Primus HSM

Security Target

Version: 3.1

Date: 2025-10-28

Securosys SA
Max-Höggerstrasse 2
8048 Zürich
Switzerland

Version	Date	Author	Description
0.1	2020.04.24.	Securosys SA	Initial version
0.2	2020.08.25.	Securosys SA	Fixing observations of analysis 1 Update FIPS certificate reference of supported Algorithms
0.3	2020.10.22	Securosys SA	Fixing observations of analysis 2
0.4	2020.12.04	Securosys SA	Clarifying non-toe hw/sw parts
1.0	2021.01.22	Securosys SA	Finalizing TOE version
1.01	2021.02.12	Securosys SA	Fixing version references
1.02	2021.03.19	Securosys SA	Fixing standard references
2.0-rc1	2023.04.20	Securosys SA	Prepared for the re-evaluation of Primus HSM v2.8.22.
3.0-rc1	2024.08.23	Securosys SA	SAM PP conformance added
3.0-rc2	2025.05.07	Securosys SA	Minor updates
3.0-rc3	2025.07.22	Securosys SA	Updates according to evaluator feedback.
3.0-rc4	2025.10.08	Securosys SA	Updates according to evaluator feedback.
3.0	2025.10.22	Securosys SA	Final version for firmware 3.1
3.1	2025.10.28	Securosys SA	Reference updates.

Table of content

List of Tables

List of Figures

1 Terminology

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

Common Criteria terms and definitions are given in [CCP1].

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

Assigned Key

A key (usually a secret key) with the ‘Assigned Flag’ attribute set to ‘assigned’, meaning that:

- the ‘Re-authorisation conditions’ and ‘Key Usage’ attributes cannot be changed
- the Authorisation Data attribute can only be changed by presentation of the current Authorisation 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.

ST Application Note 1

The ST Writer keeps using “Assigned Key” and “Assigned Flag” so can use the same terminology in the ST but technically the Assigned Key flag doesn’t exist as a single attribute but is a combination of different other attributes. A key is considered “assigned key” if the following attributes with all authorisation attributes are set: export=false; modify=false; never-extractable=true; imported=false. Whenever ST writer uses Assigned Key or Assigned Flag we mean the combination of the previously mentioned combination of attributes.

Authorisation 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 authorisation data may include data held within the cryptographic module, data held by administrator(s) or data provided by the application.

Digital 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

NOTE: Such trust services are typically but not necessarily using cryptographic techniques or involving confidential material.

Crytographic Module

As the ST includes the requirements of [EN 419241-2] Cryptographic Module is often used. Please note that Primus HSM fulfills both [EN 419241-2] and [EN 419221-5] requirements so whenever the term Cryptographic Module is used It means Primus HSM, so the TOE itself not an external device.

2 Conformance Claim

2.1 CC Conformance Claim

This ST claims conformance to:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model; CCMB-2022-11-001, CC:2022, Revision 1, November 2022 [CCP1].
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components; CCMB-2022-11-002, CC:2022, Revision 1, November 2022 [CCP2].
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components; CCMB-2022-11-003, CC:2022, Revision 1, November 2022 [CCP3].
- Common Criteria for Information Technology Security Evaluation Part 4: Framework for the specification of evaluation methods and activities; CCMB-2022-11-004, CC:2022, Revision 1, November 2022 [CCP4].
- Common Criteria for Information Technology Security Evaluation Part 5: Pre-defined packages of security requirements; CCMB-2022-11-005, CC:2022, Revision 1, November 2022 [CCP5].

The following must be considered:

- Common Methodology for Information Technology Security Evaluation, Evaluation methodology; CCMB-2022-11-006, CEM:2022, Revision 1, November 2022, [CEM].

2.2 PP Claim

This Security Target claims strict conformance to the

1. Protection Profile [EN 419221-5] Protection Profiles for TSP Cryptographic Modules – Part 5: Cryptographic Module for Trust Services; [EN 419221-5]
2. Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing [EN 419241-2]

2.3 Package Claim

The assurance level for this Security Target is EAL4 augmented with AVA_VAN.5 **(EAL4+ conformant)**.

2.4 Conformance Rationale

This Security Target claims strict conformance with the Protection Profile [EN 419221-5] and Protection Profile [EN 419241-2].

As stated, the ST claims conformance to CC2022 meanwhile the protection profiles above are claiming conformance to CC v3.1 R4. According to CC2022 Transition Policy [TP] the following modifications were done:

Extended components

FCS_RNG.1 was defined as an extended component in the above protection profiles. It was removed from the ST as FCS_RNG is now part of [CCP2]. Also, FPT_TST_EXT.1 was defined as an extended component in Protection Profile [EN 419221-5]. It was removed from the ST from extended components as now it is part of [CCP2]. Even FPT_TST.1 is slightly different from the extended component defined in [EN 419221-5] the essence of the requirement is the same.

TOE fulfils both SFRs and the Security Functional Requirements section contain both SFRs, but those are not extended anymore.

Updated SFRs

FAU_STG.3 was included instead of FAU_STG.2 to match the requirements of the FAU_STG.2 of CCv3.1

FCS_CKM.6 was included instead of FCS_CKM.4 as FCS_CKM.4 is deprecated. Also, FCS_CKM.6 is dependency of all relevant SFRs where it used to be FCS_CKM.4.

FDP_ETC.2 is updated according to new requirements (2.4 and 2.5).

SFRs trivially satisfied

There are a few SFRs in the Security Target that are trivially satisfied due to different reasons. To keep the compliance for the protection profiles the ST Author chose to keep those SFRs in the Security Target in the Security Problem Definition and the different rationales sections but here it is described why those SFRs are trivially satisfied. These are explained in Application Notes in the Security Functional Requirements section as well.

FCS_CKM.3 - The SFR is not relevant. The TOE does key backup using AES-GCM encryption but there is no way of accessing the keys outside the TOE. Those can be decrypted only when it’s restored inside the TOE.

FDP_ACC.1/Supply DTBS/R, FDP_ACF.1/Supply DTBS/R – The TOE does not provide the DTBS/R. It signs the DTBS/R provided by the remote client application (User).

FDP_ETC.2/Signer - The TOE does not export Signer data.

FDP_ITC.2/Signer – There is no Signer data imported to the TOE

FDP_ITC.2/Privileged User – There is no Privileged User data imported to the TOE.

FTP_TRP.1/Local – Installing local client applications on the TOE is not supported, therefore there is no channel between the TOE and local client applications.

FTP_TRP.1/SIC – The TOE does not verify the SIC as a communication end point and it relies on the signer authentication.

FTP_ITC.1/CM – The TOE itself is the CM so there is no “channel” between the TOE and the CM.

3 ST Introduction

The Security Target (ST) was developed based on the Protection Profile (PP) [EN 419221-5] “Protection Profiles for TSP Cryptographic Modules - Part 5: Cryptographic Module for Trust Services”.

3.1 ST Reference

Title	PRIMUS HSM Security Target
Version	3.1
Date	2025.10.28.
Protection Profile	Protection Profiles for TSP Cryptographic modules – Part 5 [EN 419221-5] Trustworthy Systems Supporting Server Signing - Part 2 [EN 419241-2]
Assurance level	EAL 4+ (augmented with AVA_VAN.5)
ST Author	Securosys SA

3.2 TOE Reference

Name	PRIMUS HSM
Version	FW 3.1.0
Series	Series E, Series E2, Series X, Series X2

The TOE is not only one product but the whole E and X series of the PRIMUS HSM. All products of the series run the same firmware and differ only in storage and computing resources. Everything else is the same. The evaluated types of PRIMUS HSM are:

- Series E: E20, E60, E150, ED250, EP700
- Series X: X200, X400, X700, X1000, X2000
- Series E2: E2-C914, E2-LX212
- Series X2: E2Y-C914, E2Y-LX212

3.3 TOE Overview

3.3.1 TOE type

A hardware security module (HSM) is a physical computing device that creates, safeguards, and manages digital keys for digital signatures and other cryptographic operations. The TOE is the Primus HSM which is a physically secure HSM with cryptographic toolkit functionality provided over multiple APIs (PKCS11, JCE, CNG). The Module meets [FIPS 140-3] overall Level 3 requirements.

The TOE software implements the Signature Activation Protocol (SAP). It uses the Signature Activation Data (SAD) from the signer to activate the corresponding signing key. The TOE (as a SAM and a CM) is a QSCD. It can also be used as a standalone CM and can be connected to another external SAM.

3.3.1.1 Usage and major security features of the TOE

The Primus HSM generates cryptographic keys, stores these keys, and manages the distribution of these keys. Besides key management, it performs a variety of authentication and encryption tasks. Primus supports symmetric, asymmetric, key agreement and encapsulation and hashing cryptographic algorithms.

The TOE can be used (but not limited to) as a Cryptographic Module and a SAM module of TSP's supporting requirements for remote signing, or sealing, as specified in Regulation 910/2014. The TOE meets the requirements for Sole Control Assurance Level 2 as defined in [EN 419241-1].

Other than that, the TOE can be used as a general Cryptographic Module, providing network interfaces for external applications for many cryptographic functions. Primus HSM provides a wide selection of application programming interfaces (PKCS#11, JCA/JCE, MS CSP) so that it can be used with almost any business application ranging from simple data encryption to identity management, PKI, strong authentication, and digital-signature generation and verification. The units are easy to install, configure, and integrate into existing networks. The TOE's crypto API is available for client applications via it's connectors implemented for JCE/JCA, MS CNG and PKCS#11. REST API is also available via connectors for better compatibility with client applications. They all invoke the same TOE Crypto API on the network interface through encrypted channels.

Multiple Primus HSMs may be grouped together for redundancy and load-balancing purposes. Each Primus HSM may also be partitioned for multiple users (client application)[1].

The TOE is a separate component with its own hardware and software, communicating via a well-defined physical and logical interface with the client applications.

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

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

In addition to the above, Primus HSM support backup and restore functionality, logging and different roles for the easier and more reliable usage for different use-cases.

Primus HSM uses cryptographic functionality for protecting its assets. There are system keys responsible for encrypting communication channels, backups, and other sensitive data. More details about system keys can be found in TOE Summary Specification. TSF data is also integrity protected.

There are cryptographic self-tests implemented in the TOE that run regularly in the background.

[1] Every client application has their own partition. Whenever a User (Client Application) is removed, all the data on its partition is also deleted.

The TOE is resistant to specific physical attacks as well.

3.3.1.2 Available non-TOE hardware/software/firmware

The following HW and SW components are excluded:

- Power supply (X-Module): The power supply is not considered security relevant. While the device depends on the supply of power, a faulty or rigged power supply cannot reveal any information from the device. The power supply for storing and processing CSPs is not taken directly from the PSU but is created with cascaded DC/DC converters with enough buffering capacity to avoid the risk of revealing information by side-channel monitoring, when performing key operations. In addition to this HW based attenuation of power spikes, the cryptographic cores are designed to consume constant power dependent only on the key length, but not the key content. Overvoltage could potentially destroy some of the power input circuitry and render the device unusable. The tamper circuitry, however, will remain active, due to an independent, battery based, power feed.
- Decanus - Remote access Terminal. Decanus is the remote Administration Terminal for the Primus HSM enabling remote administration for Primus HSM devices. Decanus authenticates itself in the TOE and uses the same functions/API as the local Security Officers. Decanus is not required for the TOE to operate, it is optional to use. Each TOE administrative function is available without Decanus as well.
- Cloning, Clustering - In case of cloning or clustering is configured for the TOE, the keys leave the TOE and are synchronized/imported to another instance of the TOE. These features are protected in integrity and confidentiality. The keys are always encrypted and the configuration of cloning and clustering needs the authorisation of at least two Security Officers.

3.4 TOE Description

3.4.1 Physical scope of the TOE

The exact model types of the TOE are listed in the TOE Reference section.

The physical forms of the Module are depicted in the following Figures. The boundary of the module includes the chassis and everything within. However, this does not include the removable power supplies on the X-Module – they are outside the boundary and may be removed, replaced, etc. The X-Module also relies on Smart Cards as external input/output devices, for the purposes of operator authentication.

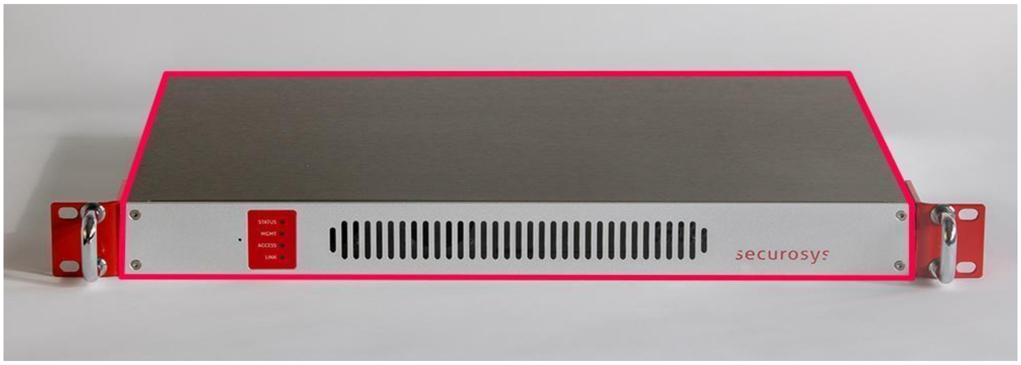

Figure 1 E-Module Front with cryptographic boundary in red

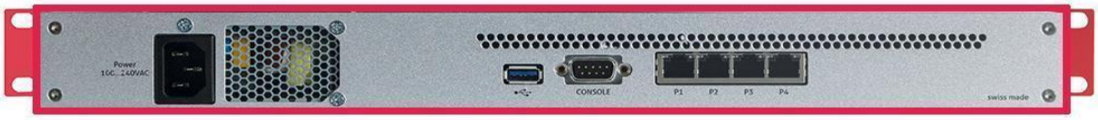

Figure 2 E-Module back with cryptographic boundary in red

Table 1 Ports and Interfaces (E-Series)

Port	Description	Logical Interface Type
Ethernet	4x Ethernet for network connections	Control in | Data in | Data out | Status out
USB	USB port for backup/restore functionality	Control in | Data in | Data out | Status out
Console	RS-232 port for local console access	Control in | Data in | Status out
Power	AC power input	Power
LEDs	Status LEDs (STATUS, MGMT, ACCESS, LINK)	Status out

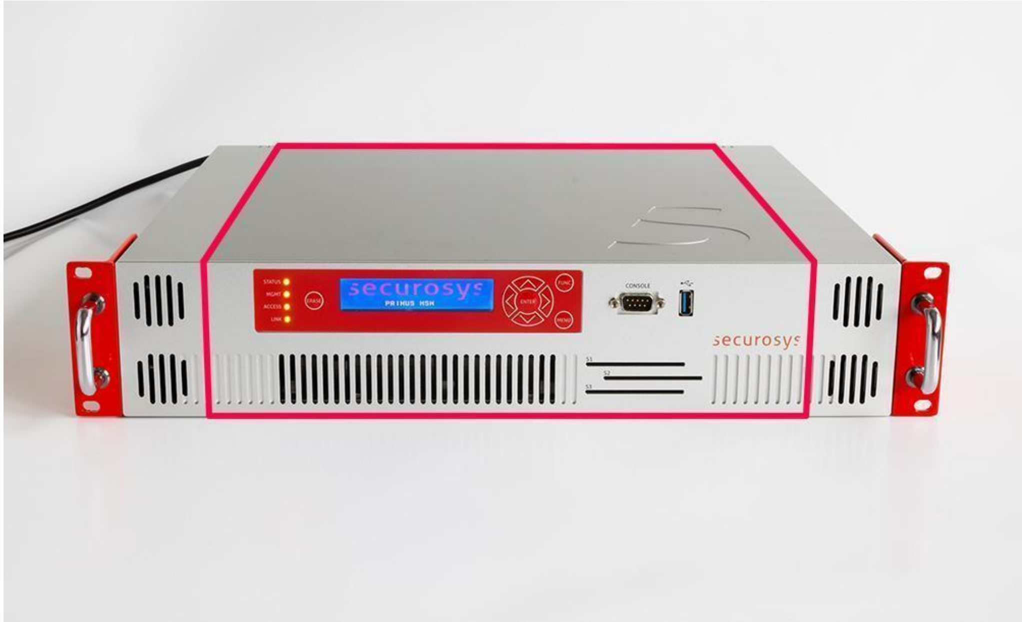

Figure 3 X-Module Front with cryptographic boundary in red

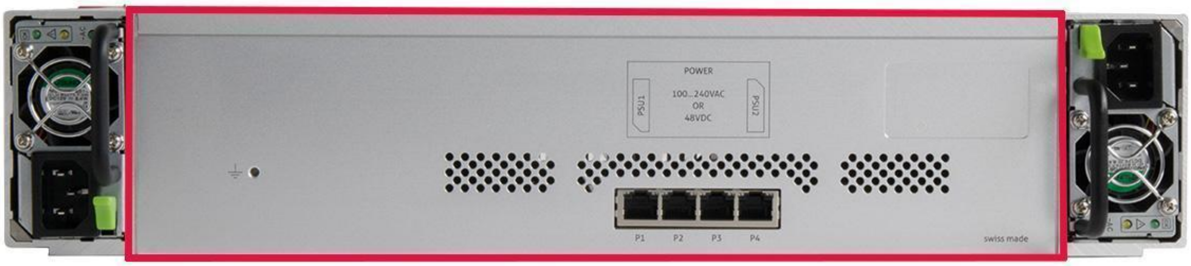

Figure 4 X-Module back with cryptographic boundary in red

Table 2 Ports and Interfaces (X-Series)

Port	Description	Logical Interface Type
Ethernet	4x Ethernet for network connections	Control in | Data in | Data out | Status out
USB	USB port for backup/restore functionality	Control in | Data in | Data out | Status out
Console	RS-232 port for local console access	Control in | Status out | Data in
Card readers	3x Card readers for operator authentication	Data in | Data out
Power	2x DC power inputs (redundant)	Power
Front panel	Front panel LCD and front panel keypad	Control in | Status out
Status LEDs	Status LEDs (STATUS, MGMT, ACCESS, LINK)	Status out

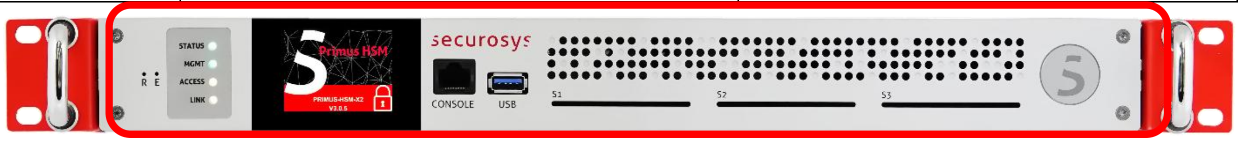

Figure 5 X2-Module front

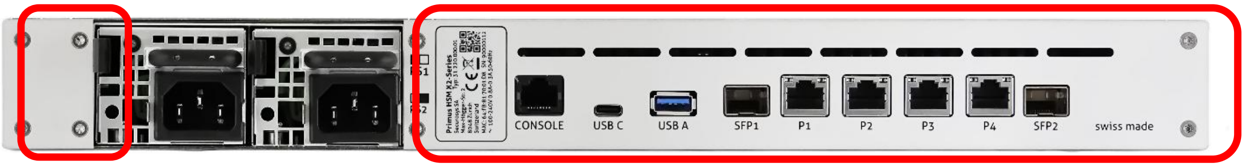

Figure 6 X2/E2-Module back

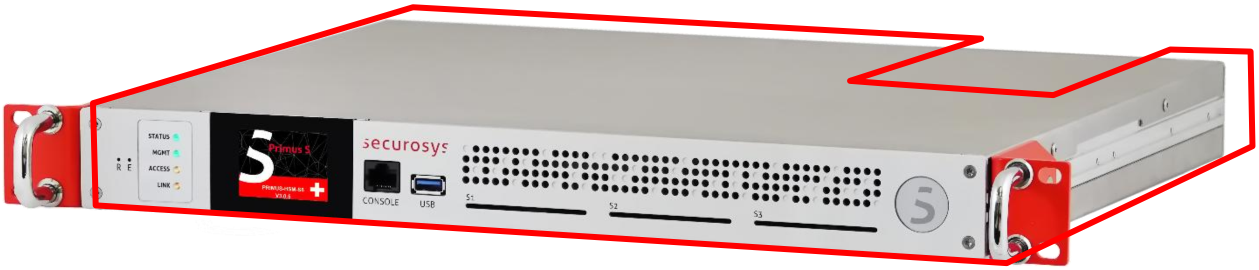

Figure 7 X2-Module top

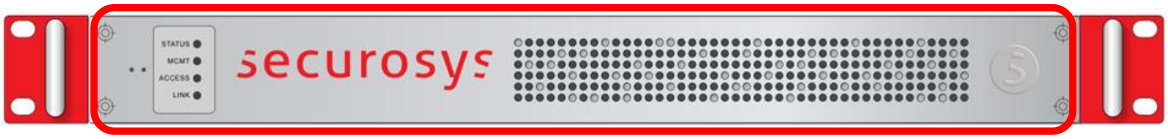

Figure 8 E2-Module front

Table 3 Ports and Interfaces (E2, X2-Series)

Port	Description	Logical Interface Type
Ethernet	4x Ethernet for network connections	Control in | Data in | Data out | Status out
USB	USB port for backup/restore functionality	Control in | Data in | Data out | Status out
Console	RS-232 port for local console access	Control in | Status out | Data in
Card readers	3x Card readers for operator authentication	Data in | Data out
Power	2x DC power inputs (redundant)	Power
Front panel	Front panel LCD and front panel keypad	Control in | Status out
Status LEDs	Status LEDs (STATUS, MGMT, ACCESS, LINK)	Status out

The TOE deliverable parts are as follows:

Table 4 TOE Deliverables

Type	Description	Delivery
HSM module	All E, X, E2, X2 series	Courier
Accessories	**E and E2-Series**	Courier

	- power cable(s) - 1 USB memory stick **X and X2-Series** - 2 power cables - 1 USB memory stick - 2 Genesis Card (GN) - 3 Security Officer (SO) Card	
Guidance	QuickStart guide (printed format)[2]	Courier
Guidance	User Guide (.pdf format) [UG]	Web Download
Firmware	Primus HSM Firmware 3.1.0 (.hsm - encrypted file format)	Courier (pre-installed) or Web Download

[2] QuickStart guide is a part of the [UG]. [UG] completely contains the QuickStart guide in sec 3.1.

3.4.2 Logical scope of the TOE

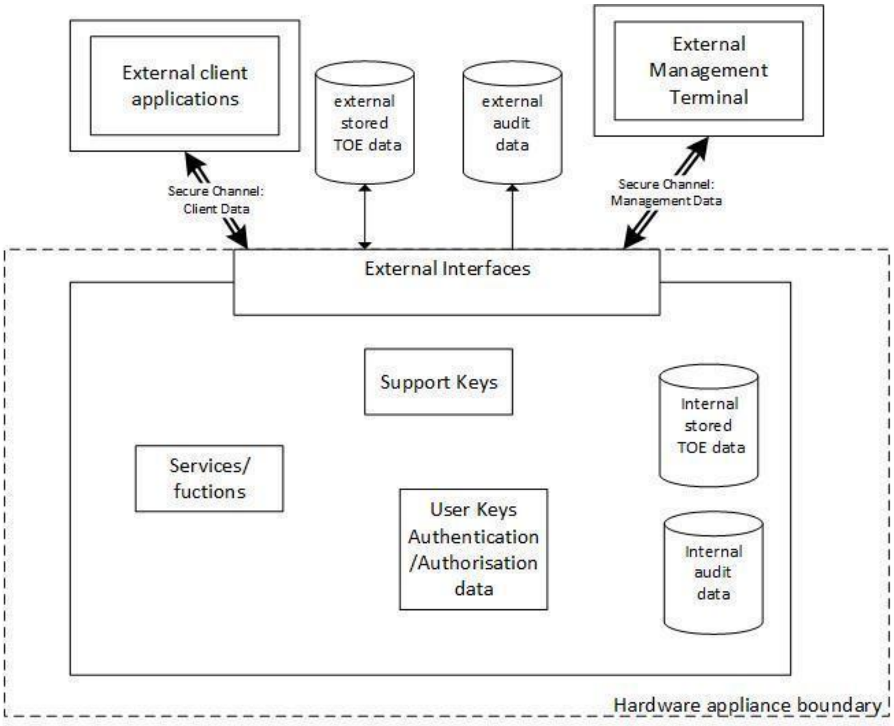
External client applications
external stored TOE data
external audit data
External Management Terminal
Secure Channel: Client Data
Secure Channel: Management Data
External Interfaces
Support Keys
Internal stored TOE data
Services/ fuctions
User Keys Authentication /Authorisation data
Internal audit data
Hardware appliance boundary

Figure 9 TOE Architecture

The hardware appliance boundary in Figures 1-8 represents the enclosure of the computing appliance which hosts the TOE.

3.4.2.1 Authentication and authorization

The TOE implements separate authentication or authorisation of the following distinct types of entity:

- administrators of the TOE
- application users of TOE cryptographic functions (external client applications, authenticated by their use of secure channels)
- users of secret keys (which in at least some cases need to have their use limited to a certain natural person or legal person).

According to [EN 419221-5] terminology Genesis, SO, Partition SO and Partition Auditor roles are the Administrators of the TOE. The detailed Role description of Primus HSM is as follows:

Table 5 Roles and authentication Data

Role ID	Role Description	Authentication Type	Authentication Data
Genesis			PIN and Card

	Administrative role. Sets up the module. Performs factory reset.	Identity-based	PIN Only[3]
Security Officer (SO)	Administrative role which manages the module.	Identity-based	PIN and Card
			PIN Only
User (client application)[4]	Technical User. This role is access through the API and provides general cryptographic functionality for the client application.	Identity-based	Username and Setup Password
			Username and User Secret
Partition SO (Partition security officer)	Administrative role which manages only a partition	Identity-based	Username and one-time Mgmt Setup Password
			Username and User Mgmt Secret
Partition Auditor	Audit role for a partition. PSO with read-only access to management features	Identity-based	Equal to PSO
Signer (End User)[5]	Signer "person" who have access to their keys for cryptographic purposes	Identity-based	Requests signed by their private key and the signature is verified before any cryptographic operation.

TOE supports external client applications. They use a channel that provides authentication of its end-points and protection of confidentiality and integrity of data sent on the channel.

[3] PIN Only Authentication method which consists of card name and pin is available as an option and default on E-Series where there is no card reader slot in the Hardware. This case a "virtual" card is used in the background with all the security features of physical cards including blocking the card – and the admin account as well.

[4] User (Client application) is blocked for 5 minutes in case of 100 failed login attempts within 5 minutes.

[5] Signer does not directly access the TOE but through the client application. Still, they are authorised before using their key and in case of failed authorisation attempt, they are blocked for 5 minutes.

Authorisation as a user (key owner) of a secret key before a key can be used in a cryptographic function (or exported), regardless of any other authorisation that may have been established for administrators or client applications can be done with Primus HSM's SKA (Smart Key Attributes) keys.

A cryptographic function will only be carried out by the TOE if authorisation is obtained for use with a key that can be used with that cryptographic function for SKA and Assigned Keys. Thus, a request by a user (client application) to use a specific cryptographic function may fail if the attributes of the key supplied do not allow its use for that operation.

In case the TOE is configured in eIDAS SAM mode TOE administrators can configure specific CA's on the relevant TOE partition registering their public key on a white list. Whenever a Signer wants to use their signing key, they must authorize with an authorization signature created by a certificate issued by one of the CA's that were previously registered on the whitelist. The CA authenticating the Signer must fulfill the requirements of delegated authentication defined in [EN 419241-1].

Multiple users (client application) can be registered to the TOE. Each user (client application) will have their separate partition of the TOE with their Partition Security Officers defined. A Partition is a totally separate part of the HSM. Each user (client application) has access only to their partition and each Partition Security Officer has administrative access only to their partition. With this solution the TOE can serve multiple client applications.

3.4.2.2 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 agreement and distribution
- Key derivation
- Generation of shared secret values
- Cryptographic support for one-time password and other non-PKI based authentication mechanisms
- Random number generation.
- Key Zeroisation according to [FIPS 140-3]

These functions may also be used to support TSP system functions to create electronic seals and electronic timestamps.

The Primus HSM provides a wide selection of application programming interfaces (PKCS#11, JCA/JCE, MS CSP) so that it can be used with almost any business application ranging from simple data encryption to identity management, PKI, strong authentication, and digital-signature[6] generation and verification. The units are easy to install, configure, and integrate into existing networks.

The Module implements the Approved and allowed cryptographic functions listed in the section Cryptographic Algorithms.

Other than the available Cryptographic API, these cryptographic functions are used internally by TOE as well. TOE uses cryptographic functions to protect its data at rest in the database, protect(encrypt) all backups, sign the logs, encrypt communication channels with external IT systems during operations.

[6] For digital signatures the DTBS/R is always provided by the User (Client application)

3.4.2.3 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[7].

The TOE supports the following techniques for establishing keys:

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

Secret keys are associated with attributes that determine their use, such that the correct association between the key and its attributes are protected against unauthorised modification. The specific key attributes maintained by the TOE are as follows.

- 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)
- Authorisation data that enables access to the key (required only for SKA keys)
- Key usage constraints that determine which cryptographic functions that can use the key (e.g. encryption or signature)
- Whether the key is allowed to be exported
- Whether the key is an Assigned Key (see further discussion of assigned keys in the definition of FMT_MSA.1/AKeys in section 6.7)[8]
- 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.

The proper list of key attributes can be found in Table 11: Key attributes modification table.

Re-authorisation of the Assigned keys before using them is always required.

Authorisation to modify the authorisation attributes of an Assigned key is distinct from authorisation to use the key for cryptographic functions.

Keys may leave the TOE in one of three possible 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 authorised security infrastructure operated by a TSP). This reflects the fact that when dealing with large numbers of keys then 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 5: TOE Architecture. Keys stored outside the TOE are wrapped with KEK to protect the confidentiality and

[7] Deletion of keys always means deletion of the keys with all its attributes.

[8] Assigned keys are the keys that are already assigned to a Signer. Signers and their keypairs are always created by the User (Client Application). Signers can delete their keys.

integrity of the key and the binding of the key to its attributes. The external key storage can only be decrypted by the TOE itself.

- Export of keys

  By default, all private and secret keys are non-extractable so cannot be exported. However it is possible to create a key with the Flag "ACCESS_EXTRACTABLE" and configure the HSM to be able to export keys in encrypted form.

  Keys can 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.

- Backup

  The TOE provides facilities for secure backup and restore of the TSF state, as described in section 'Backup'.

A distinction is drawn between export of keys (as a means of storing for future use by the TOE, or for passing to client applications) and creation of backups: the TOE uses separate mechanisms for these operations.

Keys managed by the TOE

Table 6 Critical Security Parameters (CSPs)

CSP	Type	Description / Usage
Internal System CSPs		

KEK	AES-256-GCM
	AES-128-KW(P)
	AES-192-KW(P)
	AES-256-KW(P)
		Protects the Keystore Key and the Card Keys

Keystore Key	AES-256-CBC	Protects all User Keys in Keystore
DRBG Seed	Misc.	Seed for DRBG
DRBG State	Misc.	Internal DRBG state (size varies based on DRBG)
Other System CSPs		
SO Card Keys	AES-ECB-128	Keys for encrypting/decrypting data on Security Officer smart cards.
SO PINs	Misc.	PINs for logging in as a Security Officer (8-12 characters, numerical)

Genesis PIN	8-Digit PIN	Randomly created by the HSM in production and is 8 digits and cannot be changed. It is used only for genesis authentication, backup operations, and factory reset operations.
Backup Key	AES-256-GCM	Encrypts or decrypts a backup of the module configuration.
Securosys Primus Root CA Key	RSA-4096	PKI Key to sign the device PKI Key
Primus Device CA Key	RSA-3072	PKI Key for key attestation
User (client application) CSPs		
API DH Key	DH-2048	Ephemeral DH-2048 private key for establishing an API session (for User (client application) role).
API Initial Secret	Misc.	129-bit password for initial trust establishment to connect an API session, generated by the module using RBG
API Secret	Misc.	256-bit shared secret for establishing an API session, generated by the module using RBG
API Session Key	AES-256-GCM	Encrypts/decrypts between the module and the API. Unique IV per direction.
User Keys	Misc.	Keys of various types (AES, Triple-DES, HMAC, RSA, DSA, ECDSA, DH, ECDH), used by the User for various operations (encrypt data with AES key, verify data with HMAC key, etc.). Refer to Cryptographic Algorithms table for the detailed list of possible algorithm variants.
Partition SO ECDH Key	ECDH 384	Ephemeral EC 384 private key for establishing a Partition SO session.
Partition SO initial secret	Misc.	One time 129-bit password for initial trust establishment to connect a Partition SO session, generated by the module using RBG
Partition SO secret	Misc.	256-bit shared secret for establishing a Partition SO session, generated by the module using RBG

Partition SO Session Key	AES-256-GCM	Encrypts/decrypts between the module and the Partition SO API. Unique IV per direction.
Partition Auditor Session Key	AES-256-GCM	Encrypts/decrypts between the module and the Partition SO API. Unique IV per direction.
Partition Backup Secret	Misc.	One of three parts of the encryption of the Partition Backup. 256-bit secret, generated by the module using RBG
Backup PIN	8-Digit PIN	Randomly created by the HSM and is 8 digits and cannot be changed. It is used only for Partition Backup/restore.

The keys inside the TOE are protected against physical attacks as well. Upon detection of intrusion the keys are zeroised.

3.4.2.4 *Cryptographic Algorithms*

The supported algorithms by the TOE can be found in the table below. All these algorithms are certified in NIST Cryptographic Algorithm Validation Program (CAVP).

Table 7 Cryptographic Algorithms table

Algorithm	Function (Cryptographic operation)	Description
AES	Encryption, Decryption	[FIPS 197, SP 800-38A] Modes: ECB, CBC, CTR Key sizes: 128, 192, 256 bits Validation Number: A5684
AES-CMAC	MAC Generation, MAC Verification	[SP 800-38B] Functions: Key sizes: 128, 192, 256 bits Validation Number: A5687
AES-GCM	Authenticated Encryption, Authenticated Decryption, GMAC Generation, GMAC Verification	[FIPS 197, SP 800-38D] Key sizes: 128, 192, 256 bits IV-Construction: RBG-based Construction with 96-bit random field and 0-bit free field. A unique IV is constructed for each usage. For line encryption an IV is calculated for each direction (send/receive) and increased after each packet. Note: The IV is generated internally at its entirety randomly as per technique 2 of IG A.5.

		AES-GCM 256 is used for encrypting the channels with the external entities of the TOE where needed. (Administrators, client applications). Validation Number: A5685, A5686
DRBG	HMAC DRBG CTR DRBG	[SP 800-90A Rev. 1] HMAC DRBG with internal function SHA-512 Validation Number: A5692
ECDSA	Key Pair Generation, Signature Generation, Signature Verification, Public Key Validation	[FIPS 186-5] Curves/Key sizes: P-224, P-256, P-384, P-521 (Strength: 112, 128, 192, 260) Vendor Affirmed Curves: brainpool224r1, brainpool256r1, brainpool320r1, brainpool384r1, brainpool512r1, frp256v1 Validation Number: A5710
HMAC	Generation, Verification	[FIPS 198-1] SHA sizes: SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Validation Number: A5691
KAS (FFC, ECC)	Key agreement, also used for secure connection with external management device (Decanus) and external application	[SP 800-56A Rev. 3] Parameter sets/Key sizes: FC, EB, EC, ED, EE Modes: dhStatic responder, Static Unified responder Scheme: SHA2 Note: Key establishment methodology provides between 112 and 256 bits of encryption strength Validation Number: A5699
Safe Primes	Key generation Key confirmation	[MODP, ffdhe] Groups: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 Validation Number: A5698
KDA	Key derivation	[SP800-56C Rev. 2] Modes: One Step, Two Step Hash Functions: SHA-2, SHA-3 Mac-Functions: CMAC, HMAC Validation Number: A5701

KBKDF	Key derivation, also used in line encryption for secure connection with external management device (Decanus) and external application	[SP 800-108 Rev. 1] Modes: Counter, Feedback, Double Pipeline Iteration Mode PRFs: CMAC(AES-128/192/256), HMAC (SHA-1, SHA2(224, 256, 384, 512), SHA3(224, 256, 384, 512)) Validation Number: A5696
KTS (Symmetric)	Key Wrap, Key Unwrap	[SP800-38F] Variants: AES-KW, AES-KWP Key Transport – Provides between 128 and 256 bits of encryption strength. Validation Number: A5697
RSA	Key Pair Generation, Signature Generation, Signature Verification, Component Test	[FIPS 186-5, and PKCS #1 v2.1 (PSS and PKCS1.5)] Key sizes: 2048, 3072, 4096, 8192 bits Validation Number: A5709
KTS (Assymetric)	Key Wrap, Key Unwrap Encryption, Decryption	[SP800-56B Rev. 2] Key sizes: 2048, 3072, 4096 bits Modes: RSA-OAEP Validation Number: A5700
SHA	Digital Signature Generation, Digital Signature Verification, component of HMAC and HMAC_DRBG, general hashing, XOF	[FIPS 180-4, FIPS 202] SHA sizes: SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE-128, SHAKE-256 Validation Number: A5689, A5690

SLH-DSA	Key Generation, Signature Generation, Signature Verification	[FIPS 205] Modes: SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHA2-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-256s, SLH-DSA-SHA2-256f Validation Number: A6131
HSS-LMS	Key Pair Generation, Signature Generation, Signature Verification,	[SP 800-208] Hash Functions: SHA-256, SHA-256(192), SHAKE-256, SHAKE-256(128) Tree Sizes: 5, 10, 15, 20, 25 Coefficients: 1, 2, 4, 8 Multitree: 1-8 Validation Number: A5702, A5703
PBKDF	Key derivation	[SP 800-132] Functions: HMAC-based KDF using SHA-1, SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512) Validation Number: A5695
EDDSA	Key pair generation Signature generation Signature verification	[FIPS 186-5] Curves/Key sizes: Ed22519, Ed448 Modes: Pure, preHashed, Context Validation Number: A5694
ML-KEM	Key pair generation Key encapsulation, key decapsulation	[FIPS 203] Modes: ML-KEM-512, ML-KEM-768, ML-KEM-1024 Validation Number: A6129
ML-DSA	Key pair generation Signature generation Signature verification	[FIPS 204] Modes: ML-DSA-44, ML-DSA-65, ML-DSA-87 Validation Number: A6130

SLH-DSA	Key pair generation Signature generation Signature verification	[FIPS 205] Modes: SLH-DSA-SHAKE-128s, SLH-DSA-SHAKE-128f, SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-192f, SLH-DSA-SHAKE-256s, SLH-DSA-SHAKE-256f, SLH-DSA-SHA2-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHA2-192s, SLH-DSA-SHA2-192f, SLH-DSA-SHA2-256s, SLH-DSA-SHA2-256f Validation Number: A6131
XMSS	Key Pair Generation, Signature Generation, Signature Verification	[SP 800-208] Modes: SHA256-10, SHA256-16, SHA256-20, SHAKE256-10, SHAKE256-16, SHAKE256-20 Vendor affirmed stateful signature algorithm

3.4.2.5 Backup

The TOE supports backup and restoration of the TSF state necessary to re-establish an operational state after failure. Backups include their own copies of keys or may make use of a copy of the externally stored form of the keys (i.e. ‘external stored TOE data’ in Figure 5). The TOE will protect the confidentiality of the backup data and detect loss of the integrity of the backup data (including the attributes of the keys which define the intended use of the keys).

The corresponding ‘restore’ operation for the backup can only be carried out under at least dual person control, so the restore shall be approved by two separate administrators.

TOE supports full device backup that is available for Security Officer (SO). It backups the following data:

- Partitions including name, API credentials, partition config
- Keys including certificates, all partition content
- Device Security policy
- Network config
- Master/clone state
- SO operators
- HA Cluster information
- Decanus pairing
- Possibility to restore individual partition only instead of full device

A Partition Backup is also supported which is available for Partition Security Officers (Partition SO). A partition backup contains the following data:

- Partition including name, API credentials, partition config
- Keys including certificates, all partition content

The authorisation of two SOs is needed for all the above features.

3.4.2.6 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 logs audit records for its own actions internally. The audit records have reliable timestamps provided by NTP server validated by the TOE. Internal audit logs can be collected by administrators and exported to external drives via USB or WebDAV. The internal audit storage stores records cyclically, deleting the oldest records when the storage is full so this is the SO's responsibility to backup and safely store the audit logs in time. A syslog server is also configurable which can store the audit logs automatically.

3.4.2.7 Available services by roles

All services implemented by the TOE are listed in the table below. Each service description also describes all usage of CSPs by the service.

G – Genesis SO – Security Officer U – User (Client Application) PSO – Partition SO PA- Partition Audit

Table 8 Authorized Services

Service	Description	G	SO	U	PSO	PA	SIGNER
Initialize HSM	Initialize the HSM from factory settings. Creates a new KEK, a new Keystore Key, a new "first identity" for the SO role (2 SO Operators) Note that this can only be performed on first module access, or directly after performing the Factory Reset service.	X					
SO Login	Log in as the Security Officer (SO)		X				
SO Management	Create additional Security Officer identities and designate a PIN.		X				
User Login	Log in as the User			X			
User Management	Create User, Delete User, Change Username, new User setup Password, new User Secret CSP: uses Card Keys for SO activation		X				
Change Security Configurations	Configuration changes such as security policy, logging policy, user security policies. CSP: uses Card Keys for SO activation		X				

Data Management	Create Keys, Delete Keys, import/export Keys, Use Keys for encryption, signing etc. via Ethernet Port, and access through Client Application, Business Application, or API CSP: uses KEK and keystore Key			X			
Backup	Create an offline Backup File CSP: uses Card Keys for Genesis Activation (SO and Genesis cards are required)		X				
Restore	Restore Data, SO, U, C onto a new HSM device in initial State CSP: uses Card Keys for Genesis Activation		X				
Digital Seal	Display Seal; set new Seal without performing Factory Reset		X				
Factory Reset	Zeroizes all key data and CSP. Restores factory default configuration. Deletes all data, logs, user accounts (identities for the other roles), deletes KEK, sets new Digital Seal	X					
Export Logs to USB	Export all current logfiles to USB		X				
Show Security Status	User, SO, Cluster diagnostics		X				
PKI setup	Set up internal PKI		X				
Partition SO Login	Partition configuration, Partition Backup, Partition logs, partition diagnostics, key invalidation via PSO API				X		
Partition Backup Card setup	Create Partition Backup Card on the HSM to enable partition restore on the HSM.		X				
Partition Auditor Login	Partition logs, partition diagnostics, only view Partition configuration					X	
Use assigned keys	The signer can use his assigned keys for signing by providing authorization for the operation and authentication provided by an externa SAM or by whitelisted CA.						X

3.4.2.8 NON-TOE features

The TOE has some features which there are no requirements defined in the [EN 419221-5] neither in the [419241-2] therefore these features are out of the scope of this evaluation. These features are as follows:

- Seeding for blockchain technologies
- Non-approved cryptographic algorithms
- Cloning and active cloning, called Clustering are considered as special types of backup.
    - Cloning is for redundancy of keys for failover or load balancing issues. Unlimited clones of other TOE devices can be created if needed. During cloning the whole keystore and security policies are copied to the clone. The clone will use the same API credentials for the same Users (client applications). Cloning can be configured manually by two SOs via UI, HSM console or with Decanus terminal. Clones are restricted to be direct descendants form the Master.
    - Any clone instance can be made a Master in case of emergency, requiring the original Master SO role
    - Clustering is basically the same as Cloning but it automatically syncs the new keys created on the Master device or on any of the clones. It is also used for load balancing.
    - The authorisation of two SOs is needed for configuring cloning or clustering.

4 Security Problem Definition

4.1 Assets

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

Assets from [EN 419241-2]

R.Signing_Key_Id: The signing key is the private key of an asymmetric key pair used to create a digital signature under the signer's sole control. The signing key can only be used by the Cryptographic Module. The TOE uses the asset R.Signing_Key_Id, which identifies a signing key in the Cryptographic Module. The binding of the R.Signing_Key_Id with R.Signer shall be protected in integrity.

ST Application Note 2

In this specific TOE R.Signing_Key_Id is a String provided by the client application (User) that is validated by the TOE to make sure that it is unique.

R.Authorisation_Data: is data used by the TOE to activate a signing key in the Cryptographic Module. The signing key is identified by R.Signing_Key_Id. It shall be protected in integrity and confidentiality.

R.SVD: signature verification data is the public part, associated with the signing key, to perform digital signature verification. The R.SVD shall be protected in integrity.

The TOE uses a Cryptographic Module for signing key pair generation. As part of the signing key pair generation, Cryptographic Module provides the TOE with R.Signing_Key_Id and R.SVD. The TOE provides the R.SVD to the SSA for further handling for the key pair to be certified.

R.DTBS/R: set of data which is transmitted to the TOE for digital signature creation on behalf of the signer. The DTBS/R(s) is transmitted to the TOE. The R.DTBS/R shall be protected in integrity. The transmission of the DTBS/R(s) to the TOE shall require the sending party - Signer or Privileged User - to be authenticated.

R.SAD: signature activation data is a set of data involved in the signature activation protocol, which activates the signature creation data to create a digital signature under the signer's sole control. The R.SAD must combine:

- The signer's strong authentication as specified in [EN 419241-1]
- If a particular key is not implied (e.g a default or one-time key) a unique reference to R.Signing_Key_Id.
- A given R.DTBS/R.

The whole R.SAD shall be protected in integrity and confidentiality.

R.Signature: is the result of the signature operation and is a digital signature value. R.Signature is created on the R.DTBS/R using R.Signing_Key_Id by the Cryptographic Module under the signer's control as part of the SAP. The R.Signature shall be protected in integrity. The R.Signature can be verified outside TOE using R.SVD.

R.Audit: is audit records containing logs of events requiring to be audited. The logs are produced by the TOE and stored externally. The R.Audit shall be protected in integrity.

R.Signer: is a TOE subject containing the set of data that uniquely identifies the signer within the TOE. The whole R.Signer shall be protected in integrity and confidentiality.

R.Reference_Signer_Authentication_Data: is the set of data used by TOE to authenticate the signer. It contains all the data (e.g. OTP device serial number, phone numbers, protocol settings etc.) and keys (e.g. device keys, verification keys etc.) used by the TOE to authenticate the signer. This may include a SVD or certificate to verify an assertion provided as a result of delegated authentication. The whole R.Reference_Signer_Authentication_Data shall be protected in integrity and confidentiality.

R.TSF_DATA: is the set of TOE configuration data used to operate the TOE. It shall be protected in integrity.

R.Privileged_User is a TOE subject containing the set of data that uniquely identifies a Privileged User within the TOE. It shall be protected in integrity.

R.Reference_Privileged_User_Authentication_Data is the set of data used by the TOE to authenticate the Privileged User. It shall be protected in integrity and confidentiality.

R.Random is random secrets, e.g. keys, used by the TOE to operate and communicate with external parties. It shall be protected in integrity and confidentiality

Assets from [EN 419221-5]

R.SecretKey: secret keys used in symmetric cryptographic functions and private keys used in asymmetric cryptographic functions, managed and used by the CM 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 CM. The asset also includes copies of such keys made for external storage and/or backup purposes. The confidentiality and integrity of these keys must be protected.
R.PubKey: public keys managed and used by the CM 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 must 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 CM that is used to authenticate an administrator (hence to control access to privileged administrator functions such as CM backup, export of audit data) or to authorise a user for access to secret and private keys (R.SecretKey). This asset includes copies of authentication/authorisation data made for external storage and/or backup purposes. The integrity of the RAD must be protected; its confidentiality must also be protected unless the authentication method used means that the RAD is public data (such as a public key).

4.2 Subjects

Subjects from [EN 419241-2]

Signer, which is the natural or legal person who uses the TOE through the SAP where he provides the SAD and can sign DTBS/R(s) using his signing key in the Cryptographic Module.

Privileged User, which performs the administrative functions of the TOE and is able to provide a DTBS/R(s) to the TOE as part of the signature operation.

Subjects from [EN 419221-5]

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/authorisation 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 initialisation, 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.

ST Application Note 3

[EN 419241-2] Privileged Users are [EN 419221-5] S.Application and S.Admin.

[EN 419241-2] Signer is [EN 419221-5] S.User

According to Primus HSM terminology as listed in *Table 5: Roles and authentication Data*

- *Genesis, Security Officer, Partition SO, and Partition Auditor represent S.Admin (Privileged User, Admin)*
- *User (client application) represents S.Application (Privileged User, User)*
- *Signer represents S.User who is an end user of S.Application. Signer is also referenced as "key owner". This is also equal to "Key User" defined by [EN 419221-5].*

Whenever SFRs are referring to Privileged User the SFR is refined to be more specific on which Privileged User is responsible for that requirement.

The Signer does not have direct access to the TOE. The Signer can send requests to the TOE via the User (client application) indirectly accessing it. The Signer is authenticated via delegated authentication but they are authorized in the TOE before every key usage.

4.3 Threats

Threats from [EN 419241-2]

T.ENROLMENT_SIGNER_IMPERSONATION
An attacker impersonates signer during enrolment. As examples, it could be:

- by transferring wrong R.Signer to TOE from RA
- by transferring wrong R.Reference_Signer_Authentication_Data to TOE from RA

The assets R.Signer and R.Reference_Signer_Authentication_Data are threatened.
Such impersonation may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED
An attacker is able to obtain whole or part of R.Reference_Signer_Authentication_Data during enrolment. This can be during generation, storage or transfer to the TOE or transfer between signer and TOE. As examples it could be:

- by reading the data
- by changing the data, e.g. to a known value

The asset R.Reference_Signer_Authentication_Data is threatened.

Such data disclosure may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

The threats on enrolment are threats on the environment in case external authentication is supported by the TOE.

T.SVD_FORGERY
An attacker modifies the R.SVD during transmission to the RA or CA. This results in loss of R.SVD integrity in the binding of R.SVD to signing key and to R.Signer.

The asset R.SVD is threatened.

If the CA relies on the generation of the key pair controlled by the TOE as specified in [ETSI EN 319 411-1] clause 6.3.3 d) then an attacker can forge signatures masquerading as the signer.

T.ADMIN_IMPERSONATION
Attacker impersonates a Privileged User and updates R.Reference_Signer_Authentication_Data, R.Signing_Key_Id or R.SVD.

The assets R.Reference_Signer_Authentication_Data, R.SVD and R.Signing_Key_Id are threatened. Such data modification may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

T.MAINTENANCE_AUTHENTICATION_DISCLOSE
Attacker discloses or changes (e. g. to a known value) R.Reference_Signer_Authentication_Data during update and is able to create a signature.

The assets R.Reference_Signer_Authentication_Data and R.Signing_Key_Id are threatened. Such data disclosure may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

T.AUTHENTICATION_SIGNER_IMPERSONATION
An attacker impersonates signer using forged R.Reference_Signer_Authentication_Data and transmits it to the TOE during SAP and uses it to sign the same or modified DTBS/R(s).

The assets R.Reference_Signer_Authentication_Data, R.SAD and R.Signing_Key_Id are threatened.

T.SIGNER_AUTHENTICATION_DATA_MODIFIED
An attacker is able to modify R.Reference_Signer_Authentication_Data inside the TOE or during maintenance. The asset R.Reference_Signer_Authentification_Data is threatened.

Such data modification may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

T.SAP_BYPASS
An attacker bypasses one or more steps in the SAP and is able to create a signature without the signer having authorised the operation. The asset R.SAD is threatened.

T.SAP_REPLAY
An attacker replays one or more steps of SAP and is able to create a signature without the signer having authorised the operation. The asset R.SAD is threatened.

T.SAD_FORGERY
An attacker forges or manipulates R.SAD during transfer in SAP and is able to create a signature without the signer having authorised the operation.

The asset R.SAD is threatened.

T.SIGNATURE_REQUEST_DISCLOSURE
An attacker obtains knowledge of R.DTBS/R or R.SAD during transfer to TOE.
The assets R.DTBS/R and R.SAD are threatened.

If the R.DTBS/R or R.SAD do not require encrypted data then this threat is mitigated.

T.DTBSR_FORGERY
An attacker modifies R.DTBS/R during transfer to TOE and is able to create a signature on this modified R.DTBS/R

without the signer having authorised the operation on this R.DTBS/R. The asset R.DTBS/R is threatened.

T.SIGNATURE_FORGERY
An attacker modifies R.Signature during or after creation or during transfer outside the TOE. The asset R.Signature is threatened.

T.PRIVILEGED_USER_INSERTION
An attacker is able to create R.Privileged_User including R.Reference_Privileged_User_Authentication_Data and is able to log on to the TOE as a Privileged User.

The assets R.Privileged_User and R.Reference_Privileged_User_Authentication_Data are threatened.

T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION
An attacker modifies R.Reference_Privileged_User_Authentication_Data and is able to log on to the TOE as the Privileged User.

The asset R.Reference_Privileged_User_Authentication_Data is threatened.

T.AUTHORISATION_DATA_UPDATE
Attacker impersonates Privileged User and updates R.Authorisation_Data and may be able to activate a signing key.

The assets R.Authorisation_Data and R.Signing_Key_Id are threatened.

T. AUTHORISATION_DATA _DISCLOSE

Attacker discloses R.Authorisation_Data during update and is able to activate a signing key. The assets R.Authorisation_Data and R.Signing_Key_Id are threatened.

T.CONTEXT_ALTERATION
An attacker modifies system configuration R.TSF_DATA to perform an unauthorised operation. The assets R.Signing_Key_Id, R.SVD, R.SAD, R.Reference_Signer_Authentication_Data and R.TSF_DATA are threatened.

T.AUDIT_ALTERATION
An attacker modifies system audit and is able hide trace of TOE modification or usage. The assets R.SVD, R.SAD, R.Signer, R.Reference_Signer_Authentication_Data, R.DTBS/R, R.Signature, R.AUDIT and R.TSF_DATA are threatened.

T.RANDOM
An attacker is able to guess system secrets R.RANDOM and able to create or modify TOE objects or participate in communication with external systems.

Threats from [EN 419221-5]

T.KeyDisclose Unauthorised disclosure of secret/private key

An attacker obtains unauthorised 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 unauthorised 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 authorisation, usage constraints or identifier (changing the identifier to the identifier used for another key would allow unauthorised 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[9].

T.KeyMisuse Misuse of a key

An attacker uses the TOE to make unauthorised use of a secret key (R.SecretKey) that is managed by the TOE (including the unauthorised use of a secret key for a cryptographic function that is not permitted for that key[10]), 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 authorised for a specific use (e.g. to make a single signature) in other cryptographic functions that have not been authorised.

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 Unauthorised modification of client application data

An attacker modifies data (R.ClientData such as DTBS/R, authentication/authorisation 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 realised. 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.

4.4 Organisational Security Policies

Organizational Security Policies from [EN 419241-2]

OSP.RANDOM

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

ST Application Note 4

[9] See OT.KeyIntegrity in section 4.1 for further discussion of critical attributes of a key.

[10] This therefore means that the threat includes unauthorised use of a cryptographic function that makes use of a key.

This is covered by P.RNG from [EN 419221-5]

OSP.CRYPTO
The TOE shall only use algorithm, algorithm parameters and key lengths endorsed by recognized authorities as appropriate by TSPs. This includes generation of random numbers, signing key pairs and signatures as well as the integrity and confidentiality of TOE assets.

ST Application Note 5

This is covered by P.Algorithms from [EN 419221-5]

Organizational Security Policies from [EN 419221-5]

P.Algorithms Use of approved cryptographic algorithms

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

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[11]), 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 authorised cryptographic functions.

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/authorisation 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.

4.5 Assumptions

Assumptions from [EN 419241-2]

A.PRIVILEGED_USER
It is assumed that all personnel administering the TOE are trusted, competent and possesses the resources and skills required for his tasks and is trained to conduct the activities he is responsible for.

A.SIGNER_ENROLMENT
The signer shall be enrolled and certificates managed in conformance with the regulations given in [eIDAS]. Guidance for how to implement an enrolment and certificate management system in conformance with [eIDAS] are given in e.g. [EN 319 411-1] or for qualified certificate in e.g. [EN 319411-2].

A.SIGNER_AUTHENTICATION_DATA_PROTECTION
It is assumed that the signer will not disclose his authentication factors.

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

A.SIGNER_DEVICE
It is assumed that the device and SIC used by signer to interact with the SSA and the TOE is under the signer's control for the signature operation, i.e. protected against malicious code.

A.CA
It is assumed that the qualified TSP that issues qualified certificates is compliant with the relevant requirements for qualified TSP's as defined in [eIDAS].

A.ACCESS_PROTECTED
It is assumed that the TOE operates in a protected environment that limits physical access to the TOE to authorised Privileged Users. The TOE software and hardware environment (including client applications) is installed and maintained by Privileged Users in a secure state that mitigates against the specific risks applicable to the deployment environment.

It is assumed that any audit generated by the TOE are only handled by authorised personal in a physical secured environment. The personal that carries these activities should act under established practices.
It is assumed that where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities must provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment.

ST Application Note 6

Any Audit generated by the TOE are signed by the TOE and is therefore integrity protected and do not need special handling. There is no data managed outside the TOE that is not protected. There are audit logs, configurations and backups that can leave the TOE, but all of them are protected or can have any effect on the TOE when it's loaded back in the TOE and decrypted.

A.AUTH_DATA
It is assumed that the SAP is designed in such a way that the activation of the signing key is under sole control of the signer with a high level of confidence. If SAD is received by the TOE, it must be assumed that the SAD was submitted under the full control of the signer by means that are in possession of the signer.

A.TSP_AUDITED
It is assumed that the TSP deploying the SSA and TOE is a qualified TSP according to article 3 (20) of Regulation EU) No 910/2014 [eIDAS] and audited to be compliant with the requirements for TSP's given by [eIDAS].

A.SEC_REQ
It is assumed that the TSP establishes an operating environment according to the security requirements for SCAL2 defined in [EN 419241-1].

Assumptions from [EN 419221-5]

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 authorised Administrators. The TOE software and hardware environment (including client applications) is installed and 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/authorisation data from its users and securely transfer it to the TOE (protecting the confidentiality of the authentication/authorisation data as required) when required to authorise 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.

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, crypto periods and key renewal, and key/certificate revocation.

5 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.

5.1 Security Objectives for the TOE

OT from [EN 419241-2]

OT.SIGNER_PROTECTION
The TOE shall ensure that data associated to R.Signer are protected in integrity and if needed in confidentiality.

OT.REFERENCE_SIGNER_AUTHENTICATION_DATA
The TOE shall be able to securely handle signature authentication data, R.Reference_Signer_Authentication_Data, as part of R.Signer.

OT.SIGNER_KEY_PAIR_GENERATION
The TOE shall be able to securely use the Cryptographic Module to generate signer signing key pairs and assign R.Signing_Key_Id and R.SVD to R.Signer.

OT.SVD
The TOE shall ensure that the R.SVD linked to R.Signer is not modified before it is certified.

OT.PRIVILEGED_USER_MANAGEMENT
The TOE shall ensure that any modification to R.Privileged_User and R.Reference_Privileged_User_Authentication_Data are performed under control of a Privileged User.

OT.PRIVILEGED_USER_AUTHENTICATION
The TOE shall ensure that an administrator with a Privileged User is authenticated before any action on the TOE is performed.

OT.PRIVILEGED_USER_PROTECTION
The TOE shall ensure that data associated to R.Privileged_User are protected in integrity and if needed in confidentiality.

OT.SIGNER_MANAGEMENT
The TOE shall ensure that any modification to R.Signer, R.Reference_Signer_Authentication_Data, R.Signing_Key_Id and R.SVD are performed under control of the Signer or Privileged User.

OT.SAD_VERIFICATION
The TOE shall verify the SAD. That is, it shall check there is a link between the SAD elements and ensure the signer is strongly authenticated.

OT.SAP
The TOE shall implement the server-side endpoint of a Signature Activation Protocol (SAP), which provides the following:

- Signer authentication
- Integrity of the transmitted SAD.
- Confidentiality of at least the elements of the SAD which contains sensitive information.
- Protection against replay, bypass of one or more steps and forgery.

OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION
The TOE shall ensure signature authentication data is protected against attacks when transmitted to the TOE which would compromise its use for authentication.

OT.DTBSR_INTEGRITY
The TOE shall ensure that the R.DTBS/R is protected in integrity when transmitted to the TOE.

OT.SIGNATURE_INTEGRITY
The TOE shall ensure that a signature can't be modified inside the TOE.

OT.CRYPTO
The TOE shall only use algorithm, algorithm parameters and key lengths endorsed by recognized authorities. This includes generation of random numbers, signing key pairs and signatures as well as the integrity and confidentiality of TOE assets.

OT.RANDOM
Random numbers generated used by the TOE for use as keys, in protocols 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.SYSTEM_PROTECTION
The TOE shall ensure that modification of R.TSF_DATA is authorised by Privileged User and that unauthorised modification can be detected.

OT.AUDIT_PROTECTION
The TOE shall ensure that modifications to R.AUDIT can be detected

OT from [EN 419221-5]

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 authorised 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 recognised 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

See note under P.Algorithms 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 unauthorised modification (unauthorised modifications include making unauthorised copies of a key such that the attributes of the copy can be changed without the same authorisation 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 Authorisation for use of TOE functions and data

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

- administrators of the TOE
- users of TOE cryptographic functions (client applications using secure channels)
- users of secret keys.

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

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 unauthorised use of any cryptographic functions that use a key.

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

The TOE is required to define and apply clearly stated limits on when authorisation and reauthorisation are required in order for a secret key to be used[12]. For example the TOE may allow secret keys to be used for a specified time period or number of uses after initial authorisation, or may allow the key to be used until authorisation is explicitly rescinded. As another example, the TOE may implement a policy that requires re-authorisation before every use of a secret key.

Application Note

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 reauthentication 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/authorisation data) during transmission between the client application and the TOE, or during transmission between separate parts of the TOE where that transmission passes through an insecure environment.

Application Note

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

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

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/authorisation data or public key certificates) during transmission between the client application and the TOE.

Application Note

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 authorised 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/authorisation 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 selection or modification of records by providing tamper protection (either prevention or detection) for the audit log.

5.2 Security Objectives for the Operational Environment

OE from [EN 419241-2]

OE.SVD_AUTHENTICITY
The operational environment shall ensure the SVD integrity during transmit outside the TOE to the CA.

OE.CA_REQUEST_CERTIFICATE
The operational environment shall ensure that the qualified TSP that issues qualified certificates is compliant with the relevant requirements for qualified TSP's as defined in [eIDAS].
The operational environment shall use a process for requesting a certificate, including SVD and signer information, and CA signature in a way, which demonstrates the signer is in control of the signing key associated with the SVD presented for certification. The integrity of the request shall be protected.

OE.CERTIFICATE_VERFICATION
The operational environment shall verify that the certificate for the R.SVD contains the R.SVD.

OE.SIGNER_AUTHENTICATION_DATA
The signer's management of authentication factors data outside the TOE shall be carried out in a secure manner.

OE.DELEGATED_AUTHENTICATION
If the TOE has support for and is configured to use delegated authentication then the TSP deploying the SSA and TOE shall ensure that all requirements in [EN 419241-1] SRA_SAP.1.1 are met.
In addition, the TSP shall ensure that:

- the delegated party fulfils all the relevant requirements of this standard and the requirements for registration according to the Regulation (EU) No 910/2014 [eIDAS], or
- the authentication process delegated to the external party uses an electronic identification means issued under a notified scheme that is included in the list published by the Commission pursuant to Article 9 of the Regulation (EU) No 910/2014 [eIDAS]

If the signer is only authenticated using a delegated party, the TSP shall ensure that the secret key material used to authenticate the delegated party to the TOE shall reside in a certified cryptographic module consistent with the requirement as defined in [EN 419241-1] SRG_KM.1.1. The audit of the qualified TSP according to [EN 419241-1] shall provide evidence that any delegated party meets requirements from [EN 419241-1] SRA_SAP.1.1. and optionally SRG_KM.1.1 in case the signer is only authenticated using a delegated party.

OE.DEVICE
The device, computer/tablet/smart phone containing the SIC and which is used by the signer to interact with the TOE shall be protected against malicious code. It shall participate using SIC as local part of the SAP and may calculate SAD as described in [EN 419241-1]. It may be used to view the document to be signed.

OE.ENV
The TSP deploying the SSA and TOE shall be a qualified TSP according to article 3 (20) of Regulation (EU) No 910/2014 [eIDAS] and audited to be compliant with the requirements for TSP's given by [eIDAS]. The audit of the qualified TSP shall cover the security objectives for the operational environment specified in this clause.

The TOE shall operate in a protected environment that limits physical access to the TOE to authorised privileged users. 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 unauthorised 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.CRYPTOMODULE_CERTIFIED
If the TOE is implemented as a local application within the same physical boundary as the cryptographic module defined in [EN 419221-5] then the TOE relies on the cryptographic module for providing a tamper-protected environment and for cryptographic functionality and random number generation.
If the TOE is implemented within a separate physical boundary then the TOE relies on the cryptographic module for cryptographic functionality and random number generation. The physical boundary shall physically protect the TOE conformant to FPT_PHP.1 and FPT_PHP.3 in [EN 419221-5].

ST Application Note 7

The TOE is within the same physical boundary so relies on the CM on the protection crypto and random functionality.

OE.TW4S_CONFORMANT
The TOE shall be operated by a qualified TSP in an operating environment conformant with [EN 419241-1].

OE from [EN 419221-5]

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 authorised 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 unauthorised 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/authorisation data from its users and securely transfer it to the TOE (protecting the confidentiality of the authentication/authorisation data as required) when required to authorise 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.

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, crypto periods and key renewal, and key/certificate revocation.

6 Security Functional Requirements

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

Security functional requirements components are drawn from Common Criteria part 2 [CCP2]. Some security functional requirements represent extensions to [CCP2].

The TOE security assurance requirements statements are drawn from the security assurance components from Common Criteria part 3 [CCP3].

6.1 Typographical Conventions

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

SFR operations from [EN 419221-5] and [EN 419241-2] are left as they are in the Protection Profile:

- Refinements are denoted with **bold text**.
- Selections and assignments denoted *italicised*.

ST SFR operations:

- ST operations are the same as the operations in the protection profiles with an additional underline.
- Iteration operation is marked with SFR_NAME/ITERATED_INSTANCE_NAME.

Application notes by the ST Author are marked **ST Application Note**.

6.2 Cryptographic Support (FCS)

FCS_CKM.1	*Cryptographic key generation*

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation]
FCS_CKM.3 Cryptographic key access
[FCS_RBG.1 Random bit generation, or
FCS_RNG.1 Generation of random numbers]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *specified in Cryptographic Algorithms table*[13] and specified cryptographic key sizes *specified in Cryptographic Algorithms table*[14] that meet the following: *specified in Cryptographic Algorithms table*[15].

FCS_CKM.3	*Cryptographic key access*

[13] [assignment: *cryptographic key generation algorithm*]

[14] [assignment*: cryptographic key sizes*]

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

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 or FCS_CKM.5 Cryptographic key derivation]

FCS_CKM.3.1 The TSF shall perform [assignment: type of cryptographic key access] in accordance with a specified cryptographic key access method [assignment: cryptographic key access method] that meets the following: [assignment: list of standards].

ST Application Note 8

The SFR is not relevant. The TOE does key backup using AES-GCM encryption but there is no way of accessing the keys outside the TOE. Those can be decrypted only when it's restored inside the TOE.

FCS_CKM.6 *Timing and event of 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.6.1 The TSF shall destroy *all cryptographic keys listed in Critical Security Parameters (CSPs) table* when *no longer needed*.

FCS_CKM.6.2 The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method *zeroisation* that meets the following: *[FIPS 140-3] Level 3*

ST Application Note 9

Signer Keys can be deleted by User individually via API. When Admins delete a partition all keys on the partition will be deleted as well. All other CSPs are deleted via factory reset by Admins. KEK is deleted on tamper detection.

FCS_COP.1 *Cryptographic operation*

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, or FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.3 Cryptographic key access

FCS_COP.1.1 The TSF shall perform *list of functions specified in Cryptographic Algorithms table*[16] in accordance with a specified cryptographic algorithm *specified in Cryptographic Algorithms table*[17] and cryptographic key sizes *specified in Cryptographic*

[16] [assignment: *list of cryptographic operations*]

[17] [assignment*: cryptographic algorithm*]

Algorithms table[18] that meet the following: *specified in Cryptographic Algorithms table*[19]

FCS_RNG.1 *Generation of random numbers*

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a *deterministic*[20] random number generator ***as defined in [NIST800-90]***[21] that implements: *capability list of class DRG.4 as defined in [KS2011]*.

- (DRG.4.1) The internal state of the RNG shall *use PTRNG of class* ***PTG.3***[22] *as random source*[23].
- (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*[24].
- (DRG.4.5) The internal state of the RNG is seeded by an *PTRNG of class PTG.3*[25] [26]

FCS_RNG.1.2 The TSF shall provide *octets of bits*[27] that meet

- (DRG.4.6) The RNG generates output for which *2^{34}*[28] strings of bit length 128 that are mutually different with probability of *> 1 - 2^{16}*[29].
- (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 DRBG related tests listed in Conditional Self-tests table*[30].[31]

[18] [assignment*: cryptographic key sizes*]
[19] [assignment: *list of standards*]
[20] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]
[21] [refinement: *as defined in [NIST800-90]*]
[22] [refinement: *PTG.2*]
[23] [selection: *use PTRNG of class PTG.2 as random source, have [assignment: work factor], require [assignment: guess work]*]
[24] [selection: *on demand, on condition [assignment: condition], after [assignment: time]*]
[25] [selection: *internal entropy source, PTRNG of class PTG.2, PTRNG of class PTG.3, [other selection]*]
[26] [assignment: *list of security capabilities*].
[27] [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*]
[28] [assignment: *number of strings*]
[29] [assignment: *probability*]
[30] [assignment: *additional test suites*]
[31] [assignment: *a defined quality metric*]

6.3 Identification and authentication (FIA)

FIA_UID.1 *Timing of identification*

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1 The TSF shall allow

(1) *Self-test according to FPT_TST.1*

(2) *<u>None</u>*[32] [33]on behalf of the user to be performed before the user is identified.

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

FIA_UAU.1 *Timing of authentication*

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1 The TSF shall allow

1) Self-test according to FPT_TST.1,

2) Identification of the user by means of TSF required by FIA_UID.1

3) <u>None</u>[34] [35]on behalf of the user to be performed before the user is authenticated.

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

FIA_AFL.1/Admin *Authentication failure handling*

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

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

[33] *[assignment: list of TSF-mediated actions]*

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

[35] *[assignment: list of TSF-mediated actions]*

FIA_AFL.1.1/Admin The TSF shall detect when *4*[36] unsuccessful authentication **or authorisation** attempts occur related to *consecutive failed authentication or authorisation attempts*[37].

FIA_AFL.1.2/Admin When the defined number of unsuccessful authentication **or authorisation** attempts has been *met*[38], the **SO card becomes blocked**[39] *until* **forever so the Admin won't be able to authenticate anymore.**[40]

ST Application Note 10

The Administrators (Genesis, SO, Partition SO or Partition Auditor) cards have 4 PIN tries. If the Administrator fails to enter the correct PIN code, the card (and the relevant account) is disabled forever. The SO cards can be virtual but the behavior is the same with virtual cards as well. In case of successful authentication the Administrator is automatically authorised to perform Administrator operations.

FIA_AFL.1/User	*Authentication failure handling*
Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/User The TSF shall detect when *an administrator configurable positive integer within 1-100*[41] unsuccessful authentication **or authorisation** attempts occur in **an administrator configurable time period within 60-7200 seconds**[42] related to *consecutive failed authentication or authorisation attempts*[43].

FIA_AFL.1.2/User When the defined number of unsuccessful authentication **or authorisation** attempts has been *met*[44], the TSF shall *block* access to *the TOE API for the relevant user*[45] until *an administrator configurable time period within 60-259200 seconds has elapsed*[46].

ST Application Note 11

[36] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[37] [assignment: list of authentication events]
[38] [selection: met, surpassed]
[39] [refinement:TSF shall block access to [assignment: description of the relevant functionality]]
[40] [refinement: [selection: unblocked by [assignment: identification of the authorised subject or role], a time period [assignment: time period] has elapsed]]
[41] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[42] [refinement]
[43] [assignment: list of authentication events]
[44] [selection: met, surpassed]
[45] [assignment: description of the relevant functionality]
[46] [selection:unblocked by [assignment: identification of the authorised subject or role], a time period [assignment: time period] has elapsed]

The default value for API block is: 100 failed attempts within 5 minutes will suspend the client for 5 minutes. Restarting the TOE also unblocks the suspension but takes some time and is allowed only for authorised administrators. Also restarting runs all the self-tests. In case of successful authentication the User is automatically authorised to perform User operations.

FIA_AFL.1/Signer *Authentication failure handling*

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/Signer The TSF shall detect *1*[47] unsuccessful authentication **or authorisation** attempts occur in related to *consecutive failed authentication or authorisation attempts*[48].

FIA_AFL.1.2/Signer When the defined number of unsuccessful authentication **or authorisation** attempts has been *met*[49], the TSF shall *block* access to *the ability to authorise any keys*[50] until *5 minutes has elapsed*[51].

ST Application Note 12

In case of SKA keys the key owner is identified by its digital signature. The public keys of the people who can authorise the keys are stored within the key attributes. This can be different for block, unblock, use and modify authorisation settings. On each request for the usage of the SKA key, the client application forwards the authorisation (signature). If the authorisation signature cannot be verified successfully for the selected operations the authoriser will be blocked for 5 minutes. Therefore, the authoriser is not able to authorise any key in the TOE. In case of the Key Owner the authentication is performed by the User (Client Application) as described in FIA_AFL.1/User. The Key Owner will not have a session but only authorised for one key operation.

FIA_UAU.6/KeyAuth *Re-authenticating*

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1/KeyAuth The TSF shall **authorise and re-authorise**[52] the user **for access to a secret key** under the conditions

(1) Authorisation in order to be granted initial access to the key; and

[47] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]

[48] [assignment: list of authentication events]

[49] [selection: met, surpassed]

[50] [assignment: description of the relevant functionality]

[51] [selection:unblocked by [assignment: identification of the authorised subject or role], a time period [assignment: time period] has elapsed]

[52] re-authenticate

(2) Authorisation on every subsequent access to the key[53]. [54]

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: *the security attribute as defined in FIA_USB.1.* [55]

FIA_UAU.5/Signer Multiple authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1/ Signer The TSF shall provide *digital signature verification*[56] to support Signer authentication.

FIA_UAU.5.2/ Signer The TSF shall authenticate any ***Signer's*** claimed identity according to:

- The signing request is signed by the Signer's private key and verified against its public key before accessing the key[57]

ST Application Note 13

In case of eIDAS SAM mode configured not only the Signer's signature is verified but the Signer's authorization certificate is validated if it was issued by a previously configured CA.

FIA_UAU.5/Privileged User Multiple authentication mechanisms

Hierarchical to: No other components.

[53] *[assignment:list of conditions under which re-authentication is required*]

[54]*[selection:*
1.Re-authorisation of [assignment: identification of secret keys that are subject to re-authorisation conditions below] 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 authorised; □□after the number of uses of the secret key (as specified in the secret key's attributes) for which the secret key was last authorised has already been made; [assignment: *list of authentication events*] [assignment: *list of actions*] re-authenticate *after explicit rescinding of previous authorisation for access to the secret key];*
2. [assignment: list of other conditions under which authorisation and re-authorisation for access to secret keys is required];
3. Authorisation on every subsequent access to the key]

[55] [assignment: *list of security attributes*]

[56] [selection: *[assignment: list of direct authentication mechanisms conformant to [EN 419241-1] SRA_SAP.1.1, [assignment: list of delegated authentication mechanisms conformant to [EN 419241-1] SRA_SAP.1.1]*]

[57] [selection: [assignment:*the rules describing how delegated authentication is verified by the TSF], [assignment: the rules describing how direct authentication mechanisms provide authentication]*]

Dependencies: No dependencies.

FIA_UAU.5.1/ Privileged User

The TSF shall provide: -

1) <u>Smart Card (or virtual card) PIN authentication for Admins</u>

to support **Privileged User <u>(Admin)</u>** authentication;

2) <u>Username and User Secret (256 bit random)</u> [58]

to support **Privileged User <u>(User)</u>** authentication.

FIA_UAU.5.2/ Privileged User

The TSF shall authenticate any **Privileged User's** claimed identity according to: **<u>Smart Card for Admin and Username for User</u>**[59]

ST Application Note 14

The ST author refined this SFR to separate the two types of Privileged Users.

FIA_UID.2	User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification.

Dependencies: No dependencies.

FIA_UID.2.1

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

FIA_USB.1	User-subject binding

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition.

FIA_USB.1.1

The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

1) R.Reference_Signer_Authentication_Data;

2) R.Signing_Key_Id;

3) R.SVD;

4) R.Signer; and

5) <u>R. Authorisation_Data</u> [60]

to Signer

1) R.Reference_Priviliged_User_Authentication_Data; and

[58] [assignment: *list of authentication mechanisms*]

[59] [assignment: *rules describing how the multiple authentication mechanisms provide authentication*]

[60] [assignment: *list of user security attributes*]

2) *R.Privileged User*[61]

to Privileged User ***(User).***[62]

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: -

1) *Whether the subject is a Privileged User* ***(User)*** *authorized to create a new Signer.*

2) *Whether the subject is a Privileged User* ***(Admin)*** *authorized to create a new Privileged User* ***(User)****.*

3) None [63] [64]

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: -

1) *Whether the subject is a Privileged User* ***(User)*** *authorized to modify an R.Signer object.*

~~2) **Whether the subject is a Signer authorized to modify his own R.Signer object.**~~

3) *None* [65] [66]

ST Application Note 15

The Signer cannot modify his own R.Signer object.

6.4 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*[67] on

(1) subjects: all

(2) information: keys

(3) operations: all[68].

FDP_IFF.1/KeyBasics *Simple security attributes*

[61] [assignment: *list of user security attributes*]

[62] [assignment: *list of user security attributes*]

[63] [assignment: *rules for the initial association of attributes]*

[64] [assignment: *rules for the initial association of attributes*]

[65] [assignment: *rules for the changing of attributes*]

[66] [assignment: *rules for the changing of attributes*]

[67] [assignment: *information flow control SFP*]

[68][assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]

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*[69] 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*[70].

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 authorised to access the key at the time when it was blocked*[71].

[69] [assignment: *information flow control SFP*]

[70] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

[71] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

FDP_IFF.1.3/KeyBasics The TSF shall enforce the **following additional information flow control rules:** *none*[72].

FDP_IFF.1.4/KeyBasics The TSF shall explicitly authorise an information flow based on the following rules: *none*[73].

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 authorisation 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*[74]

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*[75] on

(1) *subjects: all*

(2) *objects: keys*

[72] [assignment: *additional information flow control SFP rules*]

[73] [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

[74] *rules, based on security attributes, that explicitly deny information flows*]

[75] [assignment: *access control SFP*]

(3) *operations: all*[76].

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* [77] to objects based on the following:

(1) *whether the subject is currently authorised to use the secret key*

(2) *whether the subject is currently authorised to change the attributes of the secret key*

(3) *the cryptographic function that is attempting to use the secret key*[78].

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 authorise access of subjects to objects based on the following additional rules: *none*[79].

[76] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[77] [assignment: *access control SFP*]

[78] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFPrelevant security attributes, or named groups of SFP-relevant security attributes*]

[79] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

FDP_ACF.1.4/KeyUsage The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[80]

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*[81] on

(1) *subjects: all*

(2) *objects: keys*

(3) *operations: backup, restore*[82].

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*[83] to objects based on the following:

(1) *whether the subject is an administrator*[84].

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 authorised 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*

[80] *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[81] [assignment: *access control SFP*]

[82] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[83] [assignment: *access control SFP*]

[84] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFPrelevant security attributes, or named groups of SFP-relevant security attributes*]

(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* [85].

FDP_ACF.1.3/Backup The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*[43].

FDP_ACF.1.4/Backup The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[86]

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*[87] on all **keys (including security attributes)[88]**, based on the following attributes: *integrity protection data*[89].

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[90].

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*[91] the following objects:

- *authorisation data*
- *secret keys*[92].

[85] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[86] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[87] [assignment: *integrity errors*]

[88] objects

[89] [assignment: *user data attributes*]

[90] [assignment: *action to be taken*]

[91] [Selection: *allocation of the resource to, deallocation of the resource from*]

[92] [assignment: *list of objects*]

FDP_ACC.1/Privileged User Creation	Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ Privileged User Creation

The TSF shall enforce the *Privileged User Creation SFP* [93] on: -

Subjects: Privileged User

Objects: New security attributes for the Privileged User to be created.

Operations: Create_New_Privileged_User:

*The TOE creates R.Privileged_User **(User)** and R.Reference_Privileged_User_Authentication_Data with information transmitted by Privileged User **(Admin)**.*[94]

ST Application Note 16

Primus HSM is shipped with a Genesis card or a Genesis PIN that represents the Privileged User, Admin. During the initial startup the Genesis shall create the first Security Officer (SO) and also the first Privileged User, User (client application).

FDP_ACF.1/Privileged User Creation	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/ Privileged User Creation

The TSF shall enforce the *Privileged User Creation SFP* [95] to objects based on the following: -

1) *whether the subject is a Privileged User **(Admin)** authorized to create a new Privileged User **(User)**.* [96]

FDP_ACF.1.2/ Privileged User Creation

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

1) *Only a Privileged User **(Admin)** who has been authorised for creation of new users can carry out the Create_New_Privileged_User operation.* [97]

[93] [assignment: *access control SFP*]

[94] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[95] [assignment: *access control SFP*]

[96] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

[97] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

FDP_ACF.1.3/ Privileged User Creation — The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *None.* [98]

FDP_ACF.1.4/ Privileged User Creation — The TSF shall explicitly deny access of subjects to objects based on the following additional rule: *None.* [99]

FDP_ACC.1/Signer Creation	Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ Signer Creation — The TSF shall enforce the *Signer Creation SFP* [100] on: -

Subjects: Privileged User ***(User)***

Objects: R.Signer and R.Reference_Signer_Authentication_Data

Operations: Create_New_Signer:

The TOE creates R.Signer and R.Reference_Signer_Authentication_Data with information transmitted by Privileged User [101] ***(User)***

FDP_ACF.1/Signer Creation	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/ Signer Creation — The TSF shall enforce the *Signer Creation SFP* [102] to objects based on the following: -

1) *whether the subject is a Privileged User* ***(User)*** *authorized to create a new Signer.* [103]

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

1) *Only a Privileged User* ***(User)*** *who has been authorised for creation of new users can carry out the Create_New_Signer operation.* [104]

[98] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[99] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[100] [assignment: *access control SFP*]

[101] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[102] [assignment: *access control SFP*]

[103] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

[104] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

FDP_ACF.1.3/ Signer Creation — The TSF shall explicitly authorise access of subjects to objects based on the following additional rules*: None.* [105]

FDP_ACF.1.4/ Signer Creation — The TSF shall explicitly deny access of subjects to objects based on the following additional rule: *None.* [106]

FDP_ACC.1/Signer Maintenance Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ Signer Maintenance — The TSF shall enforce the *Signer Maintenance SFP* [107] *on: -*

Subjects: Privileged User ***(User)*** ~~*and Signer*~~

Objects: The security attributes R.Reference_Signer_Authentication_Data of R.Signer

Operations: Signer_Maintenance:

The Privileged User ***(User)*** ~~*or Signer*~~ *instructs the TOE to update R.Reference_Signer_Authentication_Data of R.Signer.*[108]

FDP_ACF.1/Signer Maintenance 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/ Signer Maintenance — The TSF shall enforce the *Signer Maintenance SFP* [109] to objects based on the following:

1) *Whether the subject is a Privileged User* ***(User)*** ~~*or Signer*~~ *authorised to maintain the Signer security attributes.* [110]

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

[105] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[106] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[107] [assignment: *access control SFP*]

[108] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[109] [assignment: *access control SFP*]

[110] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

1) *Only a Privileged User* ***(User)*** *~~or Signer~~ who has been authorised to maintain a Signer can carry out the Signer_Maintenance operation.* [111]

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

1) *The Signer must be the owner of the R.Signer object to be maintained.*[112]

FDP_ACF.1.4/ Signer Maintenance — The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

1) *If the Signer does not own the R.Signer object, it can't be maintained.*[113]

FDP_ACC.1/Signer Key Pair Generation Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ Signer Key Pair Generation — The TSF shall enforce the *Signer Key Pair Generation SFP* [114] on*: -*
Subjects: Privileged User ***(User)*** *and Signer.*
Objects: The security attributes R.SVD and R.Signing_Key_Id as part of R.Signer.
Operations: Generate_Signer_Key_Pair:
The Privileged User ***(User)*** *or Signer instruct the TOE to request the Cryptographic Module to generate a signing key pair R.Signing_Key_Id and R.SVD and assign them to the R.Signer.* [115]

ST Application Note 17

The TOE supports using pre-generated keys for performance issues. These are called Assigneable keys. Same as the Assigned keys, those are just not yet assigned to any Signer. These keys cannot leave the CM so the same protection applies for the pre-generated keys as for the Assigned ones. These keys are not exportable and not importable.

FDP_ACF.1/Signer Key Pair Generation Security attribute based access control

[111] [assignment*: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[112] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[113] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]
[114] [assignment: *access control SFP*]
[115] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/ Signer Key Pair Generation

The TSF shall enforce the *Signer Key Pair Generation SFP* [116] to objects based on the following: -

1) *whether the subject is a Privileged User* ***(User)*** ~~***or Signer***~~ *authorised to generate a key pair.* [117]

FDP_ACF.1.2/ Signer Key Pair Generation

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

1) *Only a Privileged User* ***(User)*** ~~***or Signer***~~ *who has been authorised to generate the key pair can carry out the Generate_Signer_Key_Pair operation.* [118]

FDP_ACF.1.3/ Signer Key Pair Generation

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: -

1) *The Signer must be the owner of the R.Signer object where the key pair is to be generated.* [119]

FDP_ACF.1.4/ Signer Key Pair Generation

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: -

1) *If the Signer does not own the R.Signer object, key pair shall not be generated.* [120]

FDP_ACC.1/Signer Key Pair Deletion Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ Signer Key Pair Deletion

The TSF shall enforce the *Signer Key Pair Deletion SFP* [121] *on: -*

Subjects: Privileged User ***(Admin)*** ~~***and Signer***~~

Objects: The security attributes R.Signing_Key_Id and R.SVD of R.Signer

Operations: Signer_Key_Pair_Deletion:

[116] [assignment: *access control SFP*]

[117] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

[118] [assignment*: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[119] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[120] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[121] [assignment: *access control SFP*]

The Privileged User ***(Admin)*** *~~or Signer~~ instructs the TOE to delete the R.Signing_Key_Id and R.SVD of R.Signer.*[122]

FDP_ACF.1/Signer Key Pair Deletion	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/ Signer Key Pair Deletion

The TSF shall enforce the *Signer Key Pair Deletion SFP* [123] to objects based on the following: -

1) *Whether the subject is a Privileged User* ***(Admin)*** *~~or Signer~~ authorised to delete the Signer security attributes.* [124]

FDP_ACF.1.2/ Signer Key Pair Deletion

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

1) *Only a Privileged User* ***(Admin)*** *~~or Signer~~ who has been authorised to delete a key pair can carry out the Signer_Key_Pair_Deletion operation.* [125]

FDP_ACF.1.3/ Signer Key Pair Deletion

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: -

1) *The Signer must be the owner of the R.Signer object containing the key pair to be deleted.* [126]

FDP_ACF.1.4/ Signer Key Pair Deletion

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: -

1) *If the Signer does not own the R.Signer object, the key pair can't be deleted.* [127]

The DTBS/R can be supplied to the TOE either by the Signer as part of the SAP, which is covered by the FDP_ACC.1/Signing or by a Privileged User prior the signature operation. The following SFR handles the case where the Privileged User supplies the DTBS/R.

[122] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[123] [assignment: *access control SFP*]

[124] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

[125] [assignment*: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[126] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[127] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

FDP_ACC.1/Supply DTBS/R	Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ Supply DTBS/R The TSF shall enforce the *Supply DTBS/R SFP* [128] on: -
Subjects: Privileged User
Objects: The security attributes R.DTBS/R of R.Signer.
Operations: Supply_DTBS/R:
The Privileged User instructs the TOE to link the supplied DTBS/R(s) to the next signature operation for R.Signer.[129]

ST Application Note 18

The TOE does not provide facility to supply the DTBS/R(s) for signing so this and all other requirements are trivially satisfied that belongs to supplying DTBS/R.

FDP_ACF.1/Supply DTBS/R	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/ Supply DTBS/R The TSF shall enforce the *Supply DTBS/R SFP* [130] to objects based on the following: -

1) *Whether the subject is a Privileged User* ***<u>(User)</u>*** *authorised to supply a DTBS/R(s).* [131]

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

1) *Only a Privileged User* ***<u>(User)</u>*** *who has been authorised to supply a DTBS/R(s) can carry out the Supply_DTBS/R operation.* [132]

FDP_ACF.1.3/ Supply DTBS/R The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *None.* [133]

[128] [assignment: *access control SFP*]
[129] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[130] [assignment: *access control SFP*]
[131] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]
[132] [assignment*: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[133] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

FDP_ACF.1.4/ Supply DTBS/R	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *None.* [134]

FDP_ACC.1/Signing	Subset access control
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ Signing

The TSF shall enforce the *Signing SFP* [135] on: -
Subjects: Signer
Objects: R.Authorisation Data security attributes, R.Signing_Key_Id and R.DTBS/R of R.Signer and R.Signature.
Operations: Signing:
The Signer instructs the TOE to perform a signature operation containing the following steps:

- *The TOE establishes R.Authorisation_Data for the R.Signing_Key_Id.*
- *The TOE uses the R.Authorisation_Data and R.Signing_Key_Id to activate a signing key in the Cryptographic Module and signs the R.DTBS/R resulting in R.Signature.*
- *The TOE deactivates the signing key when the signature operation is completed.* [136]

ST Application Note 19

The request is signed with the Signer's private key which is validated against the SKA keys authorisation public key for signing. The DTBS/R is provided by the User. The Signer is authorised before every signature so the signing key doesn't stay active after the signing operation is finished.

FDP_ACF.1/Signing	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/ Signing

The TSF shall enforce the *Signing SFP* [137] to objects based on the following: -

1) *Whether the subject is a Signer authorised to create a signature*. [138]

[134] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]
[135] [assignment: *access control SFP*]
[136] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[137] [assignment: *access control SFP*]
[138] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

FDP_ACF.1.2/ Signing	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: - 1) *The R.SAD is verified in integrity.* 2) *The R.SAD is verified that it binds together the Signer authentication, a set of R.DTBS/R and R.Signing_Key_Id.* 3) *The R.DTBS/R used for signature operations is bound to the R.SAD.* 4) *The Signer identified in the SAD is authenticated according to the rules specified in FIA_UAU.5/Signer.* 5) *Only an R.Signing_Key_Id as bound in the SAD, and which is part of the R.Signer security attributes, can be used to create a signature.* [139]
FDP_ACF.1.3/ Signing	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: - 1) *The Signer must be the owner of the R.Signer object used to generate the signature.* [140]
FDP_ACF.1.4/ Signing	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: - 1) *If the Signer does not own the R.Signer object, it can't be used to create a signature.* [141]

FDP_ACC.1/TOE Maintenance Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ TOE Maintenance	The TSF shall enforce the *TOE Maintenance SFP* [142] on: *Subjects: Privileged User **<u>(Admin)</u>*** *Objects: R.TSF_DATA.* *Operations: TOE_Maintenance:* *The Privileged User **<u>(Admin)</u>** transmits information to the TOE to manage R.TSF_DATA.*[143]

FDP_ACF.1/TOE Maintenance Security attribute based access control

[139] [assignment*: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[140] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[141] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[142] [assignment: *access control SFP*]

[143] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/ TOE Maintenance The TSF shall enforce the *TOE Maintenance SFP* [144] to objects based on the following:

1) *Whether the subject is a Privileged User* ***<u>(Admin)</u>*** *authorised to maintain the TOE configuration data.* [145]

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

1) *Only a Privileged User* ***<u>(Admin)</u>*** *who has been authorised to maintain the TOE can carry out the TOE_Maintenance operation.* [146]

FDP_ACF.1.3/ TOE Maintenance The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *None.* [147]

FDP_ACF.1.4/ TOE Maintenance The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *None.* [148]

The TOE can store data in an external repository to meet requirements on, e.g. capacity and redundancy.

FDP_ETC.2/Signer	Export of user data with security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1/ Signer The TSF shall enforce the *Signer Creation SFP*, *Signer Key Pair Generation SFP, Signer Key Pair Deletion SFP, Signer Maintenance SFP, Supply DTBS/R SFP and Signing SFP* [149] when exporting user data, controlled under the SFP(s), outside of the TSF.

FDP_ETC.2.2/ Signer The TSF shall export the user data with the user data's associated security attributes.

[144] [assignment: *access control SFP*]

[145] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

[146] [assignment*: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[147] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[148] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[149] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

FDP_ETC.2.3/ Signer	The TSF shall ensure that the security attributes, when exported outside the TSF, are unambiguously associated with the exported user data.
FDP_ETC.2.4/ Signer	The TSF shall ensure that interpretation of the security attributes of the exported user data is as intended by the owner of the user data
FDP_ETC.2.5/ Signer	The TSF shall enforce the following rules when user data is exported from the TSF: *None*. [150]

ST Application Note 20

The TOE does not export Signer data so this SFR is trivially satisfied.

FDP_IFC.1/Signer	Subset information flow control
Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1/ Signer	The TSF shall enforce the *Signer Flow SFP* [151] on *Privileged User **<u>(User)</u>** and Signer accessing Signer security attributes for all operations.* [152]

FDP_IFF.1/Signer	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/ Signer	The TSF shall enforce the *Signer Flow SFP* [153] based on the following types of subject and information security attributes: - *Privileged User **<u>(User)</u>** and Signer accessing the Signer security attributes.* [154]
FDP_IFF.1.2/ Signer	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: *The TOE shall be initialized with FDP_ACC.1/TOE Maintenance.* *To allow a Signer to sign, the Signer shall be created in the TOE by FDP_ACC.1/Signer Creation followed by FDP_ACC.1/Signer key Pair Generation.*

[150] [assignment: *additional exportation control rules*]
[151] [assignment: *information flow control SFP*]
[152] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]
[153] [assignment: *information flow control SFP*]
[154] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

After Signer is created the following operations can be done: FDP_ACC.1/Signer Key Pair Generation, FDP_ACC.1/Signer Key Pair Deletion, FDP_ACC.1/Supply DTBS/R, FDP_ACC.1/Signer Maintenance and FDP_ACC.1/Signing. [155]

FDP_IFF.1.3/ Signer — The TSF shall enforce the: *None*. [156]

FDP_IFF.1.4/ Signer — The TSF shall explicitly authorise an information flow based on the following rules: *None*. [157]

FDP_IFF.1.5/ Signer — The TSF shall explicitly deny an information flow based on the following rules: *None*. [158]

FDP_ETC.2/Privileged User Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1/ Privileged User — The TSF shall enforce the *Privileged User Creation SFP* [159] when exporting user data, controlled under the SFP(s), outside of the TSF.

FDP_ETC.2.2/ Privileged User — The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3/ Privileged User — The TSF shall ensure that the security attributes, when exported outside the TSF, are unambiguously associated with the exported user data.

FDP_ETC.2.4/ Privileged User — The TSF shall ensure that interpretation of the security attributes of the exported user data is as intended by the owner of the user data.

FDP_ETC.2.5/ Privileged User — The TSF shall enforce the following rules when user data is exported from the TSF: *None.* [160]

ST Application Note 21

The following Privileged User data can be exported from the TOE:

- Data of Admin: none
- Data of User:
  - List of all key names belonging to the User
  - Syslogs (if enabled)

[155] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

[156] [assignment: *additional information flow control SFP rules*]

[157] [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

[158] [assignment: *rules, based on security attributes, that explicitly deny information flows*]

[159] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[160] [assignment: *additional exportation control rules*]

FDP_IFC.1/Privileged user Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/ Privileged User The TSF shall enforce the *Privileged User Flow SFP* [161] on *Privileged User* ***<u>(Admin)</u>*** *accessing Privileged User security attributes for all operations.* [162]

FDP_IFF.1/Privileged User 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/ Privileged User The TSF shall enforce the *Privileged User Flow SFP* [163] based on the following types of subject and information security attributes: - *Privileged User* ***<u>(Admin)</u>*** *accessing the Privileged User security attributes.* [164]

FDP_IFF.1.2/ Privileged User The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: - *The TOE shall be initialized with FDP_ACC.1/TOE Maintenance.* [165]

FDP_IFF.1.3/ Privileged User The TSF shall enforce the: *None*. [166]

FDP_IFF.1.4/ Privileged User The TSF shall explicitly authorise an information flow based on the following rules: *None*. [167]

FDP_IFF.1.5/ Privileged User The TSF shall explicitly deny an information flow based on the following rules: *None*. [168]

FDP_ITC.2/Signer Import of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

[161] [assignment: *information flow control SFP*]

[162] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]

[163] [assignment: *information flow control SFP*]

[164] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

[165] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

[166] [assignment: *additional information flow control SFP rules*]

[167] [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

[168] [assignment: *rules, based on security attributes, that explicitly deny information flows*]

FDP_IFC.1 Subset information flow control]

[FTP_ITC.1 Inter-TSF Trusted channel, or

FTP_TRP.1 Trusted path]

FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1/ Signer	The TSF shall enforce the *Signer Creation SFP, Signer Key Pair Generation SFP, Signer Key Pair Deletion, Signer Maintenance SFP, Supply DTBS/R SFP and Signing SFP* [169] when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2/ Signer	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3/ Signer	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.
FDP_ITC.2.4/ Signer	The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.
FDP_ITC.2.5/ Signer	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *None.* [170]

ST Application Note 22

No Signer data can be imported to the TOE.

FDP_ITC.2/Privileged User	Import of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

[FTP_ITC.1 Inter-TSF Trusted channel, or

FTP_TRP.1 Trusted path]

FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1/ Privileged User	The TSF shall enforce the *Privileged User Creation SFP* [171] when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2/ Privileged User	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3/ Privileged User	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

[169] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[170] [assignment: *additional importation control rules*]

[171] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

FDP_ITC.2.4/ Privileged User — The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/ Privileged User — The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *None.* [172]

ST Application Note 23

There is no Privileged User data imported to the TOE so this SFR is trivially satisfied.

FDP_UCT.1	Basic data exchange confidentiality

Hierarchical to: No other components.

Dependencies: [FTP_ITC.1 Inter-TSF Trusted channel, or FTP_TRP.1 Trusted path]

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

FDP_UCT.1.1 — The TSF shall enforce the *Signer Flow SFP and Privileged User Flow SFP* [173] to *transmit and receive* [174] user data in a manner protected from unauthorised disclosure.

FDP_UIT.1	Data exchange integrity

Hierarchical to: No other components.

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

[FTP_ITC.1 Inter-TSF Trusted channel, or FTP_TRP.1 Trusted path]

FDP_UIT.1.1 — The TSF shall enforce the *Signer Flow SFP and Privileged User Flow SFP* [175] to be able to *transmit and receive* [176] user data in a manner protected from *modification and insertion* [177] errors **for R.Signer and R.Privileged_User and for R.SAD also** *from modification and replay* errors.

FDP_UIT.1.2 — The TSF shall be able to determine on receipt of user data, whether *modification, deletion and insertion* [178] **for R.Signer and R.Privileged_User and for R.SAD** *for modification and replay* has occurred.

[172] [assignment: *additional importation control rules*]

[173] [assignment: *access control SFP(s) and/or information flow control SFP(s)]*

[174] [selection: *transmit, receive]*

[175] [assignment: *access control SFP(s) and/or information flow control SFP(s)]*

[176] [selection: *transmit, receive*]

[177] [selection: *modification, deletion, insertion, replay*]

[178] [selection: *modification, deletion, insertion, replay*]

6.5 Trusted path/channels (FTP)

FTP_TRP.1/Local *Trusted Path*

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1/Local The TSF shall provide a communication path between itself and *local*[179] **client applications**[180] that is logically distinct from other communication paths and provides assured **authentication** [181] of its end points and protection of the communicated data from *modification and disclosure*[182].

FTP_TRP.1.2/Local The TSF shall permit [selection: *the TSF, local client applications*][183] to initiate communication via the trusted path.

FTP_TRP.1.3/Local The TSF shall require the use of the trusted path for *[assignment: services for which trusted path is required]* [184].

ST Application Note 24

The TOE does not provide any interface for local client applications, so this SFR is not applicable and is trivially satisfied.

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*[185] **external client applications** [186] that is logically distinct from other

[179] [selection: *remote, local*]

[180] users

[181] identification

[182] [selection: *modification, disclosure, [assignment: other types of integrity or confidentiality violation]*]

[183] [selection: *the TSF, local users, remote users*]

[184] *initial user authentication, [assignment: other services for which trusted path is required]*]

[185] [selection: *remote, local*]

[186] users

communication paths and provides assured **authentication**[187] of its end points and protection of the communicated data from *modification and disclosure*[188].

FTP_TRP.1.2/External The TSF shall permit *remote* ***external client applications***[189] to initiate communication via the trusted path.

FTP_TRP.1.3/External The TSF shall require the use of the trusted path for *all API commands, and Decanus remote terminal*[190] [191]

ST Application Note 25

The Cryptographic Algorithms table is referenced from FCS_COP.1. The table contains algorithms for securing the channel between the TOE and external entities. KAS for key agreement, KDF for deriving the session key and AESGCM256 to encrypt the messages.

FTP_TRP.1/SSA	Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1/ SSA The TSF shall provide a communication path between itself and: ***Privileged User (User) through SSA*** [192] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification*. [193]

FTP_TRP.1.2/ SSA The TSF shall permit: ***Privileged User (User) through SSA***[194] to initiate communication via the trusted path.

FTP_TRP.1.3/ SSA The TSF shall require the use of the trusted path for: -

~~*1) FDP_ACC.1.1/Privileged User Creation;*~~

2) FDP_ACC.1/Signer Creation;

3) FDP_ACC.1/Signer Maintenance;

4) FDP_ACC.1/Signer Key Pair Generation;

5) FDP_ACC.1/Signer Key Pair Deletion;

[187] identification

[188] [selection: *modification, disclosure, [assignment: other types of integrity or confidentiality violation]*]

[189][selection: *the TSF, remote* ***external client applications***]

[190]*[assignment:services for which trusted path is required]*

[191] [selection: *initial user authentication, [assignment: other services for which trusted path is required]*]

[192] [selection: *remote, local]*

[193] [selection: *modification, disclosure, [assignment: other services for which trusted path is required]]*

[194] [selection: *the TSF, local users, remote users*]

~~*6) FDP_ACC.1/Supply DTBS/R;*~~

~~*7) FDP_ACC.1/TOE Maintenance;*~~

8) None. [195]

ST Application Note 26

The strike-through operations are refinements as those operations are not available for the SSA.

FTP_TRP.1/SIC	Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1/ SIC The TSF shall provide a communication path between itself and: ***Remote Signer through the SIC*** [196] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification*. [197]

FTP_TRP.1.2/ SIC The TSF shall permit: ***Remote Signer through the SIC*** [198] to initiate communication via the trusted path.

FTP_TRP.1.3/ SIC The TSF shall require the use of the trusted path for

1) FDP_ACC.1/Signer Maintenance
2) FDP_ACC.1/Signer Key Pair Generation
3) FDP_ACC.1/Signer Key Pair Deletion
4) FDP_ACC.1/Signing
5) None[199]

ST Application Note 27

The TOE does not verify the SIC as a communication end point and it relies on the signer authentication.

FTP_ITC.1/CM	Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1/ CM The TSF shall provide a communication path between itself and a ***CM certified according to [EN 419221-5]*** that is logically distinct from other communication paths and provides assured authentication of its end points and protection of the communicated data from modification or disclosure.

[195] [assignment: *other services for which trusted path is required*]
[196] [selection: *remote, local]*
[197] [selection: *modification, disclosure, [assignment: other services for which trusted path is required]]*
[198] [selection: *the TSF, local users, remote users*]
[199] [selection: *initial user authentication, [assignment: other services for which trusted path is required]*]

FTP_ITC.1.2/ CM The TSF shall permit the ***TSF and a CM certified according to [EN 419221-5]***[200] to initiate communication via the trusted channel.

FTP_ITC.1.3/ CM The TSF shall initiate communication via the trusted channel for all functions which need a CM. [201]

ST Application Note 28

The TOE itself is a CM certified according to [EN 419221-5] so this SFR is trivially satisfied. In case the TOE is connected to an external SAM the SAM is considered a client application and would be connected via trusted path.

6.6 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.

FPT_TST.1 *Basic TSF Self Testing*

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST.1.1 The TSF shall run a suite of the following self-tests *during initial start-up (or power-on) and at the conditions defined below*[202] to demonstrate the correct operation of the TSF:

- *At initial start-up (or power-on) and repeating all 8 hours:*
  - *Software/firmware integrity test*
  - *Hardware functionality and integrity test*
  - *Random number generator tests*
  - *Cryptographic algorithm tests defined in Cryptographic Self-tests table*
- *Conditional tests defined in Conditional Self-tests table*

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of *TSF data listed in FPT_TST.1.1*.

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of *TSF listed in FPT_TST.1.1*.

[200] [selection: *the TSF, another trusted IT product*]
[201] [assignment: *list of functions for which a trusted channel is required*]
[202] [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*]

ST Application Note 29

The self-tests defined in FPT_TST.1.1 run on every startup and periodically every 8 hours. Authorised users (Admins) can force the tests to be repeated by simply restarting the TOE any time they prefer.

Table 9 Cryptographic Self-tests

Test Target	Description
RSA	*Signature & Encryption Primitive KAT* *PKCS1.5 Padding Signature KAT* *PKCS1.5 Padding Verification KAT* *OAEP Padding PCT*
DSA	*Signature PCT*
AES	*ECB Mode KAT* *CTR Mode KAT*
AES-GCM	*GMAC KAT* *GCM KAT*
3-DES	*ECB Mode KAT*
DH	*Key agreement KAT*
SHS	*SHA-1 KAT* *SHA-2 KAT* *SHA-3 KAT* *SHAKE KAT*
HMAC	*HMAC(SHA-1) KAT* *HMAC(SHA-2) KAT* *HMAC(SHA-3) KAT*
ECDSA	*Signature PCT for all supported curves*
ECDH	*Key agreement KAT for P-256, P-384 and P-521*
EDDSA	*ed22519 KAT* *ed448 KAT*

DRBG	*HMAC-DRBG KAT*
KDF	*SP800-108 KAT* *SP800-56a KAT* *SP800-56c KAT*
PBKDF	*PBKDF2 KAT*
ChaCha20-Poly1305	*ChaCha20 KAT* *Poly1305 KAT* *ChaCha20Poly1305 AEAD KAT*
HSS-LMS	*Signature KAT* *Verification KAT*
XMSS	*Signature KAT* *Verification KAT*
SLH-DSA	*Key generation from seed,* *Signature KAT* *Verification KAT*
ML-DSA	*Key generation from seed,* *signature generation KAT,* *signture verifiation KAT*
ML-KEM	*Key generation from seed,* *key encapsualtion KAT* *key decapsulation KAT*

Table 10 Conditional Self-tests

Test Target	Description
DRBG	*DRBG Continuous Test performed when a random value is requested from the DRBG.* *DRBG tests that previous value is not same as next value (stuck fault test)* *DRBG 11.3 Health checks per SP 800-90A*

DSA	DSA Pairwise Consistency Test performed on every DSA key pair generation. DSA Pairwise Consistency Test performed on every DSA signature calculation.
ML-DSA	Pairwise Consistency Test performed on every ML-DSA key pair generation
SLH-DSA	Pairwise Consistency Test performed on every SLH-DSA key pair generation
HSS-LMS	Pairwise Consistency Test performed on every key pair generation
XMSS	Pairwise Consistency Test performed on every key pair generation
ECDSA	ECDSA Pairwise Consistency Test performed on every ECDSA key pair generation. ECDSA Pairwise Consistency Test performed on every ECDSA signature calculation.
ECDH	ECDH tests if public point is on curve on every ECDH key pair generation.
ML-KEM	encapsulation/decapsulation Pairwise Consistency Test on every ML-KEM key gen
DH	DH tests if the public key is calculated correctly within parameters on every DH key pair generation.
NDRNG	Performed continuously per SP 800-90B Section 4.4.
RSA	RSA Pairwise Consistency Test performed on every RSA key pair generation. RSA Pairwise Consistency Test performed on every RSA signature calculation.
Firmware integrity	RSA 4096 digital signature is validated during firmware load.
Manual Key Entry Test	Confirms the key components entered to decrypt the backup file are correct

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 *Resistance to physical attack*

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1 The TSF shall resist to *remove cover, light detection and Freeze attack with low or high temperatures*[203] to *the entire TOE*[204] by responding automatically such that the SFRs are always enforced.

FPT_FLS.1 *Failure with preservation of secure state*

Hierarchical to: No other components.

Dependencies: No dependencies.

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.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*[205] [206].

FPT_RPL.1 Replay detection

Hierarchical to: No other components.

[203] [assignment: *physical tampering scenarios*]

[204] [assignment: *list of TSF devices/elements*]

[205] [assignment: *list of types of failures in the TSF*]

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

Dependencies: No dependencies.

FPT_RPL.1.1 The TSF shall detect replay for the following entities: *R.SAD*. [207]

FPT_RPL.1.2 The TSF shall perform *reject the signature operation* [208] when replay is detected.

FPT_TDC.1	Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret: -

1) *R.Signer;*
2) *R.Reference_Signer_Authentication_Data;*
3) *R.SAD;*
4) *R.DTBS/R;*
5) *R.SVD;*
6) *R.Privileged_User; and*
7) *R.Reference_Privileged_User_Authentication_Data*
8) *R.TSF_DATA.* [209]

when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use *data integrity either on data or on communication channel* [210] when interpreting the TSF data from another trusted IT product.

6.7 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 recognised by having their ‘Assigned Flag’ attribute set to ‘assigned’), and general keys (keys that have their ‘Assigned Flag’ attribute set to ‘nonassigned’).

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

[207] [assignment: *list of identified entities*]
[208] [assignment: *list of specific actions*]
[209] [assignment: *list of TSF data types*]
[210] [assignment: *list of interpretation rules to be applied by the TSF*]

exported[211]. In particular, an Administrator cannot avoid the need to provide the current authorisation data in order to use such a key, nor can an Administrator change the authorisation 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[212].

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 initialisation and subsequent modification. Additional notes regarding these attributes are as follows:

- key identifier: this shall 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
- authorisation 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. Authorisation data is required only for secret keys
- re-authorisation conditions: the constraints on uses of the key that can be made before reauthorisation is required according to FIA_UAU.6/KeyAuth, and which determines whether a subject is currently authorised to use a key as in FDP_ACF.1/KeyUsage. The types of secret key to which re-authorisation conditions apply, and the details of the re-authorisation 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 this 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 authorisation data can only be changed on successful validation of the current authorisation data – it cannot be changed or reset by an Administrator – and the re-authorisation conditions and key usage attributes cannot be changed; allowed values are referred to in this PP as ‘assigned’ and ‘non-assigned’ but may be mapped to other suitable binary values in TOE implementations.

FMT_SMR.2 *Restrictions on security roles*

Hierarchical to:	FMT_SMR.1 Security Roles
Dependencies:	FIA_UID.1 Timing of identification.

[211] 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 1.3.1.

[212] 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).

FMT_SMR.2.1 The TSF shall maintain the roles: Signer, *Privileged User* ***<u>(Admin, User)</u>***[213].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions *Signer can't be a Privileged User* are satisfied.

ST Application Note 30

Primus HSM supports multiple Administration roles. Genesis for initial startup and configuration of the TOE, Security Officer (SO) for administrative functions during operational state and Partition Security Officer (Partition SO) which is the same as SO but only has access for a specific partition. Partition Auditor which is a limited Partition Security Officer with read only access. All four can be considered Administrator according to the PP terminology.

User represents the client application that invokes the TOE API.

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 authorisation failures*

(2) *Modifying attributes of keys*

(3) *Export and deletion of the audit data, which can take place only under the control of the Administrator role*

(4) *Signer management*

(5) *Privileged User management*

(6) *Configuration management*

(7) *<u>backup and restore functions</u>*[214]

(8) *<u>key import function</u>*[215]

[213] *[assignment: the authorised identified roles]*

[214] *[selection: backup and restore functions, no backup and restore functions]*

[215] *[selection: key import function, no key import function]*

(9) *key export function*[216] [217]

ST Application Note 31

The ST writer merged together the management functions of SAM and CM PP defined in FMT_SMF.1.1.

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*[218] the User accounts[219] to **automatic processes** ~~[assignment: *the authorised identified administrative roles*]~~[220].

ST Application Note 32

As it is defined in FIA_AFL.1/Admin FIA_AFL.1/User and in FIA_AFL.1/Signer there is no need for unblocking. In case of Administrators the administrator accounts are blocked forever and there is no way to unblock them. In case of Users (client application) unblock operation is automatic after a defined time period.

SO can block User (client application) account making them offline and unblock them making them online but as [EN 419221-5] Application Note 38 states **FMT_MTD.1/Unblock** is about unblocking after authorisation failures.

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*[221] the *audit log records*[222] to *the Administrator role*[223].

ST Application Note 33

Audit data within the HSM are in a ring buffer. There is no deletion operation but the oldest records are overwritten when the storage of audit records is full. Audit records can be deleted only by factory reset which is restricted to Administrator role.

FMT_MSA.1/GenKeys *Management of security attributes*

[216] *[selection: key export function, no key export function]*
[217] *[assignment: list of management functions to be provided by the TSF].*
[218] *[selection: change_default, query, modify, delete, clear, [assignment: other operations]]*
[219] [assignment: *list of TSF data*]
[220] refinement: [assignment: *the authorised identified administrative roles*]
[221] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[222] [assignment: *list of TSF data*]
[223] [assignment: *the authorised identified roles*]

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*[224] to restrict the ability to *modify*[225] the security attributes *as specified in the Key Attributes Modification Table*[226] [227] to *subjects, objects, and operations among subjects and General Keys, as specified in the Key Attributes Modification Table*[228] [229].

FMT_MSA.1/AKeys *Management of security attributes*

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*[230] to restrict the ability to *modify* [231] the security attributes *as specified in the Key Attributes Modification Table*[232] [233] to *subjects, objects, and operations among subjects and Assigned Keys specified in the Key Attributes Modification Table*[234] [235].

Table 11 Key Attributes Modification Table[236]

[224] [assignment: *access control SFP(s), information flow control SFP(s)*]

[225] [selection: *change_default, query, modify, delete, [assignment: other operations]*]

[226] [assignment: *list of security attributes , to include attributes as specified in the Key Attributes Modification Table]*

[227] [assignment*: list of security attributes]*

[228] [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]*

[229] [assignment: the authorised identified roles]

[230] [assignment: *access control SFP(s), information flow control SFP(s)*]

[231] [selection: *change_default, query, modify, delete, [assignment: other operations]*]

[232] [assignment: *list of security attributes*]

[233] [assignment: *list of security attributes, to include attributes as specified in the Key Attributes Modification Table]*

[234] [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*]

[235] [assignment: the authorised identified roles]

[236] It is acceptable for a Security Target to specify more restrictive modification conditions than listed in this table, but not to specify less restrictive modification conditions. Where no specific condition is specified (denoted by '---')

Key Attribute (MSA.1)	Assigned Key	Standard SKA key	General Key
Key ID	Cannot be modified	Cannot be modified	Cannot be modified
Key Name	Cannot be modified because modifiable = false	key name can be modified if key flag modifiable == true	can be modified if key flag modifiable == true
Key type	Cannot be modified	Cannot be modified	Cannot be modified
Authorisation Data	Modified only when modification operation includes successful validation of current (pre-modification) authorisation data	Modified only when modification operation includes successful validation of current (pre-modification) authorisation data, or by an Administrator	no authorisation Data
modify flag	Cannot be modified because modifiable = false	can be modified if key flag modifiable == true only from true-> false	can be modified if key flag modifiable == true only from true-> false
Key Usage	Cannot be modified because modifiable = false	can be modified if key flag modifiable == true	can be modified if key flag modifiable == true
imported	Cannot be modified	Cannot be modified	Cannot be modified
Extractable Flag (Export flag according to PP terminology)	Cannot be modified	Cannot be modified	can be modified if key flag modifiable == true
never-extractable (includes import)	true	True	modified by key export operation
Assigned Flag	Cannot be modified (no explicit assigned flag as attribute. is a combination)	not applicable	not applicable

then the Security Target is not constrained by this PP, but clearly the requirements of the system of which the cryptographic module is a part may have more detailed requirements for a specific deployment (i.e. operational environment).

blocked flag	modified by blocked and unblocked authorization	modified by blocked and unblocked authorization	not applicable
destructable flag	Cannot be modified because modifiable = false	can be modified if key flag modifiable == true only from true ->false	can be modified if key flag modifiable == true only from true ->false
Integrity Protection Data	Cannot be modified by users (maintained automatically by TSF)	Cannot be modified by users (maintained automatically by TSF)	Cannot be modified by users (maintained automatically by TSF)

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*[237] to provide *restrictive*[238] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Keys The TSF shall allow the *the authorised identified roles, according to the constraints in the Key Attributes Initialisation Table*[239] to specify alternative initial values to override the default values when an object or information is created.

Table 12 Key Attributes Initialisation Table[82]

Key Attribute (MSA.1)	Assigned Key	standard SKA key	general Key
Key ID	Initialised by generation process	Initialised by generation process	Initialised by generation process
Key type	Initialised by generation process	Initialised by generation process	Initialised by generation process
modify flag	must be initialised with false	Initialised by generation process	Initialised by generation process

[237] [assignment: *access control SFP, information flow control SFP*]

[238] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

[239] [assignment: *the authorised identified roles, according to the constraints in the Key Attributes Initialisation Table*]

Authorisation Data	Initialised by creator during generation	Initialised by creator during generation	Initialised by creator during generation
Key Usage	Initialised by creator during generation	Initialised by creator during generation	Initialised by creator during generation
imported	false, no import possible	false, no import possible	Initialised by generation process
Extractable Flag (Export flag according to PP terminology)	False (i.e. no export allowed)	False (i.e. no export allowed)	Initialised by generation process
never-extractable (includes no import)	true	true	Initialised by generation process (imported or generation)
Assigned Flag	combination of extractable, modify, never-extractable flags	not applicable	not applicable
blocked flag	Initialised by the creator during generation.	Initialised by the creator during generation.	not available
destructible flag	Initialised by creator during generation	Initialised by creator during generation	Initialised by creator during generation
Integrity Protection Data	Initialised automatically by TSF	Initialised automatically by TSF	Initialised automatically by TSF

FMT_MSA.1/Signer Management of security attributes

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/ Signer The TSF shall enforce the: -

1) *Signer Creation SFP* [240] to restrict the ability to *create* [241] the security attributes *listed in FIA_USB.1 for Signer* [242] to *authorised Privileged User* **(User)** [243]*.*

2) *Generate Signer Key Pair SFP* [244] to restrict the ability to *generate* [245] the security attributes *R.SVD and R.Signing_Key_Id* to *authorised Privileged User* **(User)** ***~~and Signer~~*** [246]*.*

3) *Signer Key Pair Deletion SFP* [247] to restrict the ability to *destruct* [248] the security attribute *R.SVD and R.Signing_Key_Id as part of R.Signer* [249] to *authorised Privileged User* **(Admin, User)** and *Signer.* [250]

4) *Supply DTBS/R SFP* [251] to restrict the ability to *create* [252] the security attribute *R.DTBS/R as part of R.Signer* [253] to *authorised Privileged User* **(User)**[254]*.*

5) *Signing SFP* [255] to restrict the ability to *create* [256] the security attribute *R.DTBS/R as part of R.Signer* [257] to *authorised Signer.* [258]

6) *Signing SFP* [259] to restrict the ability to *query* [260] the security attributes *as listed in FIA_USB.1* [261] to *authorised Signer.* [262]

[240] [assignment: *access control SFP(s), information flow control SFP(s)]*
[241] [selection: *change_default, query, modify, delete, [assignment: other operations]]*
[242] [assignment: *list of security attributes*]
[243] [assignment: *the authorised identified roles*]
[244] [assignment: *access control SFP(s), information flow control SFP(s)]*
[245] [selection: *change_default, query, modify, delete, [assignment: other operations]]*
[246] [assignment: *the authorised identified roles*]
[247] [assignment: *access control SFP(s), information flow control SFP(s)]*
[248] [selection: *change_default, query, modify, delete, [assignment: other operations]]*
[249] [assignment: *list of security attributes*]
[250] [assignment: *the authorised identified roles*]
[251] [assignment: *access control SFP(s), information flow control SFP(s)]*
[252] [selection: *change_default, query, modify, delete, [assignment: other operations]]*
[253] [assignment: *list of security attributes*]
[254] [assignment: *the authorised identified roles*]
[255] [assignment: *access control SFP(s), information flow control SFP(s)]*
[256] [selection: *change_default, query, modify, delete, [assignment: other operations]]*
[257] [assignment: *list of security attributes*]
[258] [assignment: *the authorised identified roles*]
[259] [assignment: *access control SFP(s), information flow control SFP(s)]*
[260] [selection: *change_default, query, modify, delete, [assignment: other operations]]*
[261] [assignment: *list of security attributes*]
[262] [assignment: *the authorised identified roles*]

7) *Signer Maintenance SFP* [263] to restrict the ability to *change* [264] the security attributes *R.Reference_Signer_Authentication_Data* [265] to *authorised Privileged User* **(Admin, User)** *and Signer*. [266]

FMT_MSA.1/Privileged User	Management of security attributes

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/ Privileged User The TSF shall enforce the: -

1) *Privileged User Creation SFP* [267] to restrict the ability to *create and query* [268] the security attributes *listed in FIA_USB.1 for Privileged User* [269] to *authorised Privileged User* ***(Admin)***. [270]

FMT_MSA.2	Secure security attributes

Hierarchical to: No other components.

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

FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for *all security attributes listed in FIA_USB.1.* [271]

FMT_MSA.3/Signer	Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

[263] [assignment: *access control SFP(s), information flow control SFP(s)]*

[264] [selection: *change_default, query, modify, delete, [assignment: other operations]]*

[265] [assignment: *list of security attributes*]

[266] [assignment: *the authorised identified roles*]

[267] [assignment: *access control SFP(s), information flow control SFP(s)]*

[268] [selection: *change_default, query, modify, delete, [assignment: other operations]]*

[269] [assignment: *list of security attributes*]

[270] [assignment: *the authorised identified roles*]

[271] [assignment: *list of security attributes*]

FMT_MSA.3.1/ Signer — The TSF shall enforce the *Signer Creation SFP* [272] to provide *restrictive* [273] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/ Signer — The TSF shall allow the *Privileged User* **(User)** [274] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3/Privileged User	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/ Privileged User — The TSF shall enforce the *Privileged User Creation SFP* [275] to provide *restrictive* [276] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/ Privileged User — The TSF shall allow the *Privileged User* **(Admin)** [277] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1	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 — The TSF shall restrict the ability to: -

1) *modify* [278] the *R.TSF_DATA data* [279] to *Privileged User* **(Admin)**. [280]

6.8 Security audit data generation (FAU)

FAU_GEN.1 *Audit data generation*

[272] [assignment: *access control SFP, information flow control SFP*]

[273] [*selection, choose one of: restrictive, permissive, [assignment: other property]]*

[274] [assignment: *the authorised identified roles*]

[275] [assignment: *access control SFP, information flow control SFP*]

[276] [*selection, choose one of: restrictive, permissive, [assignment: other property]]*

[277] [assignment: *the authorised identified roles*]

[278] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]]*

[279] [assignment: *list of TSF data*]

[280] [assignment: *the authorised identified roles*]

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*[281] level of audit; and[282]

c) *Startup of the TOE;*

d) *Shutdown of the TOE*

e) *Cryptographic key generation (FCS_CKM.1); (including Signing keys)*

f) *Timing and event of cryptographic key destruction (FCS_CKM.6); (including Signing keys)*

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

h) *Authentication and authorisation failure handling (FIA_AFL.1): all unsuccessful authentication or authorisation attempts, the reaching of the threshold for the unsuccessful authentication or authorisation 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.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) *Privileged User management;*

t) *Privileged User authentication;*

u) *Signer management;*

[281] [selection, choose one of: minimum,basic, detailed, not specified]

[282] Levels of audit are not required to be defined in the Security Target.

v) *Signer authentication;*

w) *Signing key activation and usage including the hash of the DTBS/R(s); and R.Signature;*

x) *Change of TOE configuration;*

y) *none*[283].

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*[284].

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.3 *Guarantees of audit data availability*

Hierarchical to: FAU_STG.2 Protected audit data storage.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.3.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.3.2 The TSF shall be able to *prevent*[285] unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.3.3 The TSF shall ensure that *all*[286] stored audit records will be maintained when the following conditions occur: *audit storage exhaustion*[287].

ST Application Note 34

[283] [assignment: *other specifically defined auditable events*]

[284] [assignment: *other audit relevant information*]

[285] [selection, choose one of: *prevent, detect*]

[286] [assignment: *metric for saving audit records*]

[287] [selection: *audit storage exhaustion, failure, attack*]

Internal audit logs can be collected by Administrators and exported to external drives via USB or WebDAV. The internal audit storage stores records cyclically, deleting the oldest records when the storage is full so this is the Administrators responsibility to backup the audit logs in time. Deletion of the logs is not possible even for the Administrators. Audit logs are deleted only in the case of Factory Reset. It is also possible to configure a Syslog Server in the HSM so the logs can be exported automatically to the Syslog server so the cyclic internal storing is not a problem. In case of using an external Syslog server the communication is initialised by the HSM and there is only outgoing communication.

7 Security Assurance Requirements

The security assurance requirement level is **EAL4** augmented with **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), it is unlikely that physical attacks will be within the scope of an evaluation against this PP.

Table 13 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)
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**)

7.1 Refinements of Security Assurance Requirements

The refinements of Security Assurance Requirements can be found in [EN 419221-5].

8 Rationales

8.1 Rationale for [EN 419221-5]

8.1.1 Security Objectives Rationale

8.1.1.1 Security Objectives Coverage

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

Table 14 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.FailuraDetect	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 unauthorised purpose, and is addressed by the requirement in OT.Auth for the TOE to carry out an authorisation check before using a secret key. OT.KeyUseConstraint expands on this to set out requirements for the granularity of authorisation.

T.KeyOveruse is concerned with the possibility that more uses may be made of an authorised 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-authorisation 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 unauthorised 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 unauthorised 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 authorised (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 is directly matched by a security objective for the operational environment in section 5.2. The wording of each objective for the operational environment includes the wording of each assumption, and no further rationale is therefore given here.

8.1.3 Security Requirements Rationale

8.1.3.1 Security Requirements Coverage

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

Table 15 TOE Security Objectives mapping to SFRs

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstrain	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FCS_CKM.1		X												
FCS_CKM.6	X													
FCS_COP.1		X												
FCS_RNG.1											X			
FIA_UID.1				X										
FIA_UAU.1				X										
FIA_AFL.1/Admin				X										
FIA_AFL.1/User				X										
FIA_AFL.1/Signer				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.1													X	
FPT_PHP.1												X		
FPT_PHP.3												X		
FPT_FLS.1													X	
FMT_SMR.2				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.3														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.6 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 and the use of an appropriate random number generator in FCS_CKM.1. Note that the refinements to assurance components in [EN419221-5] section 9.5.2 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 in [EN419221-5]).

OT.Auth is addressed by FIA_UID.1, FIA_UAU.1, FIA_AFL.1/Admin, FIA_AFL.1/User and FIA_AFL.1/Signer for administrator authentication (with FMT_MTD.1/Unblock and its dependencies on FMT_SMR.2 and FMT_SMF.1 ensuring that appropriate roles and unblocking for authorisation and authentication failures are also provided). Authorisation for external client applications is provided by the requirements for authentication of endpoints in FTP_TRP.1/Local and FTP_TRP.1/External. Authorisation 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 initialised) 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-authorisation 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.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.3, 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.2.

8.2 Rationale for [EN 419241-2]

8.2.1 Security Objectives Coverage

Table 16 TOE Security objectives (Enrolment) and threats

	Enrolment	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD
Enrolment					
T.ENROLMENT_SIGNER_IMPERSONATION		X	X		
T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED		X	X		
T.SVD_FORGERY				X	X
Signer Management					
T.ADMIN_IMPERSONATION					
T.MAINTENANCE_AUTHENTICATION_DISCLOSE			X		
Usage					
T.AUTHENTICATION_SIGNER_IMPERSONATION					
T.SIGNER_AUTHENTICATION_DATA_MODIFIED			X		

8.2.2 Rationale for the security objectives

8.2.2.1 General

This section provides a rationale objectives covers each threat, organizational security policy and assumption.

8.2.2.2 Threats and objectives

T.ENROLMENT_SIGNER_IMPERSONATION is covered by OT.SIGNER_PROTECTION requiring R.Signer to be protected in integrity and for sensitive parts in confidentiality. It is also covered by OT.SIGNER_MANAGEMENT requiring the signer to be securely created. It is also covered by OT.REFERENCE_SIGNER_AUTHENTICATION_DATA requiring the TOE to be able to assign signer authentication data to the signer. It is also covered by OE.TW4S_CONFORMANT as that requires signer enrolment to be handled in accordance with [Assurance] for level at least substantial.

T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED is covered by OT.REFERENCE_SIGNER_AUTHENTICATION_DATA requiring that authentication data be securely handled. It is also covered by OT.SIGNER_PROTECTION requiring that the attributes, including signer authentication data, be protected in integrity and if needed in confidentiality. It is also covered by OE.SIGNER_AUTHENTICATION_DATA requiring the signer to keep his authentication data secret. It is also covered by OE.DEVICE requiring the device used by the signer not to disclose authentication data.

T.SVD_FORGERY is covered by OT.SIGNER_KEY_PAIR_GENERATION requiring a Cryptographic Module to generate signer key pair. It is also covered by OT.SVD requiring the SVD to be protected while inside the TOE. It is also covered by OT.CRYPTO requiring the usage of endorsed algorithms. It is also covered by OE.SVD_AUTHENTICITY requiring the environment to protect the SVD during transmit from the TOE to the CA. It is also covered by OE.CA_REQUEST_CERTIFICATE requiring the certification request to be protected in integrity.

T.ADMIN_IMPERSONATION is covered by OT.SIGNER_MANAGEMENT and OT.PRIVILEGED_USER_AUTHENTICATION requiring any changes to the signer representation and attributes are carried out in an authorized manner.

T.MAINTENANCE_AUTHENTICATION_DISCLOSE is covered by OT.REFERENCE_SIGNER_AUTHENTICATION_DATA requiring that authentication data be securely handled.

T.AUTHENTICATION_SIGNER_IMPERSONATION is covered by OT.SAD_VERIFICATION requiring that the TOE checks the SAD received in the SAP.

T.SIGNER_AUTHENTICATION_DATA_MODIFIED is covered by OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION requiring the SAD transported protected in the SAP. It is also covered by OT.REFERENCE_SIGNER_AUTHENTICATION_DATA requiring that authentication data be securely handled. It is also covered by OT.SAP requiring the integrity of the SAD is protected during transmit in the SAP.

T.SAP_BYPASS is covered by OT.SAP requiring that all steps, including SAD verification, of the SAP shall completed. It is also covered by OE.DEVICE requiring the SIC to participate the in SAP.

T.SAP_REPLAY is covered by OT.SAP requiring that the signature activation protocol shall be able to resist whole or part of it being replayed. It is also covered by OE.DEVICE requiring the SIC to participate the in SAP.

T.SIGNATURE_REQUEST_DISCLOSURE is covered by the OT.SAP requiring the protocol to be able to transmit data securely.

T.SAD_FORGERY is covered by OT.SAP requiring the TOE to be able to detect if the SAD has been modified during transmit to the TOE. It is also covered by OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION requiring signature authentication data to be protected during transmit to the TOE. It is also covered by OE.SIGNER_AUTHENTICATION_DATA requiring the signer to protect his authentication data. It is also covered by OE.DEVICE requiring the device used by the signer to participate correctly in the SAP, in particular the device shall not disclose authentication data.

T.DTBSR_FORGERY is covered by OT.DTBSR_INTEGRITY requiring the R.DTBS/R to be protected in integrity during ransmit to the TOE. It is also covered by OE.DEVICE requiring the SIC to participate the in SAP.

T.SIGNATURE_FORGERY is covered by OT.SIGNATURE_INTEGRITY requiring that the signature is protected in integrity inside the TOE. It is also covered by OT.CRYPTO requiring the usage of endorsed algorithms.

T.PRIVILEGED_USER_INSERTION is covered by OT.PRIVILEGED_USER_MANAGEMENT requiring only Privileged User can create new R.Privileged_User and OT.PRIVILEGED_USER_AUTHENTICATION that requires a Privileged User to be authenticated.

T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION is covered by OT.PRIVILEGED_USER_MANAGEMENT requiring only Privileged User can modify R.Privileged_User and OT.PRIVILEGED_USER_AUTHENTICATION that requires a Privileged User to be authenticated. It is also covered by OT.PRIVILEGED_USER_PROTECTION requiring the Privileged User to be protected in integrity.

T.AUTHORISATION_DATA_UPDATE is covered by OT.SYSTEM_PROTECTION requiring any unauthorised modification to TOE configuration to be detectable.

T.AUTHORISATION_DATA_DISCLOSE is covered by OT.SYSTEM_PROTECTION requiring any unauthorised modification to TOE configuration to be detectable.

T.CONTEXT_ALTERATION is covered by OT.SYSTEM_PROTECTION requiring any unauthorised modification to TOE configuration to be detectable.

T.AUDIT_ALTERATION is covered by OT.AUDIT_PROTECTION requiring any audit modification can be detected.

T.RANDOM is covered by OT.RANDOM requiring that random numbers are not predictable and have sufficient entropy.

8.2.2.3 Organizational security policies and objectives

OSP.RANDOM is covered by OT.RANDOM requiring that random numbers are not predictable and have sufficient entropy.

OSP.CRYPTO is covered by OT.CRYPTO requiring the usage of endorsed algorithms and OE.CRYPTOMODULE_CERTIFIED requiring a cryptographic module to provide a tamper protected environment and for cryptographic functionality and random number generation.

8.2.2.4 Assumptions and objectives

A.PRIVILEGED_USER is covered by OE.TW4S_CONFORMANT which requires that the system where the TOE operates is compliant with [EN 419241-1] where Clause SRG_M.1.8 requires that administrators are trained.

A.SIGNER_ENROLMENT is covered by OE.ENV requiring the TSP to be audited.

A.SIGNER_AUTHENTICATION_DATA_PROTECTION is covered by OE.SIGNER_AUTHENTICATION_DATA requiring the signer to protect his authentication data.

A.SIGNER_DEVICE is covered by OE.DEVICE requiring the signer’s device to be protected against malicious code.

A.CA is covered by OE.CA_REQUEST_CERTIFICATE requiring that the CA will issue certificates containing the SVD.

A.ACCESS_PROTECTED is covered by OE.ENV requiring the TOE be operated in an environment with physical access controls.

A.AUTH_DATA are covered by OE.DEVICE requiring the device to participate correctly in the SAP.

A.TSP_AUDITED is covered by OE.ENV requiring that the TOE is operated by a qualified TSP.

A.SEC_REQ is covered by OE.TW4S_CONFORMANT requiring the system where the TOE operates is compliant with [EN 419241-1].

Table 17 TOE Security objectives (Signer Management and System) and threats

	User Management	OT.PRIVILEGED_USER_MANAGEMENT	OT.PRIVILEGED_USER_AUTHENTICATION	OT.PRIVILEGED_USER_PROTECTION	OT.SIGNER_MANAGEMENT	System	OT.RANDOM	OT.SYSTEM_PROTECTION	OT.AUDIT_PROTECTION
Enrolment									
T.ENROLMENT_SIGNER_IMPERSONATION					X				
T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED									
T.SVD_FORGERY									
Signer Management									
T.ADMIN_IMPERSONATION			X		X				
T.MAINTENANCE_AUTHENTICATION_DISCLOSE									
System									
T.PRIVILEGED_USER_INSERTION		X	X						
T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION		X	X	X					
T.AUTHORISATION_DATA_UPDATE								X	
T.AUTHORISATION_DATA_DISCLOSE								X	
T.CONTEXT_ALTERATION								X	
T.AUDIT_ALTERATION									X
T.RANDOM							X		

Table 18 TOE Security objectives (Usage) and threats

	Usage	OT.SAD_VERIFICATION	OT.SAP	OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION	OT.DTBSR_INTEGRITY	OT.SIGNATURE_INTEGRITY	OT.CRYPTO
Enrolment							
T.ENROLMENT_SIGNER_IMPERSONATION							
T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED							
T.SVD_FORGERY							X
Signer Management							
T.ADMIN_IMPERSONATION							
T.MAINTENANCE_AUTHENTICATION_DISCLOSE							
Usage							
T.AUTHENTICATION_SIGNER_IMPERSONATION		X					
T.SIGNER_AUTHENTICATION_DATA_MODIFIED			X	X			
T.SAP_BYPASS			X				
T.SAP_REPLAY			X				
T.SAD_FORGERY			X	X			
T.SIGNATURE_REQUEST_DISCLOSURE			X				
T.DTBSR_FORGERY					X		
T.SIGNATURE_FORGERY						X	X

Table 19 TOE Security Objectives and Organizational Security Policies

	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD	OT.RANDOM	OT.CRYPTO
OSP.RANDOM					X	
OSP.CRYPTO						X

Table 20 Threats and Security Objectives for the environment

	OE.SVD_AUTHENTICITY	OE.CA_REQUEST_CERTIFICATE	OE.CERTIFICATE_VERIFICATION	OE.SIGNER_AUTHENTICATION_DATA	OE.DELEGATED_AUTHENTICATION	OE.DEVICE	OE.ENV	OE.CRYPTOMODULE_CERTIFIED	OE.TW4S_CONFORMANT
Enrolment									
T.ENROLMENT_SIGNER_IMPERSONATION									X
T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED				X		X			
T.SVD_FORGERY	X	X							
Usage									
T.AUTHENTICATION_SIGNER_IMPERSONATION									
T.SIGNER_AUTHENTICATION_DATA_MODIFIED									
T.SAP_BYPASS						X			
T.SAP_REPLAY						X			
T.SAD_FORGERY				X		X			
T.SIGNATURE_REQUEST_DISCLOSURE									
T.DTBSR_FORGERY						X			
T.SIGNATURE_FORGERY									

Table 21 Organizational Security Policies, Assumptions and Security Objectives for the environment

	OE.SVD_AUTHENTICITY	OE.CA_REQUEST_CERTIFICATE	OE.CERTIFICATE_VERIFICATION	OE.SIGNER_AUTHENTICATION_DATA	OE.DELEGATED_AUTHENTICATION [288]	OE.DEVICE	OE.ENV	OE.CRYPTOMODULE_CERTIFIED	OE.TW4S_CONFORMANT
Organisational Security Policies									
OSP.RANDOM									
OSP.CRYPTO								X	
Assumptions									
A.PRIVILEGED_USER									X
A.SIGNER_ENROLMENT							X		
A.SIGNER_AUTHENTICATION_DATA_PROTECTION				X					
A.SIGNER_DEVICE						X			
A.CA		X							
A.ACCESS_PROTECTED							X		
A.AUTH_DATA						X			
A.TSP_AUDITED							X		
A.SEC_REQ									X
A.CERTIFICATION_AUTHORITY			X						

8.2.3 Security Requirements Rationale

8.2.3.1 Security Requirements Coverage

The following table is used to demonstrate that every SFR is used to cover an objective and that every objective is covered by an SFR.

[288]

Table 22 security requirements coverage

	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD	OT.PRIVILEGED_USER_MANAGEMENT	OT.PRIVILEGED_USER_AUTHENTICATION	OT.PRIVILEGED_USER_PROTECTION	OT.SIGNER_MANAGEMENT	OT.SYSTEM_PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION	OT.DTBSR_INTEGRITY	OT.SIGNATURE_INTEGRITY	OT.CRYPTO	OT.RANDOM
Security Audit																	
FAU_GEN.1										X							
FAU_GEN.2										X							
Cryptographic Support																	
FCS_CKM.1			X													X	
FCS_CKM.6			X														
FCS_COP.1			X												X	X	
FCS_RNG.1			X														X
User Data Protection																	
FDP_ACC.1/Privileged User Creation					X												
FDP_ACF.1/Privileged User Creation					X												
FDP_ACC.1/Signer Creation		X						X									
FDP_ACF.1/Signer Creation		X						X									

	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD	OT.PRIVILEGED_USER_MANAGEMENT	OT.PRIVILEGED_USER_AUTHENTICATION	OT.PRIVILEGED_USER_PROTECTION	OT.SIGNER_MANAGEMENT	OT.SYSTEM_PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION	OT.DTBSR_INTEGRITY	OT.SIGNATURE_INTEGRITY	OT.CRYPTO	OT.RANDOM
FDP_ACF.1/Signer Maintenance		X															
FDP_ACC.1/Signer Maintenance		X															
FDP_ACC.1/Signer Key Pair Generation			X	X													
FDP_ACF.1/Signer Key Pair Generation			X	X													
FDP_ACC.1/Signer Key Pair Deletion								X									
FDP_ACF.1/Signer Key Pair Deletion								X									
FDP_ACC.1/Supply DTBS/R														X			
FDP_ACF.1/Supply DTBS/R														X			
FDP_ACC.1/Signing											X				X		
FDP_ACF.1/Signing											X				X		
FDP_ACC.1/TOE Maintenance									X								
FDP_ACF.1/TOE Maintenance									X								

	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD	OT.PRIVILEGED_USER_MANAGEMENT	OT.PRIVILEGED_USER_AUTHENTICATION	OT.PRIVILEGED_USER_PROTECTION	OT.SIGNER_MANAGEMENT	OT.SYSTEM_PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION	OT.DTBSR_INTEGRITY	OT.SIGNATURE_INTEGRITY	OT.CRYPTO	OT.RANDOM
FDP_ETC.2/Signer	X																
FDP_IFC.1/Signer	X																
FDP_IFF.1/Signer	X																
FDP_ETC.2/Privileged User					X		X										
FDP_IFC.1/Privileged User					X		X										
FDP_IFF.1/Privileged User					X		X										
FDP_ITC.2/Signer	X																
FDP_ITC.2/Privileged User					X		X										
FDP_UCT.1	X																
FDP_UIT.1	X																
Identification and Authentication																	
FIA_AFL.1/Admin						X					X						
FIA_AFL.1/Signer											X						
FIA_ATD.1	X				X		X										
FIA_UAU.1						X					X						
FIA_UAU.5/Signer											X						

	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD	OT.PRIVILEGED_USER_MANAGEMENT	OT.PRIVILEGED_USER_AUTHENTICATION	OT.PRIVILEGED_USER_PROTECTION	OT.SIGNER_MANAGEMENT	OT.SYSTEM_PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION	OT.DTBSR_INTEGRITY	OT.SIGNATURE_INTEGRITY	OT.CRYPTO	OT.RANDOM
FIA_UAU.5/Privileged User						X											
FIA_UID.2					X		X	X									
FIA_USB.1	X		X		X		X										
Security Management																	
FMT_MSA.1/Signer								X									
FMT_MSA.1/Privileged User					X			X									
FMT_MSA.2					X			X									
FMT_MSA.3/Signer								X									
FMT_MSA.3/Privileged User					X			X									
FMT_MTD.1									X								
FMT_SMF.1									X								
FMT_SMR.2									X								
Protection of the TSF																	
FPT_PHP.1									X								
FPT_PHP.3									X								
FPT_RPL.1												X					

	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD	OT.PRIVILEGED_USER_MANAGEMENT	OT.PRIVILEGED_USER_AUTHENTICATION	OT.PRIVILEGED_USER_PROTECTION	OT.SIGNER_MANAGEMENT	OT.SYSTEM_PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION	OT.DTBSR_INTEGRITY	OT.SIGNATURE_INTEGRITY	OT.CRYPTO	OT.RANDOM
FPT_STM.1										X							
FPT_TDC.1	X				X												
Trusted Path/Channels																	
FTP_TRP.1/SSA									X					X			
FTP_TRP.1/SIC												X	X	X			
FTP_ITC.1/CM			X												X		

OT.SIGNER_PROTECTION is handled by requirements export and import of R.Signer in a secure way. (FDP_ETC.2/Signer, FDP_IFC.1/Signer, FDP_IFF.1/Signer, FDP_ITC.2/Signer, FDP_UCT.1 FDP_UIT.1 and FPT_TDC.1). The actual description of the data is described in FIA_ATD.1 and FIA_USB.1.

OT.REFERENCE_SIGNER AUTHENTICATION_DATA is handled by FDP_ACC.1/Signer Creation, FDP_ACF.1/Signer Creation, FDP_ACC.1/Signer Maintenance and FDP_ACF.1/Signer Maintenance, which describes access control for creating and updating R.Signer and R.Reference_Signer_Authentication_Data.

OT.SIGNER_KEY_PAIR_GENERATION is handled by the requirements for key generation and cryptographic algorithms in FCS_CKM.1, FCS_COP.1. FCS_RNG.1 provides a random source for key generation. FCS_CKM.6 describes the requirements for key destruction. FDP_ACC.1/Signer Key Pair Generation and FDP ACF.1/Signer Key Pair Generation describes access control for creating a key pair. FIA_USB.1 describes that R.Signing_Key_Id is associated with Signer. FTP_ITC.1/CM can be used to communicate securely with a CM.

OT.SVD is handled by the requirements in FDP_ACC.1/Signer Key Pair Generation and FDP_ACF.1/Signer Key Pair Generation.

OT.PRIVILEGED_USER_MANAGEMENT is handled by requirements for export and import of R.Privileged User in a secure way (FDP_ETC.2/Privileged User, FDP_IFC.1/Privileged User, FDP_IFF.1/Privileged User, FDP_ITC.2/Privileged User and FPT_TDC.1). The actual description of the data is described in FIA_ATD.1 and FIA_USB.1. Authentication of Privileged User is handled by FIA_UID.2, FMT_MSA.1/Privileged User, FMT_MSA.2 and FMT_MSA.3/Privileged User. FDP_ACC.1/Privileged User Creation and FDP_ACF.1/Privileged User Creation describes access controls for creating Privileged Users.

OT.PRIVILEGED_USER_AUTHENTICATION is handled by FIA_AFL.1/User, FIA_AFL.1/Admin, FIA_UAU.1 and FIA_UAU.5/Privileged User.

OT.PRIVILEGED_USER_PROTECTION is handled by requirements export and import of Privileged User in a secure way. FDP_ETC.2/Privileged User, FDP_IFC.1/Privileged User, FDP_IFF.1/Privileged User, FDP_ITC.2/Privileged User, FIA_UID.2. The actual description of the data is described in FIA_ATD.1 and FIA_USB.1. FIA_UID.2 ensures that Privileged Users are authenticated they can carry out any operation

OT.SIGNER_MANAGEMENT is handled by the requirements for access control in FDP_ACC.1/Signer Creation, FDP_ACF.1/Signer Creation, FDP_ACC.1/Signer Key Pair Deletion and FDP_ACF.1/Signer Key Pair Deletion. Authentication of Signers and Privileged Users are handled by FIA_UID.2, FMT_MSA.1/Signer, FMT_MSA.1/Privileged User, FMT_MSA.2, FMT_MSA.3/Signer and FMT_MSA.3/Privileged User.

OT.SYSTEM_PROTECTION is handled by FMT_MTD.1, FMT_SMF.1 and FMT_SMR.2. FDP_ACC.1/TOE Maintenance and FDP_ACF.1/TOE Maintenance describes access control rules for managing TSF data. FPT_PHP.1 and FPT_PHP.3 describes requirements for TSF protection. FTP_TRP.1/SSA describes that only a Privileged User can maintain a TOE.

OT.AUDIT_PROTECTION is handled by the requirements for audit record generation FAU_GEN.1 and FAU_GEN.2 using reliable time stamps in FPT_STM.1.

OT.SAD_VERIFICATION is handled by the FIA_AFL.1/User, FIA_AFL.1/Admin, FIA_AFL.1/Signer, FIA_UAU.1 and FIA_UAU.5/Signer. FDP_ACC.1/Signing and FDP_ACF.1/Signing describes access control rules for the signature operation and well as for SAP verification.

OT.SAP is covered by the requirements FTP_TRP.1/SIC and FPT_RPL.1 the protocol between the SIC and TSF.

OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION is covered by FTP_TRP.1/SIC, which describes the requirements for data transmitted to the TOE, is protected in integrity.

OT.DTBSR_INTEGRITY is covered by FTP_TRP.1/SSA and FTP_TRP.1/SIC requiring data transmission to be protected in integrity. Also covered by access control rules FDP_ACC.1/Supply DTBS/R and FDP_ACF.1/Supply DTBS/R for transmitting DTBS/R to the TSF.

OT.SIGNATURE_INTEGRITY is handled by FCS_COP.1, which describes requirements for algorithms. FTP_ITC.1/CM may be used to transmit data securely between the TOE and the CM. Access control for the signature operation is ensured by FDP_ACC.1/Signing and FDP_ACF.1/Signing.

OT.CRYPTO is covered by FCS_CKM.1, FCS_COP.1, which describes requirements for key generation and algorithms.

OT.RANDOM is handled by FCS_RNG.1, which describes requirement on the random number generation.

8.3 SFR Dependencies

Table 23 SFR Dependencies

Requirement	Dependencies	Fulfilled by
FCS_CKM.1	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1] FCS_CKM.3 [FCS_RBG.1 or FCS_RNG.1] FCS_CKM.6	FCS_COP.1 FCS_CKM.3 is irrelevant FCS_RNG.1 FCS_CKM.6
FCS_CKM.6	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FDP_ITC.2/Signer, FDP_ITC.2/Privileged User, FCS_CKM.1
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.3 Cryptographic key access	FDP_ITC.2/Signer, FDP_ITC.2/Privileged User, FCS_CKM.1, FCS_CKM.3 is irrelevant
FCS_RNG.1	None	No dependents
FIA_UID.1	None	No dependents
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_AFL.1/Admin	FIA_UAU.1	FIA_UAU.1
FIA_AFL.1/User	FIA_UAU.1	FIA_UAU.1
FIA_AFL.1/Signer	FIA_UAU.1	FIA_UAU.1
FIA_UAU.6/KeyAuth	None	No dependents
FIA_ATD.1	None	No dependents
FIA_UAU.5/Signer	None	No dependents
FIA_UAU.5/Privileged User	None	No dependents
FIA_UID.2	None	No dependents
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FDP_IFC.1/KeyBasics	FDP_IFF.1	FDP_IFF.1/KeyBasics
FDP_ACC.1/KeyUsage	FDP_ACF.1	FDP_ACF.1/KeyBasics
FDP_ACF.1/KeyBasics	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, FMT_MSA.3/Keys
FDP_SDI.2	None	No dependents
FDP_RIP.1	None	No dependents

Requirement	Dependencies	Fulfilled by
FDP_ACC.1/Privileged User Creation	FDP_ACF.1	FDP_ACF.1/Privileged User Creation
FDP_ACF.1/ Privileged User Creation	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1/Privileged User Creation FMT_MSA.3/Keys
FDP_ACC.1/Signer Creation	FDP_ACF.1	FDP_ACF.1/Signer Creation
FDP_ACF.1/Signer Creation	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1/Signer Creation FMT_MSA.3/Keys
FDP_ACC.1/Signer Maintenance	FDP_ACF.1	FDP_ACF.1/Signer Maintenance
FDP_ACF.1/Signer Maintenance	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1/Signer Maintenance FMT_MSA.3/Keys
FDP_ACC.1/Signer Key Pair Generation	FDP_ACF.1	FDP_ACF.1/Signer Key Pair Generation
FDP_ACF.1/Signer Key Pair Generation	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1/Signer Key Pair Generation FMT_MSA.3/Keys
FDP_ACC.1/Signer Key Pair Deletion	FDP_ACF.1	FDP_ACF.1/Signer Key Pair Deletion
FDP_ACF.1/Signer Key Pair Deletion	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1/Signer Key Pair Deletion FMT_MSA.3/Keys
FDP_ACC.1/Signing	FDP_ACF.1	FDP_ACF.1/Signing
FDP_ACF.1/Signing	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1/Signing FMT_MSA.3/Keys
FDP_ACC.1/TOE Maintenance	FDP_ACF.1	FDP_ACF.1/TOE Maintenance
FDP_ACF.1/TOE Maintenance	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1/TOE Maintenance FMT_MSA.3/Keys
FDP_IFC.1/Signer	FDP_IFF.1	FDP_IFF.1/Signer
FDP_IFF.1/Signer	FDP_IFC.1, FMT_MSA.3	FDP_IFC.1/Signer FMT_MSA.3/Keys
FDP_ETC.2/Privileged User	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1/Privileged User

Requirement	Dependencies	Fulfilled by
FDP_IFC.1/Privileged User	FDP_IFF.1	FDP_IFF.1/Privileged User
FDP_IFF.1/Privileged User	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/Privileged User FMT_MSA.3/Keys
FDP_UCT.1	[FTP_ITC.1 or FTP_TRP.1] [FDP_ACC.1 or FDP_IFC.1]	FTP_TRP.1/*, FDP_ACC.1/*, FDP_IFC/*[289]
FDP_UIT.1	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	FDP_ACC.1/*, FDP_IFC/*, FTP_TRP.1/*
FTP_TRP.1/External	None	No dependents
FTP_TRP.1/SSA	None	No dependents
FPT_STM.1	None	No dependents
FPT_TST.1	None	No dependents
FPT_PHP.1	None	No dependents
FPT_PHP.3	None	No dependents
FPT_FLS.1	None	No dependents
FPT_RPL.1	None	No dependents
FPT_TDC.1	None	No dependents
FMT_SMR.2	FIA_UID.1	FIA_UID.1, FIA_UID.2
FMT_SMF.1	None	No dependents
FMT_MTD.1/Unblock	FMT_SMR.1 FMT_SMF.1	FMT_SMR.2 FMT_SMF.1
FMT_MTD.1/AuditLog	FMT_SMR.1 FMT_SMF.1	FMT_SMR.2 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/Signer, FMT_SMR.2 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/Signer, FMT_SMR.2 FMT_SMF.1

[289] /* means all iteration of the SFR.

Requirement	Dependencies	Fulfilled by
FMT_MSA.3/Keys	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/* FMT_SMR.2
FMT_MSA.1/Signer	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage, FDP_IFC.1/Signer, FMT_SMR.2 FMT_SMF.1
FMT_MSA.1/Privileged User	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage, FDP_IFC.1/Privileged User, FMT_SMR.2 FMT_SMF.1
FMT_MSA.2	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	FDP_ACC.1/* FDP_IFC.1/* FMT_MSA.1/* FMT_SMR.2
FMT_MSA.3/Signer	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/* FMT_SMR.2
FMT_MSA.3/Privileged User	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/* FMT_SMR.2
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	FMT_SMR.2 FMT_SMF.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 FIA_UID.2
FAU_STG.3	FAU_GEN.1	FAU_GEN.1

8.4 Rationale for SARs

The assurance level for this ST is **EAL4 augmented with AVA_VAN.5**.

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialised 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 security target is just such a product. Augmentation results from the selection of **AVA_VAN.5**. All the dependencies of AVA_VAN.5 are satisfied by other assurance components in the EAL4 assurance package.

8.5 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.

9 TOE Summary Specification

This section describes how the TOE meets each SFR.

9.1 Authorisation

The TOE requires the identification and authentication before giving access to any security relevant function. There are five different roles in Primus HSM. Genesis, Security Officer, Partition Security Officer, Partition Auditor and User (client application). Genesis, Security Officer (SO) and Partition Security Officer (Partition SO) and Partition Auditors are considered the Administrators of the TOE. Users represent the remote client applications accessing the TOE via its API.

Administrators

The Administrators (Genesis, SO Partition SO and Partition Auditors) authenticate themselves using their smart cards and PINs. In some types of the TOE (E-Series) the Administrators are using their 'virtual' cards but the authentication/authorisation process is the same. The operator inserts a Card and provides a PIN. The module retrieves and decrypts the correct PIN from the Card and compares it with the PIN entered by the operator. The PIN is 8-digits in length.

This method of authentication is impossible without possession of a valid Card. As such, false authentication would require a Card to be spoofed. Card integrity is provided by a 32-bit CRC across the internal data; both are stored encrypted with one of the Smart Card Keys. After four wrong tries of entering the PIN, the smart card becomes locked along with its Administrator account and there is no way of unblocking it.

Privileged Users

Security Officers can create new users (partitions). At creation, a client application (privileged user) belonging to this role is given the User Setup Password. User Setup Password is a temporary password. It consists of 25 alphanumeric characters, each of which can be any of 36 values (A-Z, 0-9). This password expires after three days by default.

After the first-time use with the User Setup Password, a User Secret is exchanged between the TOE and the privileged User. This is a random 256-bit value for machine-to-machine authentication. This User Secret along with the user name is used to derive the trusted path for the Users in operational use. By default after 100 failed login attempts to the TOE within 5 minutes the User becomes locked for 5 minutes. These values are configurable by Administrators. Also the failed attempts are logged.

Key Owner (Signer)

The TOE uses delegated authentication to identify and authenticate Signers.

The public keys of the people who can authorise the keys are stored within the SKA key attributes. This can be different for block, unblock, use(sign) and modify authorisation settings. In eIDAS SAM mode, the certificate that is used for authorization for signing must be issued by one of the previously configured CAs. On each request for the usage of the SKA key, the client application forwards the authorisation (signature). R.SAD is signed by the authorization key meaning it is integrity protected during transit of the data outside and inside the TOE.

If the authorisation signature cannot be verified successfully for the selected operations the authoriser will be blocked for 5 minutes. Therefore, the authoriser is not able to authorise any key in the TOE during this time.

Whenever a User tries to use one of its private keys a re-authentication is needed.

Related SFRs: FIA_UAU.6/KeyAuth, FIA_AFL.1/Admin, FIA_AFL.1/User, FIA_AFL.1/Signer, FIA_ATD.1, FIA_UAU.5/Signer, FIA_UAU.5/Privileged User, FIA_UID.2, FIA_USB.1, FPT_RPL.1, FDP_UCT.1, FDP_UIT.1, FMT_MTD.1

9.2 Key Management

The TOE handles System keys and user keys as described in Critical Security Parameters (CSPs) table.

System keys

System keys are supporting the operation of the TOE. Encrypting keystore, backups, supports authentication etc… Some system keys are generated in setup wizard and cannot be changed (KEK, Keystore Key, Genesis PIN, SO Card Keys, Backup Key). SO PINs are created when creating new SO. API keys are created when a new User (client application) is created. User keys are created by the client applications in operational state. Partition SO keys are generated by Security Officers during creating new users (new partitions). All those keys have their predefined format and size.

Administrators can create backup of the keystore therefore the keys as well. They can restore the backup on the same device or on other devices as well. The keys can be exported for external storage as well but there is no way any key can leave the TOE in plain format. Both backups or wrapped keys leave the TOE only in encrypted format and protected by integrity and confidentiality. The backup and restore operation always need at least two Security Officers to be performed due to dual control.

User Keys

User keys are generated by the Users (client application) and they can be used for different purposes as the User wants to use them controlled by API commands. User keys can be generated, used and deleted by the Users. The supported algorithms key sizes and operations can be found in Cryptographic Algorithms table.

User keys have many attributes and capabilities stored along with the keys. The capabilities and attributes store all information of the keys. For example: whether the key can be exported or not, whether the key is modifiable or deletable. Whether it is a private or public key etc… Capabilities define what can be done with the keys. For example the key can be used for encrypt, decrypt, sign etc…

The different types of keys have their default values for all capabilities and flags but some of the values can be changed on creation. Not all of them as there are rules, for example an assigned key is never extractable.

The default values and the modifiable attributes can be found in Table 10: Key Attribute Modification Table and in Table 11: Key Attribute initialization table.

Destroying keys are according to [FIPS-140-3] Level 3 zeroisation method.

SKA Keys

SKA Keys are special user keys implemented by Securosys. Smart Key Attributes feature allows for a fine-grained authorization of private key usage.

They have additional authorisation properties defining who can authorise the keys for different purposes. It can be defined who can block/unblock the key, who can use it (sign) and who can change the authorisation rules. With SKA Keys it is possible to identify the Signer (key owner not the client application). If the TOE is configured in eIDAS SAM mode, for signing purposes the use attribute of the key must be issued by one of the previously registered CAs. , other than stateful hash based (HSS_LMS, XMSS), from the TOE can be backed up and restored in the TOE if needed as described above. The backups are always encrypted and can be used only after the backups are restored and decrypted inside the TOE. On top of the keys, creating backup is available for all configuration, TOE state and every data. It is possible for the whole device or for specific partitions. The detail of the backup is further described in sec. 3.4.2.5 Backup.

Related SFRs: FCS_CKM.1, FCS_CKM.6, FDP_IFC.1/KeyBasics, FDP_IFF.1/KeyBasics, FDP_ACC.1/Backup, FDP_ACF.1/Backup, FMT_MSA.1/GenKeys, FMT_MSA.1/AKeys, FMT_MSA.3/Keys, FMT_MSA.1/Signer, FMT_MSA.1/Privileged User, FMT_MSA.2, FMT_MSA.3/Signer, FMT_MSA.3/Privileged User

9.3 Cryptographic functions

Crypto API

The Primus HSM provides a wide selection of application programming interfaces (PKCS#11, JCA/JCE, MS CSP) so that it can be used with almost any business application ranging from simple data encryption to identity management, PKI, strong authentication, and digital-signature generation and verification. The units are easy to install, configure, and integrate into existing networks.

Cryptographic operations are available through the above mentioned APIs for the Users (client application). The User role is accessed over the API (e.g., by business applications or clients) and serves to manage and use the User Keys. The User role may generate, load, and perform cryptographic operations with these keys.

User Keys, private, secret and public can only be accessed if the user (client application or in case of SKA keys the key owner) is authenticated. This includes listing of available keys or any other operation with keys.

The supported cryptographic algorithms, operations and key sizes are listed in Cryptographic Algorithms table.

Destroying keys are according to [FIPS-140-3] Level 3 zeroisation method.

Random number generation

The random number generator used by the TOE is composed of two main blocks:

- PTG.3 compliant entropy source, block_cipher_df (based on AES256), SP800-90Ar1
- DRG.4 compliant Random number generator seeded by the above entropy source. This is HMAC-DRBG SP800-90Ar1 with SHA256.

The RNG provides forward secrecy, backward secrecy, enhanced forward secrecy as defined in DRG.4 class.

Related SFRs: FCS_CKM.1, FCS_CKM.6, FCS_COP.1, FDP_ACC.1/KeyUsage, FDP_ACF.1/KeyUsage, FCS_RNG.1,

9.4 Audit/Administration

The TOE maintains the following roles: Administrator (Genesis, SO, Partition SO, Partition Auditor) User (External client application). Details can be found in sec 8.1 Authorisation, and in sec 2.4.2.6 Available services by roles.

In case the TOE is used as a CM only and it is connected to a certified SAM according to [EN 419241-2] the Signers are identified by the SAM outside the TOE. In case the TOE is configured as a SAM+CM an external identity provider shall identify and authenticate the Signer then it will be verified in the TOE.

The management functions for the Administrators are collected in Authorized Services table.

SO can block User (client application) accounts by making them offline and unblock them making them online. Also a SKA key can be blocked/unblocked if the User (key owner) has the block/unblock rules configured on the specific key but this operation is handled by the client application, the TOE only provides API for it.

TOE logs each security relevant actions such as startup, shutdown, user authentication, all cryptographic operations and many more. Each error (if there are any) is audited during any security relevant functions. Each audit record contains a proper timestamp, the user id who caused the event and the event type. Audit data is stored securely in a ring buffer. There is no deletion operation but the oldest records are overwritten when the storage of audit records is full. Audit records can be deleted only by factory reset which is restricted to Administrator role. There is no way to modify any audit records. Administrators can export the audit logs to USB or WebDAV so they can backup the logs any time. Also they can configure an external audit server (eg. syslog). The TOE can forward the audit records to the external server. This channel is only for outgoing communication. The external server has no access to the TOE.

The timestamps for the audit logs are reliable. The TOE supports connection to multiple NTP servers or NTP pool The rfc5905 generated reference timestamp is verified against the local clock.

Related SFRs: FMT_SMR.2, FMT_SMF.1, FMT_MTD.1/Unblock, FMT_MTD.1/AuditLog, FAU_GEN.1, FAU_GEN.2, FAU_STG.3, FPT_STM.1, FDP_ACC.1/TOE Maintenance, FDP_ACF.1/TOE Maintenance, FDP_IFC.1/Privileged User, FDP_IFF.1/Privileged User,

9.5 Secure Channels/Data Protection

Secure Channels

The TOE uses a special protocol for securing the communication with the external client applications and also with Decanus remote terminal. This protocol ensures the authentication and Diffie-Hellmann key agreement between the TOE and external entities. The encryption algorithm for securing the communication uses different algorithms for securing the channel. KAS for key agreement, KDF to derive the session key and AES-CGM to encrypt the messages. More details can be found in Cryptographic Algorithms table.

Integrity Protection

The TSF data is integrity protected by a checksum (64 Bit Hash), which is verified before each use of the key. The Key files include the standard attributes (flags and capabilities) and the extended SKA Attributes (Authorizations). In case the hash doesn’t match the operation cannot be processed and the user (client application) is notified that its data is corrupted.

Whenever a key is deleted it is deleted with all its attributes. Whenever a User (client application with its partition) is deleted it is deleted with all its keys and configuration data.

Self-tests

Each time the Module is powered up it tests that the cryptographic algorithms still operate correctly and that sensitive data have not been damaged (integrity). Power up self–tests are available on demand by power cycling the module. On power up, the Module performs many self-tests. It tests all the supported cryptographic algorithms (encryption/decryption/key generation/signature verification etc…) Power up test also runs an integrity check on the firmware. All tests must be completed successfully prior to any other use of cryptography by the Module. If one of the tests fails, the Module enters the error state. The system uses simple memory comparison to test the value of a test against its expected value. In cases where the comparison operation could be used for side channel attacks, the memory compare function is expanded in a way to compare all bytes instead of just until the first mismatch. Only after successful self-test and power up, the Ethernet goes up and the HSM is available to the user (client application).

Additionally, conditional tests are also available on the TOE. These tests run each time when a condition occurs. For details see Conditional Self-tests table.

Physical protection

All critical CSPs are encrypted with KEK in the HSM. There are factory mounted tamper-evident seals on Primus HSM and a tamper-response mechanism is implemented which can zeroise KEK and the digital seal in the event of physical breach therefore none of the keys can be used in the HSM because. The TOE also has multiple sensors for detecting different types of tamper attacks. The TOE is protected against removing the cover, light detection or freeze attack with low or high temperature as well. The protection is [FIPS 140-3] Level 3 compliant.

Related SFRs: FDP_SDI.2, FDP_RIP.1, FTP_TRP.1/External, FPT_TST.1, FPT_PHP.1, FPT_PHP.3, FPT_FLS.1, FPT_TDC.1, FTP_TRP.1/SSA, FTP_ITC.1/CM

User Data Protection

Primus HSM is shipped with a Genesis card or a Genesis PIN that represents the Privileged User, Admin. During the initial startup the Genesis shall create the first Security Officer (SO) and also the first Privileged User, User (client application).

Signers are always created by Users. Signers Keys SKA attributes contains policies on who can authorise to use, delete or update the key attributes. Signer’s keypair is always generated by User. Signer’s keys can be deleted by either the one who is authorised for deletion in the SKA Policies, or by Admins during deletion of Partition.

For creating signatures, the request is signed with the Signer’s private key which is validated against the SKA keys authorisation public key for signing. The DTBS/R is provided by the User. The Signer is authorised before every signature so the signing key doesn’t stay active after the signing operation is finished.

Admins can export the following User data: List of all key names belonging to the User, Syslogs.

Related SFRs: FIA_UID.1, FIA_UAU.1, FDP_ACC.1/Privileged User Creation, FDP_ACF.1/Privileged User Creation, FDP_ACC.1/Signer Creation, FDP_ACF.1/Signer Creation, FDP_ACF.1/Signer Maintenance, FDP_ACC.1/Signer Maintenance , FDP_ACC.1/Signer Key Pair Generation, FDP_ACF.1/Signer Key Pair Generation, FDP_ACC.1/Signer Key Pair Deletion, FDP_ACF.1/Signer Key Pair Deletion. FDP_ACC.1/Signing, FDP_ACF.1/Signing, FDP_IFC.1/Signer, FDP_IFF.1/Signer, FDP_ETC.2/Privileged User

10 Bibliography

[CCP1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model; CCMB-2022-11-001, CC:2022, Revision 1, November 2022
[CCP2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements; CCMB-2022-11-002, CC:2022, Revision 1, November 2022
[CCP3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements; CCMB-2022-11-003, CC:2022, Revision 1, November 2022
[CCP4]	Common Criteria for Information Technology Security Evaluation Part 4: Framework for the specification of evaluation methods and activities; CCMB-2022-11-004, CC:2022, Revision 1, November 2022
[CCP5]	Common Criteria for Information Technology Security Evaluation Part 5: Pre-defined packages of security requirements; CCMB-2022-11-005, CC:2022, Revision 1, November 2022
[CEM]	Common Methodology for Information Technology Security Evaluation, Evaluation methodology; CCMB-2022-11-006, CEM:2022, Revision 1, November 2022, [CEM]
[TP]	Transition Policy to CC:2022 and CEM:2022, CCMC-2023-04-001, April 20th, 2023
[CEN]	CEN/ISSS WS/E-Sign; Area D1, CWA 14167-1: Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures
[CEN TS 419 241]	CEN TS 419 241 Requirements for Trustworthy Systems Supporting Server Signing
[EN 419241-1]	EN 419241-1, Trustworthy Systems Supporting Server Signing — Part 1: General System Security Requirements
[EN 419241-2]	Trustworthy Systems Supporting Server Signing - Part 2: Protection Profile for QSCD for Server Signing, EN 419241-2:2019, February 2019

[EN 419221-5] Protection Profiles for Trust Service Provider Cryptographic Modules - Part 5: Cryptographic Module for Trust Services, EN 419221 - 5:2018, May 2018

[CWA 14170] prEN 14170-1:2011

Protection profiles for signature creation and verification application Part 1: Introduction to the European Norm

[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

[EN 419221-1] Protection Profiles for TSP cryptographic modules – Part 1: Overview, prTS 419221-1:2015, 2015-08

[TS 119 312] ETSI TS 119 312

Electronic Signatures and Infrastructures (ESI); Cryptographic Suites

[NIST800-90] National Institute of Standards and Technology, Information Technology Laboratory, Computer Security Division: The NIST SP 800-90 Deterministic Random Bit Generator Validation System (DRBGVS), October 30, 2007

[KS2011] W. Killmann, W. Schindler, „A proposal for: Functionality classes for random number generators“, Version 2.0, September 18, 2011

[UG] Primus HSM User Guide v3.2 Edition 29, 2025-10

[FIPS 140-3] National Institute of Standards and Technology, Security Requirements for Cryptographic Modules, FIPS PUB 140-3, March 22, 2019

[FIPS 180-4] National Institute of Standards and Technology, *Secure Hash Standard (SHS)*, FIPS PUB 180-4, August 2015

[FIPS 186-5] National Institute of Standards and Technology (2023) Digital Signature Standard (DSS). (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 186-5

[FIPS 197] National Institute of Standards and Technology (2001) Advanced Encryption Standard (AES). (Department of Commerce, Washington, D.C.), Federal

Information Processing Standards Publication (FIPS) NIST FIPS 197-upd1, updated May 9, 2023

[FIPS 198-1] National Institute of Standards and Technology, *The Keyed-Hash Message Authentication Code (HMAC)*, FIPS PUB 198-1, July 16, 2008

[FIPS 202] National Institute of Standards and Technology, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, FIPS PUB 202, August 2015

[FIPS 203] National Institute of Standards and Technology, *Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM)*, FIPS PUB 203, issued August 14, 2024

[FIPS 204] National Institute of Standards and Technology, *Module-Lattice-Based Digital Signature Standard (ML-DSA)*, FIPS PUB 204, August 2024

[FIPS 205] National Institute of Standards and Technology, *Stateless Hash-Based Digital Signature Standard (SLH-DSA)*, FIPS PUB 205, August 13, 2024.

[SP800-38A] National Institute of Standards and Technology, *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*, NIST Special Publication 800-38A, December 2001

[SP800-38B] National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, NIST Special Publication 800-38B, May 2005

[SP800-38D] National Institute of Standards and Technology, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, NIST Special Publication 800-38D, November 28, 2007

[SP800-38F] National Institute of Standards and Technology, *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*, NIST Special Publication 800-38F, December 2012

[SP 800-56A Rev. 3] National Institute of Standards and Technology, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, NIST Special Publication 800-56A Revision 3, April 2018

[SP800-56B Rev. 2] National Institute of Standards and Technology, *Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography*, NIST Special Publication 800-56B Revision 2, March 2019

[SP 800-56C Rev. 2] National Institute of Standards and Technology, *Recommendation for Key-Derivation Methods in Key-Establishment Schemes*, NIST Special Publication 800-56C Revision 2, August 2020.

[SP800-90A Rev1] National Institute of Standards and Technology, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST Special Publication 800-90A Revision 1, June 2015

[SP 800-108 Rev. 1] National Institute of Standards and Technology, *Recommendation for Key Derivation Using Pseudorandom Functions*, NIST Special Publication 800-108 Revision 1, February 2, 2024

[SP 800-132] National Institute of Standards and Technology, *Recommendation for Password-Based Key Derivation Part 1: Storage Applications*, NIST Special Publication 800-132, December 2010

[SP 800-208] National Institute of Standards and Technology, *Recommendation for Stateful Hash-Based Signature Schemes*, NIST Special Publication 800-208, October 2020

[PKCS #1 v2.1] RSA Laboratories, *PKCS #1: RSA Cryptography Specifications Version 2.1*, November 2002.

11 Acronyms

AES	Advanced Encryption Standard
CC	Common Criteria
CMAC	Cipher-based Message Authentication Code
CSP	Critical Security Parameter
DTBS	Data To Be Signed
DTBS/R	Data to be signed or its unique representation
DRBG	Deterministic Random Bit Generator
DRNG	Deterministic Random Number Generator
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ECDSA	Elliptic Curve DSA
EDDSA	Edwards-curve Digital Signature Algorithm
GCM	Galois/Counter Mode
HSM	Hardware Security Module
HSS-LMS	Hierarchical Signature System - Leighton-Micali Signature Scheme
HMAC	Hash-based Message Authentication Code
IT	Information Technology
JCA/JCE	Java Cryptography Architecture, Java Cryptography Extension. Crypto libraries for java
KAS	Key Agreement Scheme
KAT	Known Answer Test

KBKDF	Key-Based Key Derivation Function
KDA	Key Derivation Algorithm
KDF	Key Derivation Function
KEK	Key encryption key
KTS	Key Transport Scheme
MS CSP	Microsoft Cloud Solution Provider
ML-KEM	Mulit-Level Key Encapsulation Mechanism
PCIe	Peripheral Component Interconnect Express
PBKDF	Password-Based Key Derivation Function
PKCS#11	Public-key Cryptography Standards
PKI	Public Key Infrastructure
PP	Protection Profile
RNG	Random Number Generator
RSA	Rivest–Shamir–Adleman
SAR	Security assurance requirements
SFP	Security Function Policy
SFR	Security functional requirements
SHA	Secure Hash Algorithm
SHL-DSA	Smartcard-Hardware Library – Digital Signature Algorithm
SHS	Secure Hash Standard
SIC	Signer Interaction Component (defined by [EN419241-2])
SKA	Smart Key Attributes

ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	Trust Service Provider
XMSS	eXtended Merkle Signature Scheme