ACF.1/Backup
FDP_ACF.1/Backup	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Backup The dependency on FMT_MSA.3 is not relevant in this case since the attribute used in FDP_ACF.1/Backup is determined by the ability of the user to authenticate as an administrator according to FIA_UAU.1.
FDP_SDI.2	No dependencies	-
FDP_RIP.1	No dependencies	-
FTP_TRP.1/Local	No dependencies	-
FTP_TRP.1/External	No dependencies	-
FPT_STM.1	No dependencies	-
FPT_TST_EXT.1	No dependencies	-

Requirement	Dependencies	Fulfilled by
FPT_FLS.1	No dependencies	-
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FMT_SMF.1	No dependencies	-
FMT_MTD.1/Unblock	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1/AuditLog	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/GenKeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/AKeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
FMT_MSA.3/Keys	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/GenKeys, FMT_MSA.1/AKeys FMT_SMR.1
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1 FIA_UID.1
FAU_STG.2	FAU_GEN.1	FAU_GEN.1

Key attributes during import or export: the TOE may allow import or export of keys according to the rules in FDP_IFF.1/KeyBasics. For keys that may be imported or exported, the TOE does not place any specific requirements on whether attributes are imported and exported with keys. However, the refinement to AGD_OPE.1 in section 6.4.1 requires that the behavior of the TOE in this situation is described in documentation, and that the evaluators confirm the behavior that is documented. Application Note 41 (for FMT_MSA.1) also requires that the initialization of any attributes on import is described in the Security Target.

8.2.3 Rationale for SARs

The assurance level for this protection profile is **EAL4 augmented with ALC_FLR.2 and AVA_VAN.5**.

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialized processes and practices. It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions.

The TOE described in this protection profile is just such a product. Augmentation results from the selection of **ALC_FLR.2** alongside **AVA_VAN.5**. All the dependencies of AVA_VAN.5 are satisfied by other assurance components in the EAL4 assurance package. ALC_FLR.2 has no dependencies.

The augmentation of including ALC_FLR.2 is in response to existing company practice that has been implemented to meet customer requirements for flaw reporting and fixing.

8.2.4 AVA_VAN.5 Advanced methodical vulnerability analysis

The TOE generates uses and manages the highly sensitive data in the form of secret keys, at least some of which may be used as signature creation data. The protection of these keys and associated security of their attributes and use in cryptographic operations can only be ensured by the TOE itself. While the TOE environment is intended to protect against physical attacks, a high level of protection against logical attacks (especially those that might be carried out remotely) is also necessary, and is therefore addressed by augmenting vulnerability analysis to deal with High attack potential.

8.3 Mapping of SFRs to TSS

SFR	Link to TSS
FCS_CKM.1	Addressed in section 7.3.1 “Cryptographic key generation”
FCS_CKM.4	Addressed in section 7.6.2 “Key destruction”
FCS_COP.1/SigGen_Main	Addressed in section 7.3.2 “Cryptographic operations”
FCS_COP.1/SigVer_Main	Addressed in section 7.3.2 “Cryptographic operations”
FCS_COP.1/SigVer_Bootloader	Addressed in section 7.7.1 “Self-tests”
FCS_COP.1/Digest_Main	Addressed in section 7.3.2 “Cryptographic operations”
FCS_COP.1/Sym_Enc_Dec	Addressed in section 7.3.2 “Cryptographic operations”
FCS_COP.1/ASym_Enc_Dec	Addressed in section 7.3.2 “Cryptographic operations”
FCS_COP.1/MAC	Addressed in section 7.3.2 “Cryptographic operations”
FCS_RNG.1/DRG.4	Addressed in section 7.3.2 “Cryptographic operations”
FIA_UID.1/STC_User	Addressed in section 7.2.2 “Allowed operations before authentication”
FIA_UAU.1/STC_User	Addressed in section 7.2.2 “Allowed operations before authentication”
FIA_UID.1/HSM_Roles	Addressed in section 7.2.2 “Allowed operations before authentication”
FIA_UAU.1/HSM_Roles	Addressed in section 7.2.2 “Allowed operations before authentication”
FIA_UID.1/Key_Owner	Addressed in section 7.2.2 “Allowed operations before authentication”
FIA_UAU.1/Key_Owner	Addressed in section 7.2.2 “Allowed operations before authentication”
FIA_AFL.1	Addressed in section 7.2.3 “Authentication failure handling”
FIA_UAU.6/KeyAuth	Addressed in section 7.2.4 “Re-authentication conditions”
FDP_IFC.1/KeyBasics	Addressed in section 7.4.1 “Flow control policy”
FDP_IFF.1/KeyBasics	Addressed in section 7.4.1 “Flow control policy”
FDP_ACC.1/KeyUsage	Addressed in section 7.4.2 “Access control policy”
FDP_ACF.1/KeyUsage	Addressed in section 7.4.2 “Access control policy”
FDP_ACC.1/Backup	No link, the SFR is trivially met as the functionality is not supported
FDP_ACF.1/Backup	No link, the SFR is trivially met as the functionality is not supported
FDP_SDI.2	Addressed in section 7.4.3 “Stored data integrity protection”

SFR	Link to TSS
FDP_RIP.1	Addressed in section 7.4.4 “Handling of residual data”
FTP_TRP.1/Local/Embedded	Addressed in section 7.5 “Trusted Channel”
FTP_TRP.1/Local/Appliance	Addressed in section 7.5 “Trusted Channel”
FTP_TRP.1/External	Addressed in section 7.5 “Trusted Channel”
FPT_STM.1	Addressed in section 7.8 “Audit”
FPT_TST_EXT.1	Addressed in section 7.7.1 “Self-tests”
FPT_PHP.1	Addressed in section 7.7.2 “Protection against physical attacks (K7 card)” and in section 7.7.3 “Protection against physical attacks (K7+ card)”
FPT_PHP.3/K7	Addressed in section 7.7.2 “Protection against physical attacks (K7 card)”
FPT_PHP.3/K7+	Addressed in section 7.7.3 “Protection against physical attacks (K7+ card)”
FPT_FLS.1	Addressed in section 7.7.4 “Power loss”
FMT_SMR.1	Addressed in section 7.1.4 “Roles”
FMT_SMF.1	Addressed in section 7.2.3 “Authentication failure handling”, section 7.6.1 “Key security attributes”, section 7.8 “Audit”, section 7.4.1 “Flow control policy”, section 7.9 “Firmware updates” and section 7.10 “Embedded FM application loading”.
FMT_MTD.1/Unblock	Addressed in section 7.2.3 “Authentication failure handling”
FMT_MTD.1/AuditLog	Addressed in section 7.8 “Audit”
FMT_MTD.1/FW_update	Addressed in section 7.9 “Firmware updates”
FMT_MTD.1/FM_loading	Addressed in section 7.10 “Embedded FM application loading”
FMT_MSA.1/GenKeys	Addressed in section 7.6.1 “Key security attributes”
FMT_MSA.1/AKeys	Addressed in section 7.6.1 “Key security attributes”
FMT_MSA.3/Keys	Addressed in section 7.6.1 “Key security attributes”
FAU_GEN.1	Addressed in section 7.8 “Audit”
FAU_GEN.2	Addressed in section 7.8 “Audit”
FAU_STG.2	Addressed in section 7.8 “Audit”

APPENDIX A: Bibliography

[CC1] Common Criteria for Information Technology Security Evaluation,
Part 1: Introduction and general model,
Version 3.1 Revision 5, April 2017,
CCMB-2017-04-001

[CC2] Common Criteria for Information Technology Security Evaluation,
Part 2: Security functional requirements,
Version 3.1 Revision 5, April 2017,
CCMB-2017-04-002

[CC3] Common Criteria for Information Technology Security Evaluation,
Part 3: Security assurance requirements,
Version 3.1 Revision 5, April 2017,
CCMB-2017-04-003

[CEM] Common Methodology for Information Technology Security Evaluation,
Evaluation Methodology,
Version 3.1 Revision 5, April 2017,
CCMB-2017-04-004

[CEN EN 419221-1] CEN, TS 419221-1:2016, Protection profiles for TSP Cryptograph Modules – Part 1: Overview

[CEN EN 419221-5] CEN, EN 419221-5:2018, Protection Profiles for TSP Cryptographic Modules – Part 5, Cryptographic Module for Trust Services, v1.0.

[CEN TS 419 241-1] CEN, TS 419 241-1, Requirements for Trustworthy Systems Supporting Server Signing.

[CEN EN 419241-2] CEN EN 419241-2, Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing

[Regulation] REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC

[SOG-IS-Crypto] SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, v1.0, May 2016

[TS 119 312] ETSI TS 119 312
Electronic Signatures and Infrastructures (ESI); Cryptographic Suites

[ISO 19790] ISO/IEC 19790:2012, Information technology – Security techniques – Security Requirements for cryptographic modules, 2nd Edition, 2015-11.

[AIS20] AIS 20, Anwendungshinweise und Interpretationen zum Schema (AIS), Funktionalitätsklassen und Evaluationsmethodologie für deterministischeZufallszahlengeneratoren, BSI, Version 2.1, 2011-12-02.

[AIS31] AIS31, Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 31, Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, BSI, Version 2.1, 2011-12-02.

[AIS_RNG_Classes] BSI, A Proposal for: Functional classes for random number generators, Version 2.0, 18 September 2011.

[CC User Guidance] Covers the four document below:

- 007-013968-001, Thales Luna K7(+) Cryptographic Module, Common Criteria User Guidance – Part1: Preparative Procedures, Revision E, 25th September 202020.
- 007-000465-001, Thales Luna K7(+) Cryptographic Module, Common Criteria User Guidance – Part2: Operational Guidance (General), Revision F, 25th September 2020.
- 007-000466-001, Thales Luna K7(+) Cryptographic Module, Common Criteria User Guidance – Part3: eIDAS Guidance, Revision E, 25th September 2020.
- 007-000467-001, Thales Luna K7(+) Cryptographic Module, Common Criteria User Guidance – Part4: TOE Integration for use in Composite Evaluation, Revision D, 25th September 2020.

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

[FIPS 186-2] Federal Information Processing Standards Publication 186-2, Digital Signature Standards (DSS), NIST, January 27 2000.

[FIPS 186-4] Federal Information Processing Standards Publication 186-4, Digital Signature Standards (DSS), NIST, July 2013.

[FIPS 197] Federal Information Processing Standards Publication 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001.

FIPS 202 Federal Information Processing Standards Publication 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015.

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

[SP800-38A] NIST Special Publication SP800-38A, Recommendation for Block Cipher Modes of Operation – Methods and Techniques, Morris Dworkin, December 2001.

[SP800-38B] NIST Special Publication SP800-38B, Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication, May 2005 (with October 2016 updates).

[SP800-38D] NIST Special Publication SP800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.

[SP800-38E] NIST Special Publication SP800-38E, Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices, January 2010.

[SP800-38F] NIST Special Publication SP800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, Morris Dworkin, December 2012.

[SP800-56A] NIST Special Publication SP800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, Revision 3, April 2018.

[SP800-56B] NIST Special Publication SP800-56B, Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography, Revision 2, March 2019.

[SP800-56C] NIST Special Publication SP800-56C, Recommendation for Key-Derivation Methods in Key-Establishment Schemes, Revision 1, April 2018.

[SP800-90A] NIST Special Publication SP800-90A, Recommendation for Random Number Generation Using Deterministic Bit Generators, Rev1, June 2015.

[SP800-108] NIST Special Publication SP800-108, Recommendation for Key Derivation Using Pseudorandom Functions (Revised), October 2009.

[PKCS#1] PKCS #1: RSA Cryptographic Standard, RSA Laboratories, v2.1

[PKCS#8] PKCS #8: Private-Key Information Syntax Specification, RSA Laboratories, v1.2

[PKCS#11 Base] PKCS #11: Cryptographic Token Interface Base Specification, OASIS, v2.40, 15th April 2015.

[IETF RFC Brainpool] IETF, RFC_5639 – ECC Brainpool Standard Curves & Curve Generation, March 2010.

[END OF DOCUMENT]