Primus HSM

Security Target

Version: 1.02

Date: 2021-03-19

Securosys SA

Förrlibuckstrasse 70

8005 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

Table of content

1 Terminology 6
2 Conformance Claim 7
2.1 CC Conformance Claim 7
2.2 PP Claim 7
2.3 Package Claim 7
2.4 Conformance Rationale 7
3 ST Introduction 8
3.1 ST Reference 8
3.2 TOE Reference 8
3.3 TOE Overview 8
3.3.1 TOE type 8
3.3.1.1 Usage and major security features of the TOE 8
3.3.1.2 Available non-TOE hardware/software/firmware 9
3.4 TOE Description 9
3.4.1 Physical scope of the TOE 9
3.4.2 Logical scope of the TOE 12
3.4.2.1 Cryptographic Functions 14
3.4.2.2 Key Management 14
3.4.2.3 Cryptographic Algorithms 17
3.4.2.4 Backup 19
3.4.2.5 Audit 20
3.4.2.6 Available services by roles 20
3.4.2.7 NON-TOE features 21
4 Security Problem Definition 23
4.1 Assets 23
4.2 Subjects 23
4.3 Threats 23
4.4 Organisational Security Policies 25
4.5 Assumptions 26
5 Security Objectives 28
5.1 Security Objectives for the TOE 28
5.2 Security Objectives for the Operational Environment 31
6 Extended Components Definitions 33
6.1 Generation of random numbers (FCS_RNG) 33
6.2 Basic TSF Self Testing (FPT_TST_EXT.1) 33
7 Security Requirements 35
7.1 Typographical Conventions 35
7.2 SFR Architecture 35
7.2.1 SFR Relationships 35

7.2.2 SFRs and the Key Lifecycle 38
7.3 Security Functional Requirements 39
7.3.1 Cryptographic Support (FCS) 39
7.3.2 Identification and authentication (FIA) 42
7.3.3 User data protection (FDP) 46
7.3.4 Trusted path/channels (FTP) 52
7.3.5 Protection of the TSF (FPT) 54
7.3.6 Security management (FMT) 57
7.3.7 Security audit data generation (FAU) 65
7.4 Security Assurance Requirements 68
7.4.1 Refinements of Security Assurance Requirements 69
ADV_ARC.1 Security architecture description 69
AGD_OPE.1 Operational user guidance 70
ATE_IND.2 Independent testing – sample 71
AVA_VAN.5 Advanced methodical vulnerability analysis 71
8 Rationales 73
8.1 Security Objectives Rationale 73
8.1.1 Security Objectives Coverage 73
8.1.2 Security Objectives Sufficiency 74
8.1.2.1 Threats 74
8.1.2.2 Organisational Security Policies 74
8.1.2.3 Assumptions 75
8.2 7.2 Security Requirements Rationale 75
8.2.1 Security Requirements Coverage 75
8.2.2 SFR Dependencies 78
8.2.3 Rationale for SARs 80
8.2.4 AVA_VAN.5 Advanced methodical vulnerability analysis 80
9 TOE Summary Specification 81
9.1 Authorisation 81
9.2 Key Management 81
9.3 Cryptographic functions 82
9.4 Audit/Administration 83
9.5 Secure Channels/Data Protection 84
10 Bibliography 86
11 Acronyms 88

List of Tables

Table 1: Modified SFRs 7
Table 2: Ports and Interfaces (E-Series) 10
Table 3: Ports and Interfaces (X-Series) 11
Table 4: TOE Deliverables 12
Table 5: Roles and authentication Data 13
Table 6: Critical Security Parameters (CSPs) 16
Table 7: Cryptographic Algorithms table 17
Table 8: Authorized Services 20
Table 9: Conditional Self-tests 55
Table 10: Key Attributes Modification Table 62
Table 11: Key Attributes Initialisation Table[82] 64
Table 12: Security Assurance Requirements 68
Table 13: Security Problem Definition mapping to Security Objectives 73
Table 14: TOE Security Objectives mapping to SFRs 75
Table 15: SFR Dependencies Rationale 78

List of Figures

Figure 1: E-Module Front with cryptographic boundary in red 10
Figure 2: E-Module back with cryptographic boundary in red 10
Figure 3: X-Module Front with cryptographic boundary in red 11
Figure 4: X-Module back with cryptographic boundary in red 11
Figure 5: TOE Architecture 12
Figure 6: Architecture of Key Protection SFRs 36
Figure 7: Architecture of User, TSF Protection & Audit SFRs 37
Figure 8: Generic Key Lifecycle and Related SFRs 38

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

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.

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-2017-04-001, Version 3.1, Revision 5, April 2017 [CCP1].
- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017 [CCP2].
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements; CCMB-2017-04-003, Version 3.1, Revision 5, April 2017,[CCP3].

as follows:

- Part 2 extended; and
- Part 3 conformant.

The following must be considered:

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

2.2 PP Claim

This Security Target claims strict conformance to the Protection Profile EN 419 221-5 Protection Profiles for TSP Cryptographic Modules – Part 5: Cryptographic Module for Trust Services; [EN 419221-5]

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].

SFRs modified compared to [EN 419221-5] are in Table 1: Modified SFRs below. This table contains the iterations and refinements created by the ST Author. The operations needed by the PP are not listed here.

Table 1: Modified SFRs

SFR	Operation
FIA_AFL.1	Iterated: • FIA_AFL.1/Admin • FIA_AFL.1/User • FIA_AFL.1/Key owner
FCS_RNG.1	Refined
FMT_MTD.1/AuditLog	Refined
FIA_AFL.1/User	Refined

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	1.02
Date	2021.03.19.
Protection Profile	Protection Profiles for TSP Cryptographic modules – Part 5 [EN 419221-5]
Assurance level	EAL 4+ (augmented with AVA_VAN.5)
ST Author	Securosys SA

3.2 TOE Reference

Name	PRIMUS HSM
Version	FW 2.8.21
Series	Series E, Series X

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
- Series X: X200, X400, X700, X1000

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 and is already certified according to FIPS 140-2 overall Level 3 requirements.

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 (AES, Camellia), asymmetric (RSA, DSA, ECC, Diffie-Hellman), and hashing (SHA-2, SHA-3) cryptographic algorithms. Primus also contains a secure vault implemented inside a dedicated security chip, and also offers FIPS-140-2 Level3 compliant tamper protection. The Primus HSM is available in several performance levels based on the Securosys Primus Family.

The TOE can be used (but not limited to) as a Cryptographic Module of TSP's supporting requirements for remote signing, or sealing, as specified in Regulation 910/2014. This case the TOE would be used in conjunction with the protection profile to be defined in [EN 419241-2], and any other related protection profiles, to meet the requirements for Sole Control Assurance Level 2 as defined in [EN 419 241-1].
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.

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).

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.

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.

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.

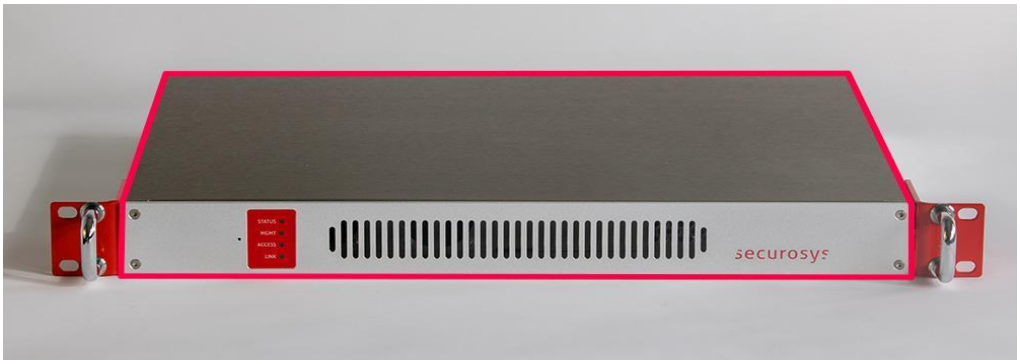


Figure 1: E-Module Front with cryptographic boundary in red

Figure 2: E-Module back with cryptographic boundary in red

Table 2: 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

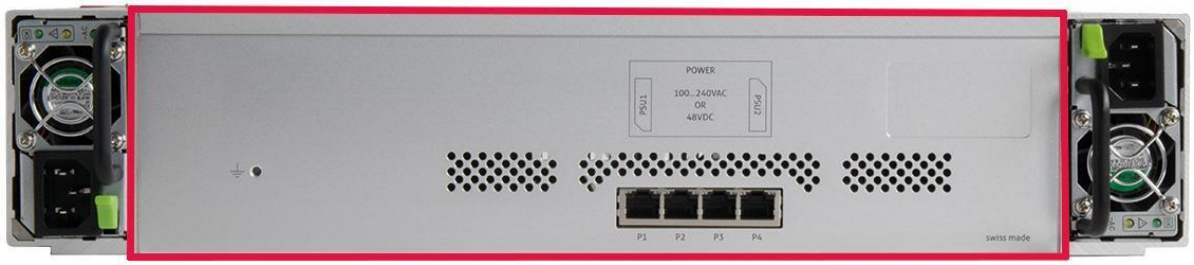


Figure 4: X-Module back with cryptographic boundary in red

Table 3: 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

The TOE deliverable parts are as follows:

Table 4: TOE Deliverables

Type	Description	Delivery
HSM module	Both E and X series	Courier
Accessories	**E-Series** - 2 power cable - 1 USB memory stick - 2 Genesis Card (GN) - 3 Security Officer (SO) Card **X-Series** - 1 power cable - 1 USB memory stick	Courier
Guidance	QuickStart guide (.pdf format)	Courier
Guidance	User Guide (.pdf format)	Web Download
Firmware	Primus HSM Firmware 2.8.21 (.hsm - encrypted file format)	Courier (pre-installed) or Web Download

3.4.2 Logical scope of the TOE

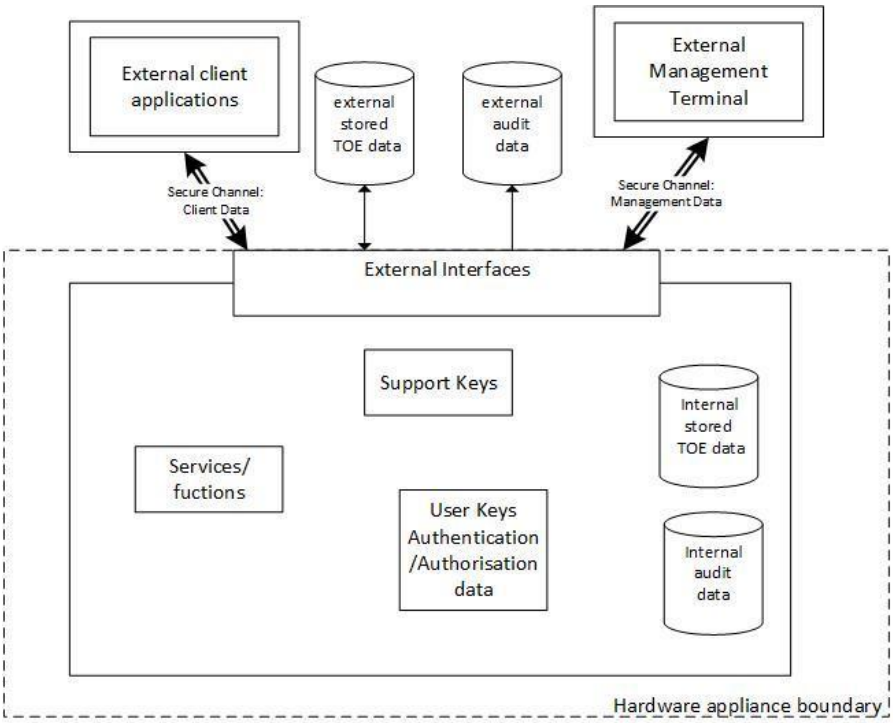


Figure 5: TOE Architecture

The hardware appliance boundary in Figure 5 represents the enclosure of the computing appliance which hosts the TOE.

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 and Partition SO 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	Administrative role. Sets up the module. Performs factory reset.	Identity-based	PIN and Card
			PIN Only[1]
Security Officer (SO)	Administrative role which manages the module.	Identity-based	PIN and Card
			PIN Only
User (client application)	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

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.

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. If the client application is a certified SAM according to [EN 419241-2] the use of the normal keys is also allowed for signatures without the user (key owner) authorisation because that case the sole control is guaranteed by the SAM.

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.

[1] 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 its security features including blocking the card – and the admin account as well.

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.

3.4.2.1 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.

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 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.

3.4.2.2 Key Management

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

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

The TOE supports the following techniques for establishing keys:

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 6.3.6)
- 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 10: 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 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
  - External backup
  - Cloning
  - Clustering

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

  In case of cloning or clustering is configured for the TOE, the keys also leave the TOE and are synchronised/imported to another instance of the TOE. These features are also 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.

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-2048	PKI Key to sign the device PKI Key
Primus Device CA Key	RSA-2048	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 Table 7: 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 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.

3.4.2.3 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) under the cert number C1899 (https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=12706).

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
AES-CMAC	MAC Generation, MAC Verification	[SP 800-38B] Functions: Key sizes: 128, 192, 256 bits
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).
AES-KW	Key Wrap, Key Unwrap	[SP 800-38F] Modes: KW, KWP Key sizes: 128, 192, 256
DRBG	HMAC DRBG CTR DRBG	[SP 800-90A] HMAC DRBG with internal function SHA-512 CTR DRBG with internal function AES-256
DSA	PQG Generation, Key Pair Generation, Signature Generation, Signature Verification	[FIPS 186-4] Key sizes: 2048, 3072 bits
ECDSA	Key Pair Generation, Signature Generation, Signature Verification, Public Key Validation	[FIPS 186-4] Curves/Key sizes: P-224, P-256, P-384, P-521 (Strength: 112, 128, 192, 260)
HMAC	Generation, Verification	[FIPS 198-1] SHA sizes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512
KAS (FFC, ECC)	Key agreement, also used for secure connection with external management device (Decanus) and external application	[SP 800-56Ar1] 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

KDF	Line encryption for secure connection with external management device (Decanus) and external application	[SP 800-108] Modes: Counter, Feedback, Double Pipeline Iteration Mode PRFs: CMAC(AES-128/192/256), HMAC (SHA-1, 224, 256, 384, 512)
KTS (Symmetric)	Key Wrap, Key Unwrap	[SP800-38F] Variants: 38D: AES-GCM (256 bits) 38F: AES-KW, AES-KWP Key Transport – Provides between 128 and 256 bits of encryption strength.
RSA	Key Pair Generation, Signature Generation, Signature Verification, Component Test	[FIPS 186-4, ANSI X9.31-1998, and PKCS #1 v2.1 (PSS and PKCS1.5)] Key sizes: 2048, 3072, 4096 bits Some RSA-4096 functions are listed here but not displayed on RSA Cert. #2946. These are vendor-affirmed, as CAVP does not provide testing for these functions.
SHA	Digital Signature Generation, Digital Signature Verification, component of HMAC and HMAC_DRBG, general hashing	[FIPS 180-4, FIPS 202] SHA sizes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 SHA-256 digest of Backup Key is used with other measures for HSM system and partition backup as well.
Triple-DES (TDES)	Decryption	[SP 800-67] Modes: TECB, TCBC Key sizes: 3-key

3.4.2.4 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.

3.4.2.5 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. Internal audit logs can be collected by administrators and exported to external drives via USB. 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.

3.4.2.6 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

Table 8: Authorized Services

Service	Description	G	SO	U	PSO
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		X		

	CSP: uses Card Keys for SO activation				
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		

3.4.2.7 NON-TOE features

The TOE has some features which there are no requirements defined in the [EN 419221-5] 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.

R.SecretKey: secret keys used in symmetric cryptographic functions and private keys used in asymmetric cryptographic functions, managed and used by the TOE in support of the cryptographic services that it offers. This includes user keys, owned and used by specific users, and support keys used in the implementation and operation of the TOE. The asset also includes copies of such keys made for external storage and/or backup purposes. The confidentiality and integrity of these keys shall be protected.

R.PubKey: public keys managed and used by the TOE in support of the cryptographic services that it offers (including user keys and support keys). This asset includes copies of keys made for external storage and/or backup purposes. The integrity of these keys shall be protected.

R.ClientData: data supplied by a client for use in a cryptographic function. Depending on the context, this data may require confidentiality and/or integrity protection.

R.RAD: reference data held by the TOE that is used to authenticate an administrator (hence to control access to privileged administrator functions such as TOE backup, export of audit data) or to 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 shall be protected; its confidentiality shall also be protected unless the authentication method used means that the RAD is public data (such as a public key).

4.2 Subjects

The types of subjects identified in this PP are:

S.Application: a client application, or process acting on behalf of a client application and that communicates with the TOE over a local or external interface. Client applications will in some situations be acting directly on behalf of end users (see S.User).

S.User: an end user of the TOE who can be associated with secret keys and authentication/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.

4.3 Threats

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

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[2].

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[3]), 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:

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

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

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

4.4 Organisational Security Policies

P.Algorithms Use of approved cryptographic algorithms

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

Application Note 1

The relevant authorities and endorsements are determined by the context of the client applications that use the TOE. For digital signatures within the European Union this is as indicated in [Regulation] and an exemplary list of algorithms and parameters is given in [TS 119 312] or [SOG-IS-Crypto] (see also section 1.3.1.3).

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[4]), 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.

Application Note 2

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

P.RNG Random Number Generation

The TOE is required to generate random numbers that meet a specified quality metric, for use by client applications. These random numbers shall be suitable for use as keys, authentication/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.

Application Note 3

The cryptographic module TOE is assumed to be part of a larger system that manages audit data. The TOE therefore logs audit records, and it is assumed that these are collected, maintained and reviewed in the larger

[4] 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.

system. Hence there is no separate auditor role within the cryptographic module TOE, but the role of System Auditor is assumed to exist in the larger system – cf. A.AuditSupport in section 3.5.

4.5 Assumptions

A.ExternalData Protection of data outside TOE control

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

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

A.Env Protected operating environment

The TOE operates in a protected environment that limits physical access to the TOE to 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.

Application Note 4

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

A.AppSupport Application security support

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

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

OT.PlainKeyConf Protection of confidentiality of plaintext secret keys

The plaintext value of secret keys is not made available outside the TOE (except where the key has been exported securely in the manner of OT.ImportExport). This includes protection of the keys during generation, storage (including external storage), and use in cryptographic functions, and means that even 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 5

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

OT.KeyIntegrity Protection of integrity of keys

The value and critical attributes of keys (secret or public) have their integrity protected by the TOE against 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.

Application Note 6

Local client applications within a suitable security environment (such as client applications that are connected to the TOE by a channel such as a PCIe bus within the same hardware appliance) do not require authentication to communicate with the TOE, as noted in section 1.3.1. However, use of a secret key always requires prior authorisation.

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[5]. 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 7

Such limits on the use of a key after initial authorisation are termed "re-authorisation conditions" in this ST. A wide range of policies and re-authorisation conditions are allowed, and different policies may be applied to different types of secret key, but the re-authorisation conditions for all types of secret key shall be unambiguously defined in the Security Target. The decision to use supported 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 8

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

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

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 9

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

OT.ImportExport Secure import and export of keys

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

Assigned keys cannot be imported or exported.

OT.Backup Secure backup of user data

Any method provided by the TOE for backing up user data, including secret keys, preserves the security of the data and is controlled by 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 deletion or modification of records by providing tamper protection (either prevention or detection) for the audit log.

5.2 Security Objectives for the Operational Environment

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

OE.ExternalData Protection of data outside TOE control

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

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

OE.Env Protected operating environment

The TOE shall operate in a protected environment that limits physical access to the TOE to 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.

Application Note 10

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

OE.AppSupport Application security support

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

6 Extended Components Definitions

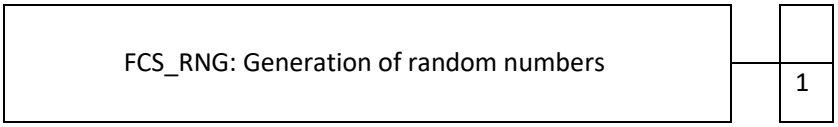
6.1 Generation of random numbers (FCS_RNG)

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

Family behaviour

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

Component levelling:



Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1	*Generation of random numbers*
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1	The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].
FCS_RNG.1.2	The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet [assignment: *a defined quality metric*].

Application Note 11

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

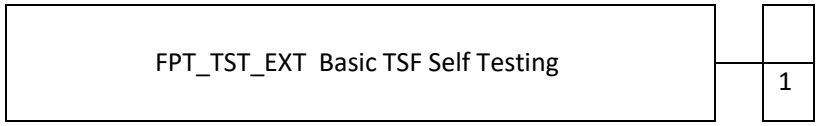
6.2 Basic TSF Self Testing (FPT_TST_EXT.1)

The extended component defined here is a simplified version of FPT_TST.1 in [CCP2]

Family behaviour

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

Component levelling:



Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

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

- Indication that TSF self test was completed.

FPT_TST_EXT.1	*Basic TSF Self Testing*
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TST_EXT.1.1	The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*] to demonstrate the correct operation of the TSF: [assignment: *list of self-tests run by the TSF*].

7 Security 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 given in section 6.3 "*TOE security functional requirements*" are drawn from Common Criteria part 2 [CCP2]. Some security functional requirements represent extensions to [CCP2], with a reasoning given in section 6.5. Operations for assignment, selection and refinement have been made. Operations not performed in this PP are identified in order to enable instantiation of the PP to a Security Target (ST).

The TOE security assurance requirements statements given in section 6.4 "*TOE Security Assurance Requirement*" are drawn from the security assurance components from Common Criteria part 3 [CCP3].

7.1 Typographical Conventions

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

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

- Refinements are denoted in one of two ways, depending on whether they add detail to an SFR ('explanatory refinements') or update the text of an SFR element ('element refinements').
- Explanatory refinements follow the SFR that they update and are marked by the word "Refinement" in bold followed by text describing the refinement. Element refinements are indicated by bold text within an SFR element, with the original text indicated in a footnote.
- Selections and assignments made in this PP are italicised, and the original text is indicated in a footnote.

ST SFR operations:

- ST operations are the same as the operations in the [EN 419221-5] with an additional underline.

- Iteration operation is marked with SFR_NAME/ITERATED_INSTANCE_NAME. The iterated SFRs are listed in "Table 1: Modified SFRs".

Application notes from [EN 419221-5] are not changed. They are the same as they are in the PP. The section, table and figure references therefore refers to their numbers of the PP which can be different from the ST's in some cases.

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

7.2 SFR Architecture

7.2.1 SFR Relationships

Figure 6 and Figure 7 give a graphical presentation of the connections between the Security Functional Requirements (SFRs) from section 6.3 below and the underlying functional areas and operations that the TOE provides. The diagrams provide a context for SFRs that relates to their use in the TOE, whereas section 6.3 defines the SFRs grouped by the abstract class and family groupings in [CCP2].

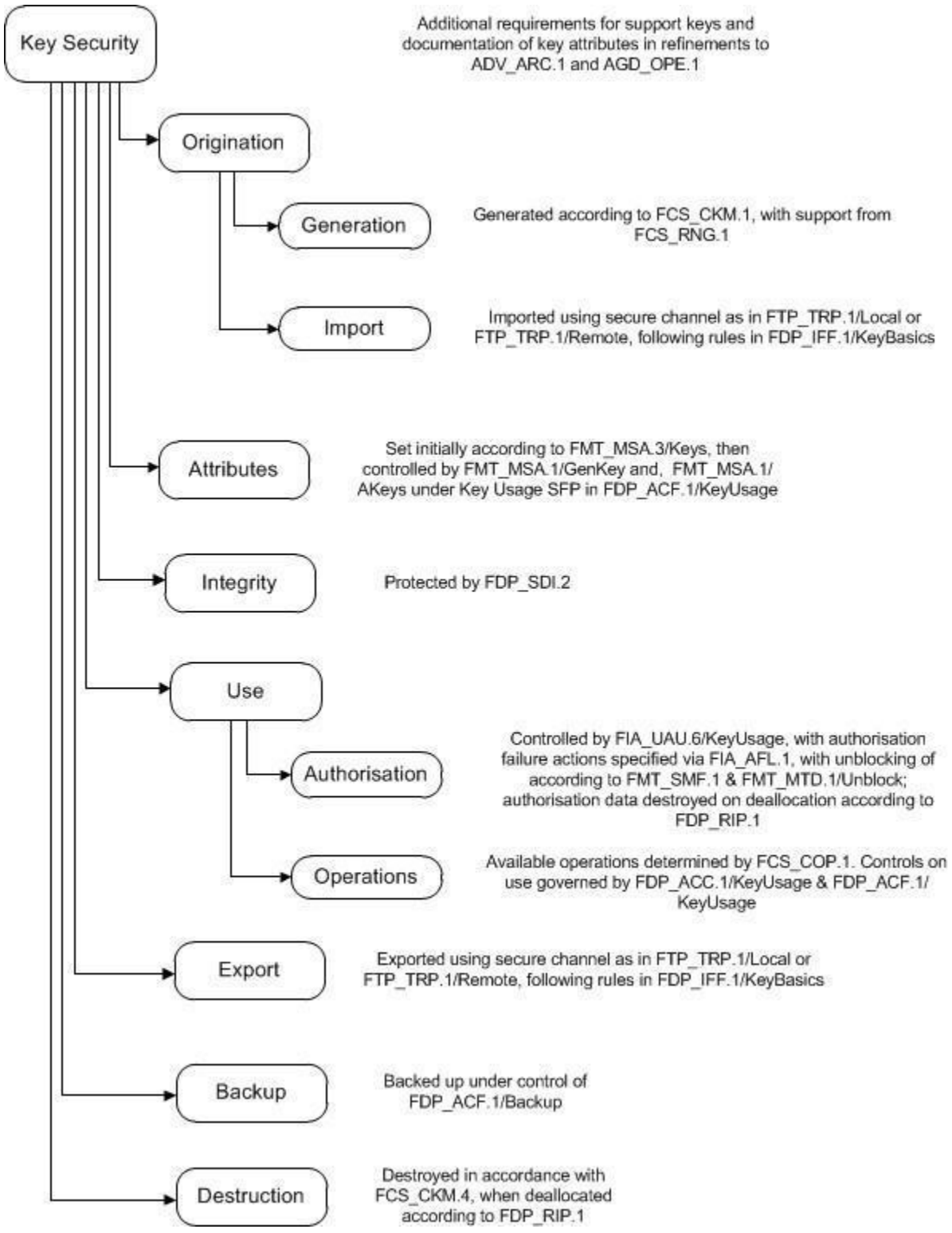


Figure 6: Architecture of Key Protection SFRs

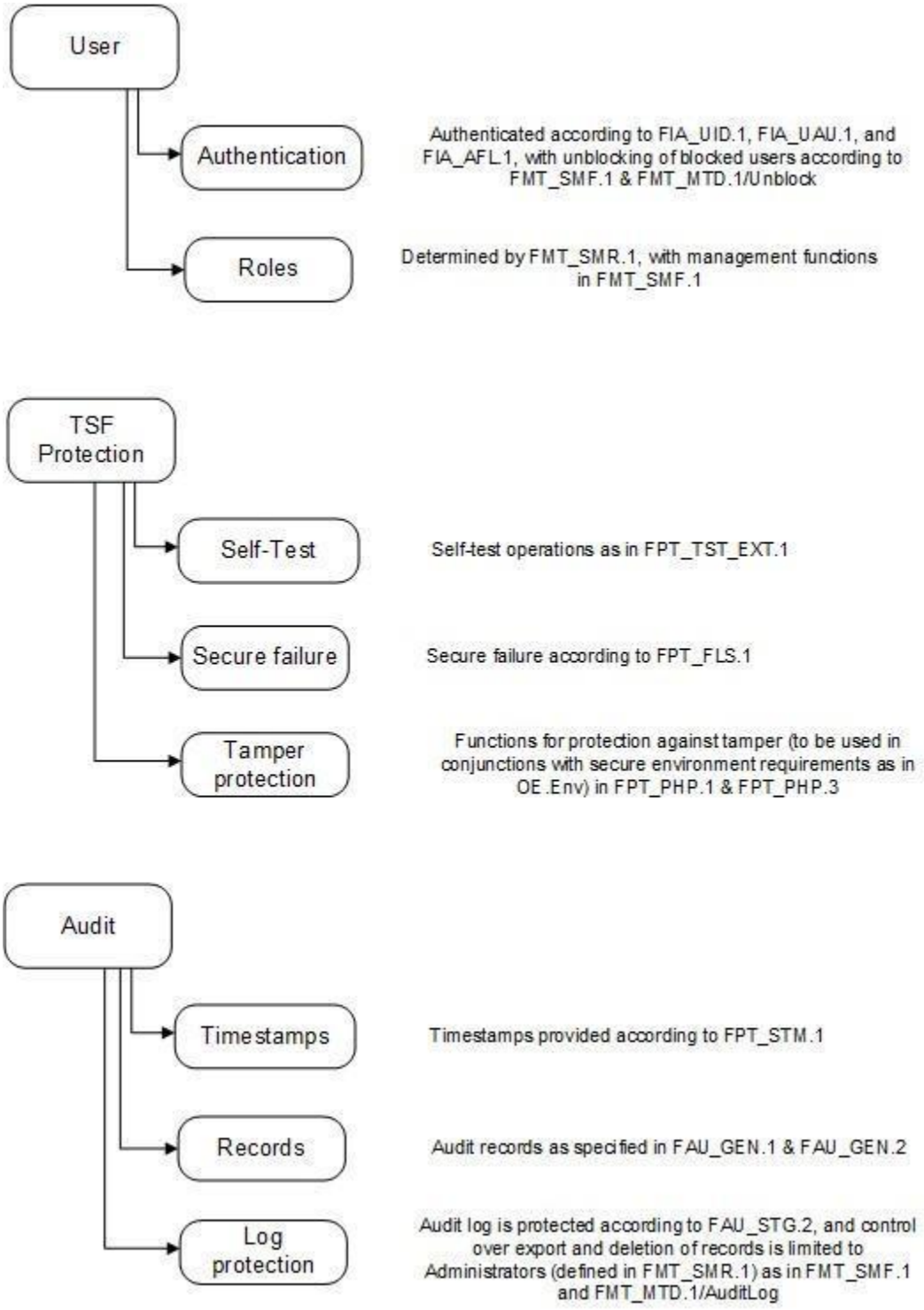


Figure 7: Architecture of User, TSF Protection & Audit SFRs

7.2.2 SFRs and the Key Lifecycle

The generic life cycle for a key is illustrated in Figure 8. This shows the methods by which a key may arrive in the TOE (import, generation or restore from backup), resulting in binding of a set of attributes to the key and storage of the key, and finally the ways in which a stored key may then be processed (export, use in a cryptographic function, backup, or destruction). The SFRs related to each of these aspects are then described below Figure 8.

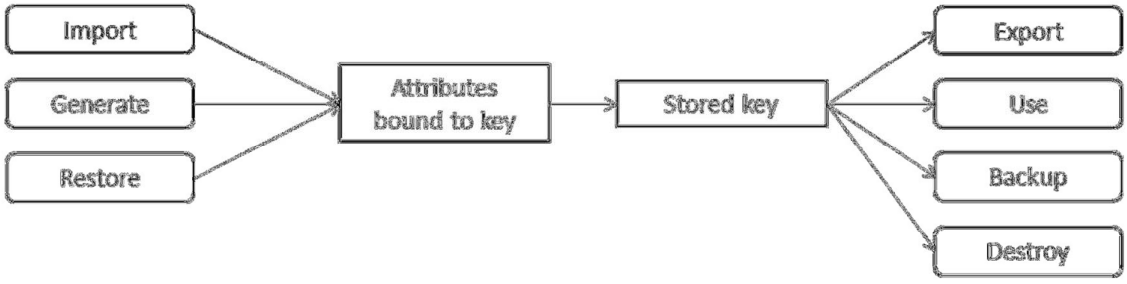


Figure 8: Generic Key Lifecycle and Related SFRs

Import:

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

Generate:

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

Restore:

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

Attributes bound to key:

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

Stored key:

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

Export:

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

- FAU_GEN.1 requires audit of export

Use:

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

Backup:

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

Destroy:

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

7.3 Security Functional Requirements

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

7.3.1 Cryptographic Support (FCS)

FCS_CKM.1	*Cryptographic key generation*
Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *specified in Cryptographic Algorithms table*[6] and specified cryptographic key sizes *specified in Cryptographic Algorithms table*[7] that meet the following: *specified in Cryptographic Algorithms table*[8].

Application Note 12

The Security Target shall include all key generation operations that are intended to support TSP operations using one or more iterations of FCS_CKM.1.

The relevant authorities and endorsements for completion of the SFRs are determined by the context of the client applications that use the TOE. For digital signatures within the European Union this is as indicated in [Regulation]

[6] [assignment: *cryptographic key generation algorithm*]
[7] [assignment*: cryptographic key sizes*]
[8] [assignment*: list of standards*]

and an exemplary list of algorithms and parameters is given in [TS 119 312] or [SOG-IS-Crypto] (see also section 1.3.1.3).

Note that key generation needs to be linked to the setting of security attributes of a key (including the link to a subject who owns the key, via the setting of authorisation data) as in FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys,

FCS_CKM.4	*Cryptographic key destruction*
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *zeroisation*[9] that meets the following: *FIPS 140-2 Level 3*[10]

Application Note 13

The Security Target shall specify the method(s) of secure destruction of all secret keys and all support keys, and shall ensure that all are covered by a secure destruction method. If necessary then more than one iteration of FCS_CKM.4 may be included to describe different standards for secure deletion. The 'list of standards' in the final assignment may be met in the Security Target by simply providing a description of the action taken to zeroise the keys rather than referencing an external standard.

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] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1	The TSF shall perform *list of functions specified in Cryptographic Algorithms table*[11] in accordance with a specified cryptographic algorithm *specified in Cryptographic Algorithms table*[12] and cryptographic key sizes *specified in Cryptographic Algorithms table*[13] that meet the following: *specified in Cryptographic Algorithms table*[14]

Application Note 14

The Security Target shall include all cryptographic functions that are intended to support TSP operations using one or more iterations of FCS_COP.1. This includes cryptographic operations for digital signatures and seals, implementing trusted paths (FTP_TRP.1) and secure channels (FTP_TRP.1), key encryption (e.g. FDP_IFF.1/KeyBasics), and any backups (FDP_ACF.1/Backup) that the TOE creates. If the TOE supports software

[9] [assignment*: cryptographic key destruction method*]
[10] [assignment*: list of standards*]
[11] [assignment: *list of cryptographic operations*]
[12] [assignment*: cryptographic algorithm*]
[13] [assignment*: cryptographic key sizes*]
[14] [assignment: *list of standards*]

or firmware updates then the iterations shall include the cryptographic operations used to support the validation of digital signatures on the updates as described in the refinement to ADV_ARC.1 in section 6.4.1.

The relevant authorities and endorsements for completion of each of these iterations are determined by the context of the client applications that use the TOE. For digital signatures and seals within the European Union this is as indicated in [Regulation] and an exemplary list of algorithms and parameters is given in [TS 119 312] or [SOG-IS-Crypto] (see also section 1.3.1.3).

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*[15] random number generator ***as defined in [NIST800-90]***[16] that implements: *capability list of class DRG.4 as defined in [AIS20]*. • (DRG.4.1) The internal state of the RNG shall *use PTRNG of class **PTG.3**[17] as random source*[18]. • (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*[19]. • (DRG.4.5) The internal state of the RNG is seeded by an *PTRNG of class PTG.3*[20] [21]
FCS_RNG.1.2	The TSF shall provide *octets of bits*[22] that meet • (DRG.4.6) The RNG generates output for which *2*[34] [23] strings of bit length 128 that are mutually different with probability of *> 1 - 2^-16^* [24]. • (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 Table 9: Conditional Self-tests*[25].[26]

Application Note 15

For more information on the selections and assignments see the SFR definition in section 5.1.

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

The Security Target describes the uses made of the RNG and its relationship to other SFRs such as FCS_CKM.1, and to any random number generation function/service made available to users or clients applications.

7.3.2 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_EXT.1* (2) *None*[27] [28]on behalf of the user to be performed before the user is identified.
FIA_UID.1.2	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note 16

The 'list of additional *TSF-mediated actions' may be left empty (equivalent to an assignment of 'None') if applicable.*

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_EXT.1,* *2) Identification of the user by means of TSF required by FIA_UID.1* *3) None*[29] [30]on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note 17

The Security Target shall separately identify any different types of identification and authentication, e.g. for Administrators, local users, application users, using separate iterations of the FIA_UID.1 and FIA_UAU.1 SFRs where the methods differ. The Security Target shall also separately identify the difference between authentication of users and authorisation for use of keys as required for FIA_UAU.6/KeyAuth. Separate iterations of FIA SFRs may be necessary to capture these separate cases.

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

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

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

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

The 'list of additional TSF-mediated actions' in FIA_UAU.1.1 may be left empty (equivalent to an assignment of 'None') if applicable.

FIA_AFL.1/Admin	*Authentication failure handling*
Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication
FIA_AFL.1.1/Admin	The TSF shall detect when *4*[31] unsuccessful authentication **or authorisation** attempts occur related to *consecutive failed authentication or authorisation attempts*[32].
FIA_AFL.1.2/Admin	When the defined number of unsuccessful authentication **or authorisation** attempts has been *met*[33], the **SO card becomes blocked**[34] *until* **forever so the Admin won't be able to authenticate anymore.**[35]

ST Application note

The Administrators (Genesis, SO or Partition SO) 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 behaviour 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 0-10000*[36] unsuccessful authentication **or authorisation** attempts occur in **an administrator configurable time period within 0-10000 seconds**[37] related to *consecutive failed authentication or authorisation attempts*[38].
FIA_AFL.1.2/User	When the defined number of unsuccessful authentication **or authorisation**

[31][selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[32][assignment: list of authentication events]
[33] [selection: met, surpassed]
[34] [refinement:TSF shall block access to [assignment: description of the relevant functionality]]
[35] [refinement: [selection: unblocked by [assignment: identification of the authorised subject or role], a time period [assignment: time period] has elapsed]]
[36] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[37] [refinement]
[38] [assignment: list of authentication events]

attempts has been *met*[39], the TSF shall *block* access to *the TOE API for the relevant user*[40] until *an administrator configurable time period within 0-10000 seconds has elapsed*[41].

ST Application Note

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/Key owner	*Authentication failure handling*
Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication
FIA_AFL.1.1/Key owner	The TSF shall detect *1*[42] unsuccessful authentication **or authorisation** attempts occur in related to *consecutive failed authentication or authorisation attempts*[43].
FIA_AFL.1.2/Key owner	When the defined number of unsuccessful authentication **or authorisation** attempts has been *met*[44], the TSF shall *block* access to *the ability to authorise any keys*[45] until *5 minutes has elapsed*[46].

ST Application Note

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.

Application Note 18

The Security Target shall separately identify the different types of authentication or authorisation to which failure responses apply, and this should include all of the different types of authentication identified for FIA_UAU.1 and failed authorisation attempts related to attempts to use keys as in FIA_UAU.6/KeyAuth. Where different authentication/authorisation failure responses apply then the SFR should be iterated.

[39] [selection: met, surpassed]
[40] [assignment: description of the relevant functionality]
[41] [selection:unblocked by [assignment: identification of the authorised subject or role], a time period [assignment: time period] has elapsed]
[42] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[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 unblocking of functionality blocked as described in each iteration of FIA_AFL.1.2 shall be described in a corresponding iteration of FMT_MTD.1 (cf. section 6.3.6).

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**[47] the user **for access to a secret key** under the conditions *(1) Authorisation in order to be granted initial access to the key; and* *(2) Authorisation on every subsequent access to the key*[48]. [49]

Application Note 19

Note that any use of a key requires an initial authorisation by presentation of the correct authorisation data. Subsequent uses may require re-authorisation on every use (in this case 'Authorisation on every subsequent access to the key' is selected in FIA_UAU.6.1/KeyAuth (2)), or else the TOE may allow some uses of the key without further authorisation until one of the specified re-authorisation conditions occurs.

The TOE may also allow different re-authorisation conditions for different types of secret key. The types of secret keys may be identified (in the first assignment in (2)) as individual keys, or in terms of a generic definition (e.g. 'all non-Assigned keys'). Where different re-authorisation conditions apply to different types of key then the second assignment in (2) may be used to specify the other types of key and the conditions that apply to them in a similar manner.

The explicit rescinding of an authorisation period in (2) ensures that client applications or users can decide to revoke a previous authorisation in (2) that may still be in force. If the TOE intends to allow unlimited uses of a secret key after initial authorisation, until authorisation is rescinded by a client application or user, then the selection 'after explicit rescinding of previous authorisation for access to the secret key' is chosen in the Security Target without any accompanying selections for time periods or number of uses. The Security Target describes the method or methods used for such rescinding (such as particular API commands).

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

Each 'use' of a key is expected to relate to one cryptographic function carried out with the key. If there are circumstances where a different interpretation may be placed on the 'use' of a key then this shall be identified

[47] re-authenticate

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

[49] *[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]

and explained in the Security Target and the Operational Guidance. The intention here is to make clear any situations that are relevant to a key owner who can be held responsible for use of the key (such as any case where a single authorisation for use of a key could allow the creation of more than one signature using the authorised key). Note that in order to make qualified electronic signatures under [Regulation] then the user/application shall be able to precisely control the signatures that can be made under each authorisation.

Actions taken by the TOE in the case of successive authorisation failures shall be specified using an iteration of FIA_AFL.1

7.3.3 User data protection (FDP)

FDP_IFC.1/KeyBasics *Subset information flow control*

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1/KeyBasics	The TSF shall enforce the *Key Basics SFP*[50] on *(1) subjects: all* *(2) information: keys* *(3) operations: all*[51].

FDP_IFF.1/KeyBasics *Simple security attributes*

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
FDP_IFF.1.1/KeyBasics	The TSF shall enforce the *Key Basics SFP*[52] 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*[53]*.*

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

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

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

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

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[54].

Application Note 20

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

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

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

FDP_IFF.1.3/KeyBasics The TSF shall enforce the **following additional information flow control rules:** *none*[55].

FDP_IFF.1.4/KeyBasics The TSF shall explicitly authorise an information flow based on the following rules: *none*[56].

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.

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

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

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

(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*[57]

Application Note 21

The requirements of FDP_IFF.1/KeyBasics apply regardless of how the key is stored by the TOE, including when the key is externally stored (cf. section 1.3.1.2).

Direct access to a key value in FDP_IFF.1.5/KeyBasics (1) is access that makes the value available for reading or modification – this includes operations that would subsequently allow reading or modification of the key (e.g. making a copy of the key with different attributes, or with a different object type that would then allow direct read access). Note that this PP assumes that key values are never modified after they have been generated.

Export of a key as in FDP_IFF.1.5/KeyBasics (1), (2), (4) and (6) is not the same as backup (governed by FDP_ACF.1/Backup) or external storage of keys under continuing TOE control (governed by other parts of the Key Basics SFP in FDP_IFF.1/KeyBasics, and the Key Usage SFP in FDP_ACF.1/KeyUsage). Thus an Export Flag of 'non-exportable' does not prevent backup or external storage of the keys under continuing TOE control.

The Security Target and/or Operational Guidance shall specify how any attributes not supplied with an imported key are set when the key is imported (or alternatively how such keys are rejected). Similarly the Security Target and/or Operational Guidance shall describe how the key's attributes are represented when exported, so that their meaning can be understood by the receiver.

If the TOE does not provide facilities to import or export keys then the relevant part of the SFR is trivially satisfied, and this should be stated in the Security Target.

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*[58] on *(1) subjects: all* *(2) objects: keys* *(3) operations: all*[59]*.*

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

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

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

[59] [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/KeyUsage The TSF shall enforce the *Key Usage SFP* [60] 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[61]*.*

Application Note 22

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

Whether a subject is currently authorised to change the attributes of a secret key is determined by the iterations of FMT_MSA.1 in section 6.3.6.

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 authorised subject, and only as permitted in the Key Attributes Modification Table

(2) Only subjects with current authorisation 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[62]*.*

Application Note 23

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

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

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

[61] [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*]

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

an encrypted form of a secret key (e.g. for external storage of keys) and that are not necessarily restricted in this way.

FDP_ACF.1.3/KeyUsage The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*[63].

FDP_ACF.1.4/KeyUsage The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[64]

Application Note 24

The requirements of FDP_ACF.1/KeyUsage apply regardless of how the key is stored by the TOE, including when the key is externally stored (cf. section 1.3.1.2).

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*[65] on

(1) subjects: all

(2) objects: keys

(3) operations: backup, restore[66]*.*

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*[67] to objects based on the following:

(1) whether the subject is an administrator[68]*.*

[63] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[64] *rules, based on security attributes, that explicitly deny access of subjects to objects*]
[65] [assignment: *access control SFP*]
[66] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[67] [assignment: *access control SFP*]
[68] [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*]

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

(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 [69].

Application Note 25

Preserving the binding of a set of attributes to its key (in FDP_ACF.1.2/Backup (4)) means that it is not possible for the attributes to be changed during a backup operation, or by modification of the backup data while it is away from the TSF.

Backups may contain keys whose export flag attribute marks them as 'non-exportable'.

The ST author specifies the cryptographic operations used to protect confidentiality and integrity of any supported backups using one or more iterations of FCS_COP.1.

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*[70]

Application Note 26

If the TOE does not provide backup and restore operations then the Security Target shall include FDP_ACC.1/Backup and FDP_ACF.1/Backup but shall state in an Application Note for each of these SFRs that the relevant security requirements are trivially met because no backup facility is provided.

FDP_SDI.2	*Stored data integrity monitoring and action*
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring.
Dependencies:	No dependencies.

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

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

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for *integrity errors*[71] on all **keys (including security attributes)**[72], based on the following attributes: *integrity protection data*[73].

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[74].

Application Note 27

No specific requirement is placed here on the nature of the integrity protection data, but the Security Target shall describe this protection measure, and shall identify the iteration of FCS_COP.1 that covers any cryptographic algorithm used.

This SFR may also be used in the implementation of the mechanism for protection against modification access to the value of a secret key in FDP_IFF.1.5/KeyBasics, and in the requirement for export of public keys with integrity protection in FDP_IFF.1.2/KeyBasics.

The integrity protection data in FDP_SDI.2.1 is included in the list of attributes identified in FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys, and protects the value of the key and of its other security attributes, including when the key is externally stored by the TOE (cf. section 1.3.1.2).

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*[75] the following objects:

- *authorisation data*
- *secret keys*[76].

Application Note 28

Authorisation data is not to be stored persistently in the TOE; the refinements to ADV_ARC.1 in section 6.4.1 require the approach to minimising the time that this data is held before deallocation according to FDP_RIP.1.

7.3.4 Trusted path/channels (FTP)

FTP_TRP.1/Local	*Trusted Path*

[71] [assignment: *integrity errors*]
[72] objects
[73] [assignment: *user data attributes*]
[74] [assignment: *action to be taken*]
[75] [Selection: *allocation of the resource to, deallocation of the resource from*]
[76] [assignment: *list of objects*]

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1.1/Local	The TSF shall provide a communication path between itself and *local*[77] **client applications**[78] that is logically distinct from other communication paths and provides assured **authentication** [79] of its end points and protection of the communicated data from *modification and disclosure*[80].
FTP_TRP.1.2/Local	The TSF shall permit [selection: *the TSF, local client applications*][81] 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]* [82].

Application Note 29

FTP_TRP.1/Local shall be completed in a Security Target to identify the local client applications and to reflect the way that the TOE communicates with them, and to justify the security of this communication path. Where the TOE and local client applications are located within the physical boundary of the same hardware appliance (e.g. local applications running on a server and communicating with a PCI card on the server's internal PCI bus) then the trusted path may be mapped in the Security Target to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment).

If the TOE does not provide an interface for local client applications, then this SFR is not applicable and is trivially satisfied. This should be stated in the Security Target.

The TOE may provide other additional channels that provide only authentication and integrity protection (not confidentiality), in which case other iterations of FTP_TRP.1 may be added in the ST, allowing the selection of only modification protection in FTP_TRP.1.1 for these additional iterations.

The Security Target shall identify in an application note the iterations of FCS_COP.1 that provide any cryptographic functions that contribute to the implementation of the trusted path, and the SFRs that provide the authentication of the end points.

ST Application Note

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.

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

[78] users

[79] identification

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

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

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

FTP_TRP.1.1/External	The TSF shall provide a communication path between itself and *remote*[83] **external client applications** [84] that is logically distinct from other communication paths and provides assured **authentication**[85] of its end points and protection of the communicated data from *modification and disclosure*[86].
FTP_TRP.1.2/External	The TSF shall permit *remote **external client applications***[87] 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*[88] [89]

Application Note 30

FTP_TRP.1/External shall be completed in a Security Target to identify the external client applications and to reflect the way that the TOE communicates with them, and to justify the security of this communication path. The word "remote" in FTP_TRP.1.1/External and FTP_TRP.1.2/External refers to client applications that are described as "external" in the rest of this PP.

If the TOE does not provide an interface for external client applications, then this SFR is not applicable and is trivially satisfied. This should be stated in the Security Target.

The TOE may provide other additional channels that provide only authentication and integrity protection (not confidentiality), in which case other iterations of FTP_TRP.1 may be added in the ST, allowing the selection of only modification protection in FTP_TRP.1.1 for these additional iterations.

The Security Target shall identify in an application note the iterations of FCS_COP.1 that provide any cryptographic functions that contribute to the implementation of the trusted path, and the SFRs that provide the authentication of the end points.

ST Application Note

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.

7.3.5 Protection of the TSF (FPT)

FPT_STM.1	*Reliable time stamps*
Hierarchical to:	No other components.
Dependencies:	No dependencies
FPT_STM.1.1	The TSF shall be able to provide reliable time stamps.

Application Note 31

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

[84] users

[85] identification

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

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

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

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

The TOE shall provide timestamps suitable for supporting the time in an audit record for FAU_GEN.1. If the TOE provides additional timestamping services for client applications, or other record of the time of an operation for client applications, then these should be covered in one or more separate iterations of the SFR, with an Application Note added to define any specific requirement for reliability of the time information for that service.

FPT_TST_EXT.1	*Basic TSF Self Testing*

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests *during initial start-up (or power-on) and at the conditions defined below*[90] to demonstrate the correct operation of the TSF:

- *At initial start-up (or power-on):*
  - *Software/firmware integrity test*
  - *Cryptographic algorithm tests*
  - *Random number generator tests*
- *Conditional tests defined in Conditional Self-tests table*[91].

Table 9: 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.
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.
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.

[90] [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]*]

[91] [assignment: *list of self-tests run by the TSF*]

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

Application Note 32

Completion of the selection in FPT_TST_EXT.1.1 may be by 'None' (in which case the 'and' preceding the selection should be deleted and no selection text included). Completion of the list of additional tests in the final assignment may include tests performed at initial start-up (or power-on) and/or tests run under the conditions specified in the earlier selection and assignment. The term 'start-up' (or power-on) means that the tests should be executed at least any time that the TOE is powered-on.

The tests of the cryptographic functions shall include all cryptographic functions covered by FCS_COP.1. The Operational Guidance shall include a description of the errors that may arise from self-test and the actions that should be taken in response to each.

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.

Application Note 33

Passive detection of a physical attack is typically achieved by using physical seals and an appropriate physical design of the TOE that allows the TOE administrator to verify the physical integrity of the TOE as part of a routine inspection procedure.

Because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 for this TOE is equivalent to the physical security mechanisms for tamper detection and response required by section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3. (Cf. refinement of AVA_VAN.5 in section 6.4.1.)

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*[92] to *the entire TOE*[93] by responding automatically such that the SFRs are always enforced.

Application Note 34

This SFR is linked to the requirements for passive detection of physical attacks in FPT_PHP.1, and should identify the relevant responses of the TOE involved in meeting the key zeroization requirements of ISO/IEC 19790:2012 Security Level 3. As in the case of FPT_PHP.1, because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.3 for this TOE is equivalent to the level of assessment for this aspect of tamper detection and response required for section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements by each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3. (Cf. refinement of AVA_VAN.5 in section 6.4.1.)

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_EXT.1 fails*

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

(3) Failures of critical TOE hardware components (including the RNG) occur

(4) Corruption of TOE software occurs

(5) *none*[94] [95].

Application Note 35

The Operational Guidance shall include a description of the specific failures that are detected (e.g. the thresholds for environmental conditions, and the nature of the monitoring of specific critical TOE hardware components), how these failures are notified, and the actions that should be taken in response to each.

7.3.6 Security management (FMT)

For the purposes of specifying a minimum set security attributes of keys, and the constraints on initialisation and modification of these attributes in FMT_MSA.1 and FMT_MSA.3, two separate types of keys are defined: Assigned Keys (defined and 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

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

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

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

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

attributes below) and, since they are intended for use within the TOE, because they cannot be imported or exported[96]. 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[97].

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.1	*Security roles*
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.

[96] 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.

[97] 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.1.1	The TSF shall maintain the roles *Administrator, External Client Application*[98]*, Key User, none*[99] [100].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

Application Note 36

The Local Client Application role represents an identifiable subject that communicates locally with the TOE, i.e. within the same hardware appliance. The External Client Application role represents an identifiable subject that communicates remotely with the TOE over a secure channel. A TOE can support one or both types of Client Applications.

The Key User role represents a normal, unprivileged subject who can invoke operations on a key according to the other authorisation requirements for the key – this role may sometimes act through a client application.

ST Application Note

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. All three can be considered Administrator according to the PP terminology.

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) backup and restore functions[101]

(5) key import function[102]

(6) key export function[103] [104]

Application Note 37

The unblocking of authentication or authorisation failures in FMT_SMF.1.1 (1) is related to the authentication failures described in FIA_AFL.1. The attributes of keys in FMT_SMF.1.1 (2) correspond to the attributes in

[98] *[selection: Local Client Application, External Client Application]*
[99] [*assignment*: *list of additional authorised identified roles*]
[100] *[assignment: the authorised identified roles]*

[101] *[selection: backup and restore functions, no backup and restore functions]*
[102] *[selection: key import function, no key import function]*
[103] *[selection: key export function, no key export function]*
[104] *[assignment: list of management functions to be provided by the TSF].*

FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys. Export of audit data in FMT_SMF.1.1 (3) relates to the ability to export audit data from the TOE for preservation and storage elsewhere. The selections in FMT_SMF.1.1 (4), (5) and (6) identify whether or not the TOE provides the relevant functions (and shall therefore correspond to the relevant statements in the ST for FDP_IFF.1.2/KeyBasics, FDP_ACC.1/Backup and FDP_ACF.1/Backup.

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*[105] the User accounts[106] to **automatic processes** [~~assignment: *the authorised identified administrative roles*~~][107].

Application Note 38

The list of TSF data assigned shall correspond to the relevant data blocked by authentication or authorisation failures according to the associated iteration(s) of FIA_AFL.1. For the purposes of unblocking, the TSF data in the assignment includes any key that can be affected by blocking due to failure of authorisation (as in FIA_UAU.6), as well as user accounts (as in FIA_UAU.1) blocked by authentication/authorisation failures.

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

ST Application Note

As it is defined in FIA_AFL.1/Admin FIA_AFL.1/User and in FIA_AFL.1/Key owner 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 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

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

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

[107] refinement: [assignment: *the authorised identified administrative roles*]

FMT_MTD.1.1/AuditLog The TSF shall restrict the ability to *control export and deletion of*[108] the *audit log records*[109] to *the Administrator role*[110].

Application Note 39

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

ST Application Note

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*	
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*[111] to restrict the ability to *modify*[112] the security attributes *as specified in the Key Attributes Modification Table*[113] [114] to *subjects, objects, and operations among subjects and General Keys, as specified in the Key Attributes Modification Table*[115] [116].

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

[108] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[109] [assignment: *list of TSF data*]
[110] [assignment: *the authorised identified roles*]
[111] [assignment: *access control SFP(s), information flow control SFP(s)*]
[112] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[113] [assignment: *list of security attributes , to include attributes as specified in the Key Attributes Modification Table]*
[114] [assignment*: list of security attributes]*
[115] [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]*
[116] [assignment: the authorised identified roles]

FMT_MSA.1.1/AKeys The TSF shall enforce the *Key Usage SFP*[117] to restrict the ability to *modify* [118] the security attributes *as specified in the Key Attributes Modification Table*[119] [120] to *subjects, objects, and operations among subjects and Assigned Keys specified in the Key Attributes Modification Table*[121] [122].

Application Note 40

The Key Attributes Modification Table is referenced from FMT_MSA.1/GenKeys, and FMT_MSA.1/AKeys. The required constraints on security attribute modification specified in this PP are shown in Table 1; the Security Target completes the other parts not specified here (along with any other information for other security attributes relevant to a particular TOE). The specific attributes used by a particular TOE may vary, but the Security Target shall make clear how control is achieved over the ability to modify attributes of keys in terms of the specific attributes and controls imposed by the TOE. Where applicable to the operational environment for a particular TOE, these controls should be described with reference to the ways that they are used to provide qualified electronic signatures and qualified electronic seals that meet the requirements of [Regulation] (cf. the refinement to AGD_OPE.1 in section 6.4.1).
Where a TOE does not support one of the individual types of key then the Security Target states this, and the requirements for that type of key are considered to be trivially satisfied.

Authorisation Data and Re-authorisation conditions are required for secret keys only. Re-authorisation conditions include the conditions specified for FIA_UAU.6.1/KeyAuth (matching the assignments and selections made for that SFR in the Security Target).

Table 10: Key Attributes Modification Table[123]

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	Modified only when modification operation includes successful	no authorisation Data

[117] [assignment: *access control SFP(s), information flow control SFP(s)*]
[118] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[119] [assignment: *list of security attributes*]
[120] [assignment: *list of security attributes, to include attributes as specified in the Key Attributes Modification Table]*
[121] [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*]
[122] [assignment: the authorised identified roles]
[123] 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 '---') 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).

	validation of current (pre-modification) authorisation data	validation of current (pre-modification) authorisation data, or by an Administrator	
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
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*[124] to provide *restrictive*[125] 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*[126] to specify alternative initial values to override the default values when an object or information is created.

Table 11: 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
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

Application Note 41

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

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

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

The Key Attributes Initialisation Table is referenced from FMT_MSA.3/Keys and matches the attributes covered by the separate iterations of FMT_MSA.1 above. The required constraints on security attribute initialisation specified in this PP are shown in Table 2; the Security Target completes the other parts not specified here (along with any other information for other security attributes relevant to a particular TOE). The specific attributes used by a particular TOE may vary, but the Security Target shall make clear how control is achieved over the ability to modify attributes of keys in terms of the specific attributes and controls imposed by the TOE. Where applicable to the operational environment for a particular TOE, these controls should be described with reference to the ways that they are used to provide qualified electronic signatures and qualified electronic seals that meet the requirements of [Regulation] (cf. the refinement to AGD_OPE.1 in section 6.4.1).

Where a TOE does not support one of the individual types of key then the Security Target states this, and the requirements for that type of key are considered to be trivially satisfied.

Authorisation Data and Re-authorisation conditions are required for secret keys only, and only as described in the assignments and selections made in the Security Target for FIA_UAU.6/KeyAuth.

Attributes assigned by the TOE to any imported keys shall be described in the Security Target and in operational user guidance (see the refinements to AGD_OPE.1 in section 6.4.1), noting that a secret key can only be imported if it is a non-Assigned key (cf. FDP_IFF.1/KeyBasics).

The Integrity Protection Data for a key is used to support FDP_SDI.2 and covers not only the key but also its other attributes.

7.3.7 Security audit data generation (FAU)

FAU_GEN.1	*Audit data generation*
Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps

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

a) Start-up and shutdown of the audit functions;

b) All auditable events for the *not specified*[127] level of audit; and[128]

c) *Startup of the TOE;*

d) *Shutdown of the TOE*

e) *Cryptographic key generation (FCS_CKM.1);*

f) *Cryptographic key destruction (FCS_CKM.4);*

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

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

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

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

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

s) *none*[129].

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*[130].

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

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

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

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

Application Note 42

The Security Target is not required to identify separate levels of audit in FAU_GEN.1.1. However, the Operational Guidance is required to describe any configuration or other actions that apply to audit functions, and to make clear, in cases where logging of particular audit events is optional, how to ensure that any individual audit event is logged. Default logging actions of the TOE shall also be described in Operational Guidance.

The Administrative Actions logged need not be limited to those related to FMT SFRs: other administrative actions affecting the operation of SFRs should also be included (and listed as part of the assignment in FAU_GEN.1.1).

FAU_GEN.2	*User identity association*
Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification
FAU_GEN.2.1	For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_STG.2	*Guarantees of audit data availability*
Hierarchical to:	FAU_STG.1 Protected audit trail storage
Dependencies:	FAU_GEN.1 Audit data generation
FAU_STG.2.1	The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
FAU_STG.2.2	The TSF shall be able to ***prevent***[131] unauthorised modifications to the stored audit records in the audit trail.
FAU_STG.2.3	The TSF shall ensure that *all*[132] stored audit records will be maintained when the following conditions occur: *audit storage exhaustion*[133].

Application Note 43

The Operational Guidance is required to describe any use that the TOE makes of an external audit server, the situation regarding records held locally on the TOE and those held externally on an audit server (e.g. the TOE might accumulate records locally before transferring them to an external audit server), and the way in which audit records are maintained when local audit storage is exhausted (including description of the actions taken by the TOE when audit storage exhaustion is detected). The Operational Guidance shall describe the protection applicable to all records created by the TOE (in order to provide prevention or detection of unauthorised modifications as in FAU_STG.2.2), and shall identify any obligations for the environment in maintaining audit trail protection. The expectation is that this will comprise cryptographic methods of prevention or detection of unauthorised modification (including deletion) of audit records.

Control over export and deletion of the audit log records is limited to the Administrator role as specified in FMT_MTD.1/AuditLog.

ST Application Note

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

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

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

Internal audit logs can be collected by Administrators and exported to external drives via USB. 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.

7.4 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 12: 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**)

7.4.1 Refinements of Security Assurance Requirements

The following refinements are made to selected assurance requirements in Table 12:

ADV_ARC.1 Security architecture description

Refinement:

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

1. In general cryptographic modules will make use of 'support keys' as part of their implementation of protection mechanisms, where these keys are generally not held on behalf of specific users[134] or client applications, but are used by the TOE to carry out its normal operations and as part of the implementation mechanism other SFRs and to protect the TSF itself. These support keys may be used for a variety of purposes (including aspects such as authentication, authorisation, secure channels, security of external storage, or internal data protection), For the purposes of this PP, support keys used by the TOE are treated as TSF data, and require a specific security rationale to be included as part of the ADV_ARC.1 deliverables. This rationale shall include a description of the key architecture, identifying all support keys used by the TOE (at least in its evaluated configuration), their method of generation and storage, their purpose in TOE operation, and the ways in which they are protected so as to support the requirements of FDP_IFF.1/KeyBasics and FDP_ACF.1/KeyUsage (noting that the mechanisms used for support keys may differ from those used for user keys). Examples would be keys used for wrapping user keys in order to allow secure storage of the user keys, keys used to implement secure channels, and keys used to protect backups. The description shall demonstrate that sufficient entropy has been used in the generation of each support key, and the source of that entropy. The rationale shall demonstrate that these support keys cannot be exported/imported in a way that threatens the secure operation of the TOE. The evaluator shall include the description of the support keys in their analysis of the protection of user data (e.g. to confirm that it does not introduce vulnerabilities in the implementation of the SFRs).
2. If updates to the TOE software or firmware are supported then the ADV_ARC.1 deliverables shall describe how the TOE is protected against unauthorised updates, by using digital signatures. This shall be confirmed by evaluator testing (if updates are supported) to confirm that updates with invalid signatures are rejected without being executed. The digital signature algorithms used to protect updates shall be included in the scope of FCS_COP.1 signature SFR(s).
3. The ADV_ARC.1 deliverables shall in particular describe
    a. Any use that the TOE makes of an audit server

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

b. The locations used for any externally stored keys and the structure and format of the externally stored keys including the cryptographic structures that protect the keys in their externally stored form, and that bind them to their attributes (support keys are separately addressed by the description required in item 1 above)
c. All key import and/or export functions and the secure channels that they use
d. The secure channels supported by the TOE and the authentication mechanisms that they use (cf. FTP_TRP.1/Local & FTP_TRP.1/External)
e. All local and external interfaces used for communications with users, client applications, audit data, and stored TOE data (cf. Figure 1)
f. The specific key attributes supported, their method of representation (e.g. the relevant data structures and permitted values) and the method by which they are bound to the corresponding key value (cf. FMT_MSA.1). This also includes identifying the types of keys
(if any) that support re-authorisation conditions described in FIA_UAU.6/KeyAuth
g. The user types and roles supported, the interfaces by which they interact with the TOE (e.g. a local administrator console or an externally available API), the authentication methods used (cf. FIA_UAU.1 and Application Note 17), and any privileges available to the user type/role
h. All of the cryptographic functions provided (cf. section 1.3.1.1) and whether any non-endorsed cryptographic algorithms and/or cryptographic functions are available (cf.
FCS_COP.1 and section 1.3.1.3)
i. The authorisation methods used for keys (cf. FIA_UAU.6/KeyAuth & FDP_ACC.1/KeyUsage)
j. Description of the way in which the TOE ensures that it only holds authorisation data for the minimum time possible before deallocating it according to FDP_RIP.1
k. If the TOE provides backup operations then the ADV_ARC deliverables shall describe the use of support keys by the backup and restore processes (cf. FDP_ACF.1/Backup), and in particular shall describe the ways in which confidentiality and integrity of the backup are provided, and the way in which the TOE rejects an attempt to carry out a restore process using backup data that has been modified
l. Any mechanisms that the TOE uses to support dual person control (cf. FDP_ACF.1/Backup).

AGD_OPE.1 Operational user guidance

Refinement:

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

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

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

2. The use of trusted channels (cf. FTP_TRP.1/Local & FTP_TRP.1/External).
3. The available key attributes, their possible values, and the meaning of each of these values (cf. FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys, including their use to constrain the period and number of uses that are enabled by authorisation of a key (cf. FIA_UAU.6/KeyAuth and Application Note 19).
4. Identification of any non-endorsed cryptographic algorithms and/or cryptographic functions that are available (cf. FCS_COP.1 and section 1.3.1.3).
5. Identification of any other cryptographic algorithms and operations that are not included in the scope of the Security Target.
6. Possible errors from the self-test process and the actions that should be taken in response to each (cf. FPT_TST_EXT.1 & Application Note 32).
7. Specific failures detected by the TOE (cf. FPT_FLS.1 & Application Note 35).
8. Audit functions and their configuration (including specification of the available audit records), along with any other actions that are associated with audit functions (e.g. archiving or viewing audit records, or use of an external audit server) (cf. FAU_GEN.1 & Application Note 42, FAU_STG.2 & Application Note 43, FMT_MTD.1/AuditLog & Application Note 39).
9. Any configuration and operation requirements for dual-control operations (cf. FDP_ACF.1/Backup).
10. If backup is provided by the TOE (cf. FDP_ACF.1/Backup), then the Operational Guidance shall describe the backup and restore functions, and the administrator roles that are required to carry them out.
11. If key import is provided by the TOE, then the Operational Guidance shall describe how attributes are defined for any imported keys (cf. FMT_MSA.3/Keys). The evaluators shall test the import process to demonstrate that the description in the Operational Guidance is suitably complete, and that the keys imported have attributes appropriately defined. Similarly if key export is provided by the TOE then the Operational Guidance shall describe whether attributes are exported with keys (and if so, then how the attributes are represented and associated with the exported key), and the evaluators shall test the export process to demonstrate that the description in the Operational Guidance is suitably complete, and that the handling of attributes is as described.

ATE_IND.2 Independent testing – sample

Refinement:

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

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

AVA_VAN.5 Advanced methodical vulnerability analysis

Refinement:

Regarding the protection of the TOE against physical attacks: because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 and FPT_PHP.3 for this TOE is equivalent to the

level of assessment in section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3.

8 Rationales

8.1 Security Objectives Rationale

8.1.1 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 13: 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 in section 3.5 is directly matched by a security objective for the operational environment in section 4.2. The wording of each objective for the operational environment includes the wording of each assumption, and no further rationale is therefore given here.

8.2 7.2 Security Requirements Rationale

8.2.1 Security Requirements Coverage

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

Table 14: TOE Security Objectives mapping to SFRs

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FCS_CKM.1		X												
FCS_CKM.4	X													
FCS_COP.1		X												
FCS_RNG.1											X			
FIA_UID.1				X										
FIA_UAU.1				X										
FIA_AFL.1/Admin				X										
FIA_AFL.1/User				X										

FIA_AFL.1/Key owner				X										
FIA_UAU.6/KeyAuth				X		X								
FDP_IFC.1/KeyBasics	X				X				X					
FDP_IFF.1/KeyBasics	X		X		X				X					
FDP_ACC.1/KeyUsage					X	X								
FDP_ACF.1/KeyUsage					X	X								
FDP_ACC.1/Backup										X				
FDP_ACF.1/Backup										X				
FDP_SDI.2			X											
FDP_RIP.1	X				X									
FTP_TRP.1/Local			X	X			X	X	X					
FTP_TRP.1/External			X	X			X	X	X					
FPT_STM.1														X
FPT_TST_EXT.1													X	
FPT_PHP.1												X		
FPT_PHP.3												X		
FPT_FLS.1													X	
FMT_SMR.1				X										X
FMT_SMF.1				X										X
FMT_MTD.1/Unblock				X										
FMT_MTD.1/AuditLog														X
FMT_MSA.1/GenKeys					X									
FMT_MSA.1/AKeys					X									
FMT_MSA.3/Keys					X									
	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FAU_GEN.1														X
FAU_GEN.2														X
FAU_STG.2														X

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

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

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

OT.Auth is addressed by FIA_UID.1, FIA_UAU.1, FIA_AFL.1/Admin, FIA_AFL.1/User and FIA_AFL.1/Key owner for administrator authentication (with FMT_MTD.1/Unblock and its dependencies on FMT_SMR.1 and FMT_SMF.1 ensuring that appropriate roles and unblocking for 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_EXT.1 and secure failure requirements of FPT_FLS.1.

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

8.2.2 SFR Dependencies

The dependencies between SFRs are addressed as shown in Table 15. Where a dependency is not met in the manner defined in [CCP2] then a rationale is provided for why the dependency is unnecessary or else met in some other way.

Table 15: SFR Dependencies Rationale

Requirement	Dependencies	Fulfilled by
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1 FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1 See also note below on key attributes during import or export.
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1 FCS_CKM.4 See also note below on key attributes during import or export.
FCS_RNG.1	No dependencies	
FIA_UID.1	No dependencies	
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_AFL.1/Admin	FIA_UAU.1	FIA_UAU.1
FIA_AFL.1/User	FIA_UAU.1	FIA_UAU.1
FIA_AFL.1/Key owner	FIA_UAU.1	FIA_UAU.1
FIA_UAU.6/KeyAuth	No dependencies	
FDP_IFC.1/KeyBasics	FDP_IFF.1	FDP_IFF.1/KeyBasics
FDP_IFF.1/KeyBasics	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/KeyBasics FMT_MSA.3/Keys
FDP_ACC.1/KeyUsage	FDP_ACF.1	FDP_ACF.1/KeyUsage
FDP_ACF.1/KeyUsage	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/KeyUsage FMT_MSA.3/Keys
FDP_ACC.1/Backup	FDP_ACF.1	FDP_ACF.1/Backup
Requirement	**Dependencies**	**Fulfilled by**

FDP_ACF.1/Backup	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Backup The dependency on FMT_MSA.3 is not relevant in this case since the attribute used in FDP_ACF.1/Backup is determined by the ability of the user to authenticate as an administrator according to FIA_UAU.1.
FDP_SDI.2	No dependencies	
FDP_RIP.1	No dependencies	
FTP_TRP.1/Local	No dependencies	
FTP_TRP.1/External	No dependencies	
FPT_STM.1	No dependencies	
FPT_TST_EXT.1	No dependencies	
FPT_FLS.1	No dependencies	
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FMT_SMF.1	No dependencies	
FMT_MTD.1/Unblock	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1/AuditLog	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/GenKeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/AKeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
FMT_MSA.3/Keys	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/GenKeys, FMT_MSA.1/AKeys FMT_SMR.1
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1 FIA_UID.1
FAU_STG.2	FAU_GEN.1	FAU_GEN.1

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

Application Note 41 (for FMT_MSA.1) also requires that the initialisation of any attributes on import is described in the Security Target.

8.2.3 Rationale for SARs

The assurance level for this protection profile is **EAL4 augmented with 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.

8.2.4 AVA_VAN.5 Advanced methodical vulnerability analysis

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

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 four different roles in Primus HSM. Genesis, Security Officer, Partition Security Officer and User (client application). Genesis, Security Officer (SO) and Partition Security Officer (Partition SO) are considered the Administrators of the TOE. Users represent the remote client applications accessing the TOE via its API.

Administrators

The Administrators (Genesis, SO and Partition SO) 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.

Users

Security Officers can create new users (partitions). At creation, an identity 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 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

In case of SKA key 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 during this time.

Whenever a User tries to use one of its private keys a re-authentication is needed.

Related SFRs: FIA_UID.1, FIA_UAU.1, FIA_UAU.6/KeyAuth, FIA_AFL.1/Admin, FIA_AFL.1/User, FIA_AFL.1/Key owner

9.2 Key Management

The TOE handles System keys and user keys as described in Table 6: Critical Security Parameters (CSPs).

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 Table 7: 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-140-2 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 and who can change the authorisation rules. With SKA Keys it is possible to identify the Signer (key owner not the client application).

Related SFRs: FCS_CKM.1, FCS_CKM.4, 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

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 Table 7: Cryptographic Algorithms table.

Destroying keys are according to FIPS-140-2 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.4, 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) User (External client application). Details can be found in sec 8.1 Authorisation, and in sec 2.4.2.6 Available services by roles.

Key Users (key owner) are identified by a certified SAM according to [EN 419241-2] outside the TOE or can be identified by the TOE if the client application uses SKA keys. SKA keys allow the TOE to identify the key owner itself, not only the client application. The details of SKA keys can be found in the Key Management section.

The management functions for the Administrators are collected in Table 8: Authorized Services.

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 (NTP configuration available), 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 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.

Related SFRs: FMT_SMR.1, FMT_SMF.1, FMT_MTD.1/Unblock, FMT_MTD.1/AuditLog, FAU_GEN.1, FAU_GEN.2, FAU_STG.2, FPT_STM.1

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 Table 7: 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 Keyfiles 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 Table 9: Conditional Self-tests.

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 140-2 Level 3 compliant.

Related SFRs: FDP_SDI.2, FDP_RIP.1, FTP_TRP.1/External, FPT_TST_EXT.1, FPT_PHP.1, FPT_PHP.3, FPT_FLS.1

10 Bibliography

[CCP1] Common Criteria for Information Technology Security Evaluation,
Part 1: Introduction and general model,
Version 3.1, Revision 5, April 2017
CCMB-2017-04-001

[CCP2] Common Criteria for Information Technology Security Evaluation,
Part 2: Security functional requirements,
Version 3.1, Revision 5, April 2017
CCMB-2017-04-002

[CCP3] Common Criteria for Information Technology Security Evaluation,
Part 3: Security assurance requirements,
Version 3.1, Revision 5, April 2017
CCMB-2017-04-003

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

[CEN] 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

[AIS20] BSI: Application Notes and Interpretation of the Scheme (AIS) 20 – Functionality classes and evaluation methodology for deterministic random number generators, Version 1 (02.12.1999)

11 Acronyms

CC	Common Criteria
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
EAL	Evaluation Assurance Level
HSM	Hardware Security Module
IT	Information Technology
JCA/JCE	Java Cryptography Architecture, Java Cryptography Extension. Crypto libraries for java
KEK	Key encryption key
MS CSP	Microsoft Cloud Solution Provider
PCIe	Peripheral Component Interconnect Express
PKCS#11	Public-key Cryptography Standards
PKI	Public Key Infrastructure
PP	Protection Profile
RNG	Random Number Generator
SAR	Security assurance requirements
SFP	Security Function Policy
SFR	Security functional requirements
SKA	Smart Key Attributes
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	Trust Service Provider