SECURITY TARGET

Trident,

the distributed remote Qualified Signature Creation Device

Table of Contents

List of Figures

List of Tables

1. ST Introduction

1.1 ST reference

ST reference: Trident-ST

ST version: 4.1

ST date: 19 March 2025

CC version 3.1, revision 5

Assurance level: EAL4 augmented by AVA_VAN.5 and ALC_FLR.3

ST author: i4p informatikai kft. (i4p informatics ltd.)

1.2 TOE reference

The TOE reference is Trident version 3.2.3

There are eight models of Trident: A11, A21, A31, A33, B11, B31, B33, C16.

- In case of all models, except C16: the TOE reference is displayed on the LCD screen of the Multi-Party Cryptographic Appliance (MPCA) as "TRIDENT v*[version number]*" with the serial number and the model number. The serial number is also printed on a sticker on the device enclosure.
- In case of model C16: the text "Trident RSS on Trustway Proteccio" is printed on the front of the MPCA. The serial number is also printed on a sticker on the device enclosure.

After starting the appliance, the serial number, the Trident version and the model number is displayed in the welcome message as:

> Welcome to MPCM, the world's first true multi-party PKI solution
>
> Machine serial number: *[serial number]*
>
> Trident version: *[version number]*
>
> Model number: *[model number]*

1.3 TOE overview

1.3.1 TOE type

The Trident is a multi-user, multi-key device. The Trident is composed of two main components which can work together to fulfill different sets of requirements:

- The Cryptographic Module (CM) component of the Trident is a general-purpose cryptographic module suitable for cryptographic support needed by its legitimate users (e. g. service providers supporting local or remote electronic signature and electronic sealing operations, certificate issuance and revocation, time stamp operations and authentication services).
- The Signature Activation Module (SAM) component of the Trident is a local application deployed within the tamper protected boundary of the Trident and implements the Signature Activation Protocol (SAP). It uses the Signature Activation Data (SAD) from a remote signer to activate the corresponding signing key for use in a cryptographic module.

1.3.2 TOE usage

The Trident is a QSCD and is suitable for both ("Local" and "Remote") use cases of [EN 419221-5] Protection Profile.

1.3.2.1 The "Local" use case

This use case (see *Figure 1.1* and 4.4.2.2 Use Case 1: Local signing in [EN 419221-5]) is aimed at local key owners applying their own electronic signatures or seals. In this use case only the CM functionality of the TOE is used, which performs local cryptographic operations, and associated key management. These operations can be used by a client application to create qualified and non-qualified electronic signatures and electronic seals for the local key owner natural or legal person. Examples include TSPs issuing certificates and time-stamps, as well as supporting application services such as e-invoicing and registered e-mail where the service provider applies its own seal or signature.

In this use case the local key owner is responsible for the security of the environment in which the Trident is used and managed. In this use case the Trident generates, stores and uses only keys that belong to and represent the local end entity, apart from its infrastructural support keys (used in internal protection mechanisms).

The Trident provides its own development API (called CMAPI enabling the easy integration with a wide range of applications) and other well-known APIs (eg. the PKCS#11 and OpenSSL API).

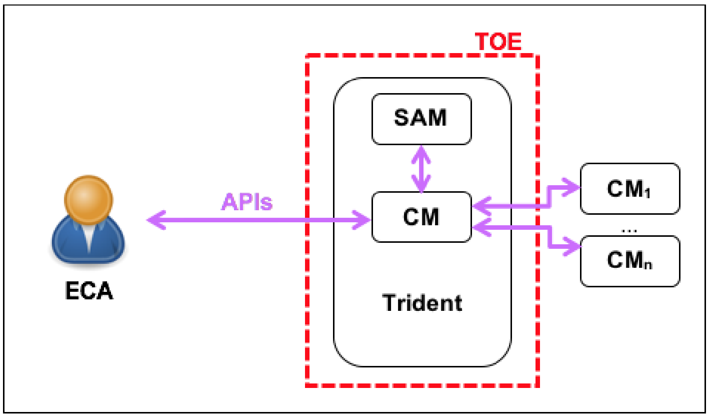

Figure 1.1 The TOE in the "Local" use case

1.3.2.2 The "Remote" use case

This use case (see *Figure 1.2* and 4.4.2.3 Use case 2: Support for Remote Server Signing in [EN 419221-5]) is aimed at TSPs supporting requirements for remote signing, or sealing, as specified in [eIDAS]. In this case the inbuilt CM and the SAM functionality of the Trident together meets the requirements for QSCDs in the context of remote signing set out in Annex II of [eIDAS].

The SAM functionality of the Trident meets the requirements for Sole Control Assurance Level 2 as defined in [EN 419241-1].

In this use case the CM functionality of the Trident performs cryptographic operations, and associated key management, which can be used by an application using server signing, as defined in [EN 419241-1], to create qualified electronic signatures and qualified electronic seals on behalf of a legal or natural person which is distinct from and remote from the TSP which manages the Trident.

The CM functionality of the Trident generates, stores and uses signing, sealing keys in a way that maintains the remote control of an identified signatory or seal creator who operates through the use of a client application. The CM functionality of the Trident deals with ensuring the security of keys and their use for signature or seal creation.

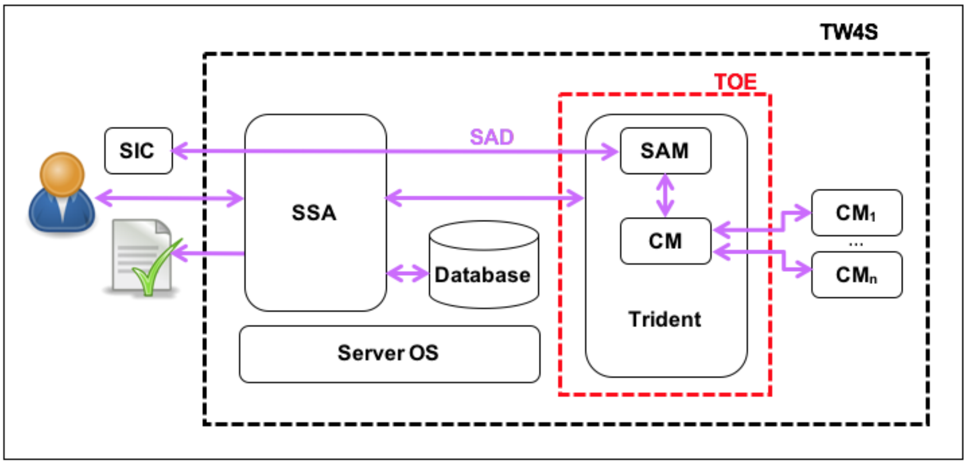

Figure 1.2 The TOE in the "Remote" use case

The Signer's Interaction Component (SIC) is a piece of software and/or hardware, operated on the signer's environment under its sole control.

The Server Signing Application (SSA) uses the Trident in order to generate, maintain and use the signing keys.

The Signature Activation Protocol (SAP) allows secure use of the signing key for the creation of a digital signature to be performed by a Cryptographic Module (CM part of the Trident) on behalf of a signer. The use of the Signature Activation Data (SAD), which is the essential part of the SAP, ensures control over the signer's key.

The Signature Activation Module (SAM) is a software part of the Trident, which uses the SAD in order to guarantee with a high level of confidence that the signing keys are used under sole control of the signer.

The Cryptographic Modules (CM part of the Trident) implement the main security functions, including cryptographic algorithms and key generation.

Signature activation for the Trident is the following:

- Signing key confidentiality and integrity are ensured by the CM part of the Trident (located in a tamper protected environment).
- The Trident (SAM + CM) are under control of the SSA.
- The SAM part of the Trident participates in SAP and ensures that the signature operation is under the legitimate signer's control.
- The SSA interfaces via a secure channel the SAM which verifies the SAD in order to activate the corresponding signing key.
- The signer authentication can remain for a given period and/or for a given number of signatures.
- SAD computation shall be done for each signature operation, but the SAD may be linked to

a set of DTBS/R, this allows the SSA to be used for bulk/batch signature purposes.

- Signer authentication is done using the SIC creating a link between the signer and the signature as part of the SAD.
- The SAD is transferred securely from the SIC to the SAM for verification.

1.3.3 Major security features of the TOE

The Trident can provide both SAM and CM functionality. In the distributed configuration different parts of the Trident implement secure multi-party computation (MPC) protocols.

1.3.3.1 CM functionality

Based on its CM component the Trident is a cryptographic module. CM functionality includes but is not limited to:

- generating, storing, using, backing up, restoring and destructing symmetric (e.g. AES, 3DES, ARIA, SEED), asymmetric (e.g. RSA, ECC) and post-quantum (e.g. SPHINCS+) keys,
- ensuring the security (confidentiality and integrity) of symmetric (e.g. AES, 3DES, ARIA, SEED) keys, asymmetric (e.g. RSA, ECC) and post-quantum (e.g. SPHINCS+) private keys, and pre-generated primes for RSA key pairs,
- performing post-quantum key-encapsulation mechanisms (e.g. NTRU, ML-KEM (Kyber)),
- creating qualified electronic signatures and electronic seals,
- performing additional supporting cryptographic operations, such as creation of non-qualified electronic signatures and seals, verification of electronic signatures and seals, cryptographic hash function, keyed-hash message authentication, encryption and decryption, key derivation, key-encapsulation, TOTP verification, JWT token verification,
- supporting of authentication of client applications or authorised users of secret keys, and support of authentication for electronic identification, as identified by [eIDAS],
- allowing the key owners to use TOTP one-time-passwords or JWT tokens when activating their keys,
- allowing the key owners to have a common secret key (MOSK, multi owner single key), activation of which all of the owners' successive authorization is needed.

The cryptographic services/functions above are available for ECAs and LCAs through an API.

The CM functionality of the Trident allows to use external Cryptographic Modules (based on a configuration parameter).

In this case some keys are generated, stored and used by an external CM configured to be used. The CM does not perform cryptographic operations but invokes the external CM with appropriate parameters whenever a cryptographic operation is required. This invocation is performed through a Local Client Applications (CMbr on *Figure 1.5*) using Standard PKCS#11 API. Using external CM functionality is not part of the TOE evaluated/certified configuration.

1.3.3.2 SAM functionality

Based on its SAM functionality Trident ensures that the remote signer has sole control of his signature keys, according to [EN 419241-1] SCAL2 for qualified signatures.

SAM functionality includes but is not limited to:

- authenticating the remote signer based on two authentication factors (a password and a one-time-password calculated from a shared secret or using delegated authentication),
- authorising the signature operation,
- activating the signing key within the internal CM.

SAM and the signer (via the SIC) communicate in order to generate the SAD. The SAD binds together signer authentication with the signing key and the data to be signed (DTBS/R).

Using the SAM functionality is optional: the SAM functionality of the Trident can also be performed by an External Client Application, using CM APIs (see *Figure 1.1*).

1.3.3.3 Additional functionality

The security features expected by [EN 419221-5] and [EN 419241-2] are complemented by with the following functionalities:

1.3.3.3.1 *Distributed functionality*

In case of distributed configuration, the Trident consists of n (n=2, 3 or 4) identical TOE parts (Multi-Party Cryptographic Appliances or MPCAs) to operate as a logical whole in order to fulfill the requirements of this Security Target (see *Figure 1.3 TOE in distributed configuration (the number of TOE parts could be 2, 3 or 4)*).

The user sends to one (any) of the TOE parts the full input (request), and later receives back the output (reply), exactly as in the standalone configuration. It is an active-active configuration.

In case of distributed configuration, the Trident supports three types of key generation:

1. Non-distributed (symmetric and asymmetric) key generation with mirroring

   The key is generated in one of the MPCAs, then is mirrored into the others.

   Advantage: the TOE parts (nodes) together provide a high availability cluster for redundancy and fault tolerance.

2. Distributed (symmetric and asymmetric) key generation with a trusted dealer

   The key is generated in one of the MPCAs, then the shares of the key are distributed to the other MPCAs.

   Advantage: providing secret sharing (a single MPCA never stores the whole key) much faster than without a trusted dealer

3. Distributed asymmetric key generation without a trusted dealer

   The MPCAs jointly generate key pairs so that at the end of the generation (1) public key is publicly known, (2) each MPCA holds only a share of the private key and (3) crypto operation will be impossible in the circumstance where less than all MPCAs are present.

   Advantage: providing advanced secret sharing (a single MPCA never knew and never knows, neither processes, nor stores the whole key).

Trident ensures the consistency among the MPCAs (e.g. their databases, internal states).

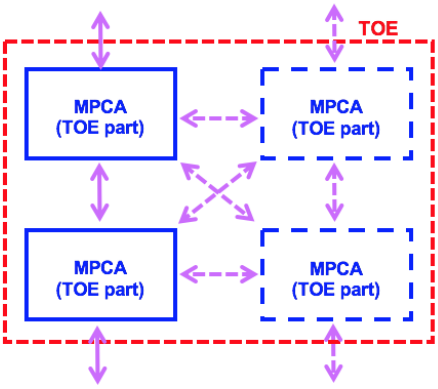

Figure 1.3 TOE in distributed configuration (the number of TOE parts could be 2, 3 or 4)

If some of the n (n=2, 3 or 4) MPCAs become dysfunctional, the remaining intact MPCAs (if there are any) can ensure a limited functionality.

In case of standalone configuration (n=1), the Trident consists of only one MPCA, and that alone fulfills the requirements of this Security Target (but of course cannot offer the additional services described in 6.1.4 and 7.1.8).

1.3.3.3.2 Active-passive node clustering

Besides the distributed configuration with non-distributed key generation, Trident can also provide high-availability mode by arranging one active (online) and one or more passive (standby) nodes into a cluster. A standby node is brought online when its associated active node fails, and can also be configured to provide further limited functionality.

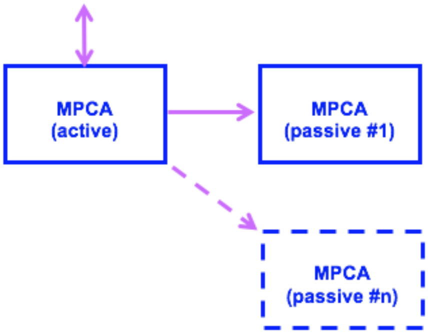

Figure 1.4 Trident in active-passive configuration. There can be one active and n passive nodes in a cluster.

Only the active MPCA can be called using database modification operations (like create, update and delete type commands). In this case, the active MPCA executes the command and synchronizes the modified database with the passive MPCA(s).

Depending on the configuration, there are two types of passive MPCAs:

- “standby”: It does not execute any external commands, but constantly synchronizes its database with the active MPCA. In case of failure of the active MPCA, it can immediately take over the active node role.
- “extended standby”: Additional to the standby functionality it is also able to execute sign commands which does not require database modification, while being in standby state.

The active-passive node clustering is not part of the TOE evaluated/certified configuration.

1.3.3.3.3 Trusted Update

Trident ensures the authenticity and integrity of software and firmware updates.

1.3.3.3.4 Quantum-resistant public-key cryptographic algorithms

Finally, Trident supports quantum-resistant public-key cryptographic algorithms, such as the SPHINCS+ digital signature algorithm [SPHINCS+], the ML-DSA digital signature algorithm [FIPS 204] the NTRU [NTRU] and the ML-KEM key-encapsulation mechanism [Kyber] and [FIPS 203].

1.3.4 Required non-TOE hardware/software/firmware

The following hardware, firmware and software supplied by the IT environment are excluded from the TOE boundary (see *Figure 1.1*):

- Signer's Interaction Component (SIC) used locally by the signer to communicate with the remote systems.
- Server Signing Application (SSA) that handles communications between SAM in the Trident and SIC in the signer device.
- Signature Creation Application (SCA) that manages the document to be signed and transfers that to the SSA through the SIC.
- External Client Applications (ECAs) which can use the cryptographic services of the Trident, including:
    - Certificate Generation Application (CGA) that issues signer certificates, or
    - other SAM used by the remote key owner entity for qualified electronic signature, or
    - other applications used by the local key owner entity for qualified electronic signature and electronic sealing operations, time stamp operations, authentication services, etc.
- Standard APIs (e.g. a PKCS#11, OpenSSL API) through which end users can securely access the Trident besides the evaluated CMAPI interface.

1.4 TOE description

Depending on its configuration the Trident consists of one, two, three or four MPCAs. If the distributed functionality is provided with high-availability clustering, then Trident consists of N-times one, two, three or four MPCAs, where N is the number of nodes within a high-availability cluster. The generic architecture of an MPCA (for all models) is shown in *Figure 1.5*.

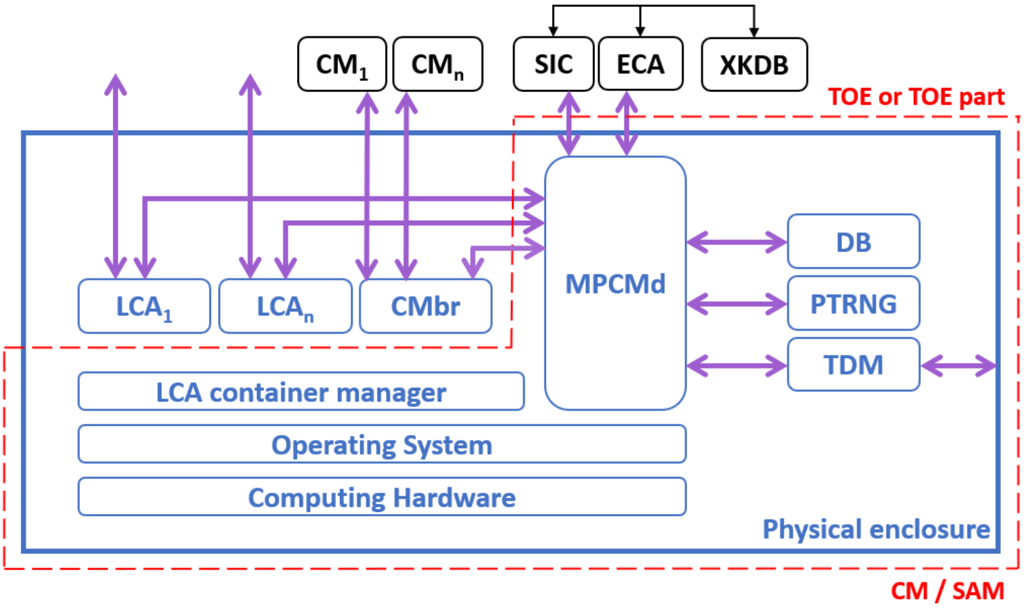

Figure 1.5 MPCA architecture

Physical enclosure: the MPCA is a metal, rack mountable box. Depending on the model:

- Model A11, A21, A31, A33: I4P v7 (see *Figure 1.6*)
- Model B11, B31, B33: I4P v8 (see *Figure 1.7*)
- Model C16: Proteccio (see *Figure 1.8*)

Computing Hardware: a hardware platform for the Operating System.

Operating System: hardened OS

LCA container manager: the service managing the Local Client Applications, which provide isolated execution environments for the LCAs

LCA: Local client applications are embedded application running inside the physical boundary of the MPCA:

- the CMbr is a non-TOE part LCA,
- other LCAs (LCA$_1$, LCA$_n$ in *Figure 1.5*) are also non-TOE parts.

LCAs can use cryptographic services/functions provided by MPCMd only through the same API which is enabled for all ECAs.

CMbr: Embedded application which transfers the PKCS#11 commands from MPCMd to an external Crypto Module (configured to be used, if there is any). External Crypto Module is not part of the TOE evaluated/certified configuration. Consequently, CMbr is not TOE part.

ECA: External client applications communicate remotely with the TOE through a network connection.

MPCMd: Multi-party Cryptographic Module daemon (also called Multi-party Cryptographic Module or MPCM) provides cryptographic services/functions for the LCAs and the ECAs. In case of the distributed configuration, the more MPCMd jointly provide the CM functionality.

SAM functionality and SAD (see *Figure 1.2*): Signature Activation Module functionality and the

Signature Activation Protocol (SAP), using the Signature Activation Data (SAD) from a remote signer to activate the corresponding signing key is also implemented by MPCMd. In case of the distributed configuration, the more MPCMd jointly provide the SAM functionality.

PTRNG: depending on the model:

- Model A33, B33: SE050 (TÜV Rheinland Nederland B.V. Certification Report JCOP 4 P71, NSCIB-CC-180212-CR4-1.0 and Certificate report - BSI-DSZ-CC-1136-2021 for NXP Smart Card Controller N7121 with IC Dedicated Software and Crypto Library (R1/R2) from NXP Semiconductors Germany GmbH)
- Model A11, A21, A31, B11, B31: IDPrime 940 (the Infineon chip SLE78CLFX400VPHM with IDPrime 940 Smart Card. This chip has a Common Criteria EAL 5 augmented by ALC_DVS.2, AVA_VAN.5, Certification: ANSSI-CC-2018/24, Maintenance of the certification: ANSSI-CC-2018/24-M01)
- Model C16: FPGA (has a Common Criteria EAL4 augmented by AVA_VAN.5, Security Target Lite: TrustWay PROTECCIO ST, v1.3, PCA4_0152, 29 July 2024, Certificate: ANSSI-CC-2024/13

  In case of this model the fulfillment of the following SFRs is confirmed by the certificate of the FPGA:
  - FCS_RNG.1 (FCS_RND.1 in the TrustWay PROTECCIO ST)
  - FPT_PHP.1 (FPT_PHP.2 in the TrustWay PROTECCIO ST)
  - FPT_PHP.3 (FPT_PHP.3 in the TrustWay PROTECCIO ST).[1]

TDM: Tamper Detection Module detects different tamper events. Depending on the model:

- Model A11, B11: TDM architecture type v1,
- Model A21: TDM architecture type v2,
- Model A31, A33, B31, B33: TDM architecture type v3.
- Model C16: FPGA

CM: External Cryptographic Module(s). Trident can communicate with these as PKCS#11 devices.

DB: Database, stores keys and sensitive information within the TOE boundaries

XKDB: optional External Key Database to store signing keys outside of the TOE boundaries. Key records are encrypted by MPCMd before sending to XKDB and the encryption key is not leaving the TOE.

1.4.1 The physical scope of the TOE

The evaluated configuration of the Trident includes the following items:

- one, two, three, four or more MPCAs, and
- one CD with the needed guides in PDF format, which provides guidance on the evaluated configuration and refers the reader to the relevant product guides to enable him to install and operate the Trident correctly:
  - Trident Administrators' Guide - CM and SAM (configuring and administering the

[1] As FPT_PHP.2 is hierarchical to FPT_PHP.1, it can be seen that the previous certification of the FPGA provides an equivalent assurance guarantee for the affected SFRs.

MPCMd) [Trident-ADMG],

- o Trident Developers' Guide - CMAPI and SAP (using the externally and internally available CMAPI and externally available SAP) [Trident-DEVG],

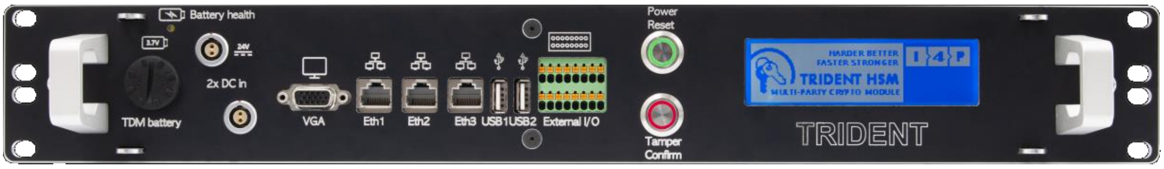

Figure 1.6 Physical appearance of an MPCA (models: A11, A21, A31, A33)[2]

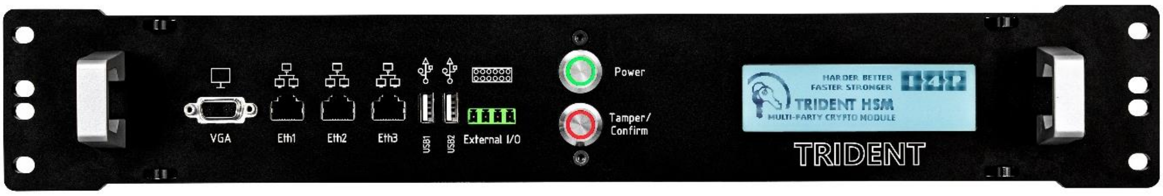

Figure 1.7 Physical appearance of an MPCA (models: B11, B31, B33)[2]

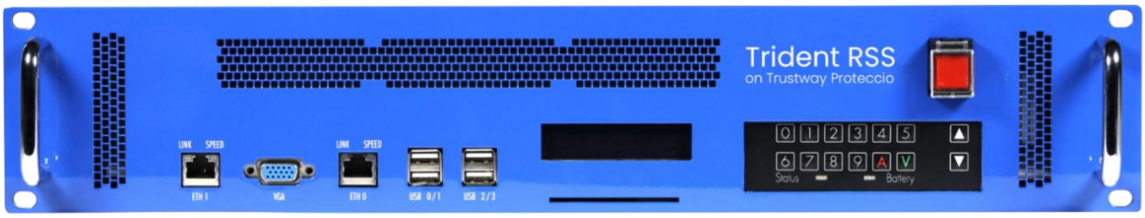

Figure 1.8 Physical appearance of an MPCA (model:C16)[2]

An MPCA is a tamper protected hardware, which itself consist of different hardware and software components in a closed and sealed, rack mountable, metal box, plus its external power supply or supplies and the needed power cables. All MPCAs include the following items:

- a metal, rack mountable box, internal or external power supply unit (see Table 1.1)
- physical interfaces and the internal hardware (see Table 1.1)
- the internal software (in all models):
  - o hardened OS,
  - o limited shell,
  - o Multi-Party Cryptographic Module (in case of distributed configuration, the n (n=2, 3, 4) MPCAs jointly provide the CM functionality),

[2] Note: as the hardware outlook is customizable on the customer's request, the color of the device (including the LCD screen) may differ from what is shown in the picture.

  - Signature Activation Module local client application (in case of distributed configuration, the n (n=2, 3, 4) SAM LCAs jointly provide the SAM functionality),
  - OpenSSL, which performs the TLS protocol and all non-distributed cryptographic functions, supports distributed cryptographic functions, and provides base functions for DRNG,
  - others LCAs (non-TOE parts).

The following table summarizes the differences between TOE models.

model	**A11**	**A21**	**A31**	**A33**	**B11**	**B31**	**B33**	**C16**
Hardware								
Hardware enclosure	I4P v7	I4P v7	I4P v7	I4P v7	I4P v8	I4P v8	I4P v8	Proteccio
Battery LED	yes	yes	yes	yes	integrated	integrated	integrated	no
Replaceable battery type	-	1 x 14500	1 x 14500	1 x 14500	-	2 x 18650	2 x 18650	-
LCD	yes	yes	yes	yes	yes	yes	yes	no
Power plug	24V LEMO	24V LEMO	24V LEMO	24V LEMO	24V DC barrel	24V DC barrel	24V DC barrel	230V C13
Cooling	passive	passive	passive	passive	passive	passive	passive	active
Height	1.5U	1.5U	1.5U	1.5U	1.5U	1.5U	1.5U	2U
Tamper detection								
Case opening	electrical circuit	TDM	TDM	TDM	electrical circuit	TDM	TDM	FPGA
Temperature monitor	OS	TDM	TDM	TDM	OS	TDM	TDM	FPGA
Voltage monitor	OS	TDM	TDM	TDM	OS	TDM	TDM	FPGA
TDM HW/FW								
TDM architecture type	1	2	3	3	1	3	3	n/a
Physical True Random Generator								
PTRNG	IDPrime 940	IDPrime 940	IDPrime 940	SE050	IDPrime 940	IDPrime 940	SE050	FPGA
Additional software								
LCD manager LCA	yes	no	yes	yes	yes	yes	yes	no

Table 1.1 The differences between TOE models

The developer uses contracted distribution service to ship the TOE to its customer. Delivery steps taken when shipping to customers:

- A TOE ("system" type stored item) with "ready" state is selected from the storage (if it is a new order fulfillment than it is a "new" or if it was serviced than it is a "used" system).
- The TOE is moved into its shipment box, sealed using security tape and labelled.
- Contracted distribution service is ordered with insurance covering the value of the TOE

- Customer is informed about the shipment information - including the serial numbers of the tamper evident seals, the serial number of the TOE, initial admin credentials, as well as the steps to be taken when the shipment arrives.
- Contracted distribution service ships the TOE to the customer.
- Customer checks the tamper evident seals on the shipment box.
- If shipment box was not physically tampered with then customer unpacks and checks the tamper evident seals and cables on the TOE.
- If the TOE was not physically tampered with then customer starts the TOE and checks the version information and the serial number shown on the screen.
- Customer checks the TOE version information and the serial number with the information he/she received earlier.
- Customer prints and fills the acceptance checklist received earlier, signs it and sends it back to I4P upon which the customer gets registered for guarantee and flaw remediation.
- If any of the tamper seals, version information and serial number control show a tamper event, customer contacts I4P for discussing further steps, which may include sending back the TOE for inspection.

1.4.2 The logical scope of the TOE

1.4.2.1 CM functionality

Roles and available functions

The CM (i.e. CM functionality of the Trident) maintains the following roles, associating users with roles:

- Administrator, a privileged subject who can perform CM specific management operations, through a local console or the externally available CMAPI, including the following:
    - Create_New_Administrator (creating a new account with security attributes for an Administrator). Creating the initial (first) Administrator requires entering an installation code.
    - Public asymmetric key export (using a PKCS#10 or a CMC ([RFC 2797]) certificate request for exporting the public asymmetric key components).
    - Unblocking (unblocking access to a blocked key)
    - Modifying attributes of keys (Key Usage),
    - Audit data export/deletion (exporting and deleting the local audit file and the ErrorLog)
    - Backup and restore functions (restore function is under dual control).
- Key User, a normal, unprivileged subject who can invoke operations on a key according to the authorisation requirements for the key. This role acts through a local client application or through an external client application.
- Local Client Application, application running inside the physical boundary of the MPCA.
- External client application, application communicating remotely with one of the MPCA through a network connection.

Authentication and Authorisation

The CM uses a common method for identification and authentication in case of each role: a unique user identifier (sent by the user during authentication) + (static password and/or TOTP and/or JWT). The static password is checked against the RAD (salted, hashed and encrypted password) stored in the user's account as a security attribute. The TOTP is checked using 256 bits long shared secret,

The CM blocks the account after a predefined number of consecutive failed authentication attempts, where these administrator configurable numbers can be different for each role.

Before using a secret key in a cryptographic operation an authorisation or a re-authorisation as a user of the key is always required. The CM blocks the secret key after a predefined number of consecutive failed authorisation attempts.

Key Management

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

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

The TOE supports the following techniques for establishing keys:

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

Keys may leave the TOE in one of three possible situations:

1. External storage of keys: The TOE may allow external storage of keys for later use by the TOE. 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 outside the TOE are wrapped with an infrastructural key 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.
2. Export of keys: The TOE may allow export of keys for use by authorized client applications, provided that they are not Assigned Keys, that other key attributes do not prohibit export, and that the correct authorization data for the key has been supplied. Although the TOE checks key attributes to determine whether to allow export, the appropriate values to use for the key attributes will depend on the application context in which the key is used, and the security measures (technical, physical and procedural) that apply to that context. Keys might be imported or exported as part of providing general cryptographic functions (e.g. in support of client applications that use the TOE to support their own authentication mechanisms) but the TOE also allows individual secret keys to be identified as non-exportable.
3. Backup: The TOE may provide facilities for secure backup and restore of the TSF state.

Key Security

The CM ensures the security of its keys for their whole lifecycle. The generic key lifecycle includes the methods by which a key may arrive in the Trident (import, generation or restore from backup), resulting in binding of a set of attributes to the key, storage of the key, and finally the ways in which a stored key may then be processed (export, use in a cryptographic function, backup, destruction).

Key export/import

The CM does not provide facilities to export or import Assigned keys.

The CM allows import and export of secret (non-Assigned) keys only in encrypted form.

Public keys may be imported and exported in a manner that protects the integrity of the data during transmission.

Key generation

The CM generates different types of keys for its supported cryptographic operations:

- RSA key pairs for end users (with key lengths of 2048, 3072, 4096, 6144 bits),
- ECC key pairs for end users (Elliptic Curves with key lengths of 224, 233, 239, 256, 272, 283, 304, 320, 359, 368, 384, 409, 431, 512, 521, 571 bits),
- SPHINCS+, NTRU, ML-KEM (Kyber) post-quantum key pairs for end users,
- infrastructural RSA (2048, 3072, 4096 bits) and ECC key pairs for internal security mechanisms,
- AES keys (256 bits) for file and record encryption/decryption,
- AES (128, 192, 256 bits), 3DES (192 bits), ARIA (128, 192, 256 bits) and SEED (128 bits) keys for end users,
- shared secrets (256 bits for TOTP and n bits for HMAC),
- master secrets (384 bits) for TLS.

The CM uses approved standards for key generation.

The security attributes of the newly generated keys have restrictive default values.

The generation of all keys (including all shares of the private keys and of the pre-generated prime numbers) based on an appropriate hybrid deterministic random number generator, whose internal state uses a physical true RNG as a random source.

Key restore from backup

The CM provides a function to restore secret keys from backup.

Only two Administrators are able to perform the restore function (dual control).

In the backups, all data (including keys, key attributes, authentication data) are signed and encrypted. Consequently, any restore operation preserves their integrity (including the binding of each set of attributes to its key) and confidentiality.

Binding of a set of attributes to the key

The CM binds the following set of attributes to the Key User’s keys, which determine their use:

Attribute	Description	Initialisation/Modification
Key ID key identifier	uniquely identifies the key within the system of which the CM is a part.	Initialised by generation process Cannot be modified
Owner ID	identifies the Key User(s) who own(s) the key or key parts.	Initialised by generation process Cannot be modified
Key Type	identifies the type of the key (e.g. AES or RSA)	Initialised by generation process Cannot be modified
Authorisation Data	Value of data that allows a secret key to be used for cryptographic operations. The CM does not store the value of the Authorisation data, but uses it for encrypt/decrypt (share of) the key.	Initialised by authenticated Key User Modified only when modification operation includes successful validation of current (pre-modification) authorisation data
Re-authorisation conditions	The constraints on uses of the key that can be made before reauthorisation, and which determine whether a subject is currently authorised to use a key.	Initialised by generation process Cannot be modified
Key Usage	The cryptographic functions that are allowed to use the key	Initialised by creator during generation Cannot be modified
Assigned Flag	Indicates whether the key has currently been assigned. For an Assigned Key, 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 ‘assigned’ and ‘non-assigned’.	Initialised by generation process Cannot be modified
Uprotected Flag	indicates whether the stored key is protected only with an infrastructural key, or additionally with a password established by the Key User (key’s owner). This flag is initialised by key generation process, setting its value to “no”. When the Key User establishes his/her Authorisation Data, the value of this flag is set to “yes”.	Initialised by generation process For an Assigned Key: modified only when the Key User establishes his/her Authorisation Data For a non-Assigned Key: modified only by Key User
Operational Flag	indicates whether the key is in operational state. This flag is initialised by key generation process to “non-operational”. A key can be used for cryptographic operations only in “operational” state. Only the Key User (key’s owner) is able to change the value of this flag from “non-operational” to “operational” and vice versa.	Initialised by generation process Can be modified only by Key User
Integrity Protection Data	is a digital signature created by an infrastructural key for key data record which contains the key and its attributes	Cannot be modified by users (maintained automatically by TSF)

Attribute	Description	Initialisation/Modification
Key Device Type	indicates whether the key is generated, stored and used by the TOE itself (default) or by an external CM[3]	Initialised by creator during generation Cannot be modified

Table 1.2 Key Attributes

Storage of the key

The CM protects the integrity of keys and their attributes:

- All stored data records (including keys with their security attributes) have a "record signature" element which is a PKCS#1 RSA signature with an infrastructural key.
- Before any use of a key a signature verification is performed for its "record signature".
- Upon detection of a data integrity error, the CM prohibits the use of the altered data and notifies the error to the user.

The CM protects the confidentiality of secret keys and their sensitive attributes:

- All stored secret keys and all sensitive key attributes are encrypted with an infrastructural key.
- The CM explicitly denies the access to the plaintext value of any secret key (neither directly nor through intermediate values in an operation).

Key export

The CM controls the key export:

- only authorized Administrators are able to perform key export,
- only non-Assigned keys are allowed to export,
- only keys with "Export Flag"="exportable" are allowed to export.

The CM protects the confidentiality of secret keys during export:

- key export requires a secure channel,
- key export is allowed only in encrypted form.

Key usage

An authorisation is required before use of a key and the key can only be used as identified in its Key Usage attribute.

In addition, the initial authorisation, a re-authorisation is required depending the re-authorisation conditions such as expiry of a time period or number of uses of a key, or after explicit rescinding of previous authorisation.

The CM protects the authorisation data: minimizes the time that authorisation data is held; stores only in RAM; zeroises before deallocation.

The CM blocks the access to a key on reaching an authorisation failure threshold. Only an administrator is able to unblock a key, but the unblocking process does not itself allow the keys to be used. Unblocking access to a key does not allow any subject other than those authorised to access the key at the time when it was blocked.

[3] Using external CM is not part of the TOE evaluated/certified configuration.

The CM supports different approved algorithms for different purposes identified in the Table 1.3.

cryptographic operations	cryptographic algorithms	cryptographic key sizes	applicable standards	supported operations
creation/ verification of digital signatures/seals	RSASSA-PKCS1-v1_5, RSASSA-PSS	2048, 3072, 4096, 6144 bits	[TS 119312], [PKCS #1], [FIPS 186-5]	local signing, remote server signing, verification
creation/ verification of digital signatures/seals	SPHINCS+ Signature Generation/ Verification	1024, 2048 bits	[SPHINCS+]	local signing, remote server signing, verification
creation/ verification of digital signatures/seals	ECDSA	224, 233, 239, 256, 272, 283, 304, 320, 359, 368, 384, 409, 431, 512, 521, 571 bits (all elliptic curves identified in Table 1.4)	[SEC 2], [X9.142], [FIPS 186-5], [RFC 5639]	local signing, remote server signing, verification
creation/ verification of digital signatures/seals	Schnorr	224, 233, 239, 256, 272, 283, 304, 320, 359, 368, 384, 409, 431, 512, 521, 571 bits (all elliptic curves identified in Table 1.4)	[FIPS 186-5] [Schnorr]	local signing, remote server signing, verification
creation/ verification of digital signatures/seals	ML-DSA	private key: ML-DSA-44: 2560 ML-DSA-65: 4032 ML-DSA-87: 4896	[FIPS 204]	local signing, remote server signing, verification
cryptographic hash function	SHA-256, SHA-384, SHA-512	none	[TS 119312], [FIPS 180-4]	TLS protocol, signing a log or a database record or a stored file, generating or checking the integrity protection data
cryptographic hash function	SHA3-256, SHA3-384, SHA3-512 SHAKE128 SHAKE256	none	[FIPS 202]	
keyed-hash message authentication	HMAC_SHA-256 HMAC_SHA-384 HMAC_SHA-512	128/192/256 bits message digest sizes: 256/384/512 bits	[RFC 2104] [FIPS 198-1] [SP800-224]	TLS protocol, PBKDF2 key derivation HKDF key derivation
cipher-based message authentication code	AES-CMAC	128, 192, 256 bits	[SP800-38B]	TLS protocol, PBKDF2 key derivation
	3DES-CMAC	192 bits	[SP800-38B]	
	ARIA-CMAC	128, 192, 256 bits	[RFC 5794]	
	SEED-CMAC	128 bits	[RFC 4493]	

cryptographic operations	cryptographic algorithms	cryptographic key sizes	applicable standards	supported operations
encryption and decryption	AES (in CBC, CCM, CFB1, CFB8, CFB, CTR, ECB, GCM, OFB, XTS mode)	128, 192, 256 bits	[FIPS 197], [SP800-38A]	data encrypting/decrypting TLS protocol, SAP protocol, writing/reading a stored file or data record
encryption and decryption	3DES (in ECB, CBC, CFB1, CFB8, CFB, OFB mode)	192 bits	[SP800-38A]	data encrypting/decrypting
encryption and decryption	ARIA (in ECB, CBC, CCM, CFB1, CFB8, CFB, OFB, CTR, GCM mode)	128, 192, 256 bits	[RFC 5794]	data encrypting/decrypting
encryption and decryption	SEED (in ECB, CBC, CFB, OFB mode)	128 bits	[RFC 4269]	data encrypting/decrypting
secure messaging - encryption and decryption	RSAES-PKCS1-v1_5	2048, 3072, 4096, 6144 bits	[PKCS#1]	TLS protocol, SAP protocol, wrapping/unwrapping the AES/3DES keys
key derivation	PBKDF2	length of password	[PKCS#5]	encrypting passwords, deriving key encryption keys
key derivation	Balloon	length of password	[Balloon]	password-hashing using as key derivation during the key import
key derivation	HKDF	length of input keying material	[RFC 5869]	key derivation during the key import
TOTP verification	HOTP	256 bits	[RFC 4226], [SP800-90A]	using for HOTP
JWT verification	ECDSA RSASSA-PKCS1-v1_5	256, 384, 521 bits (ES256. ES384, ES512) 2048, 3072, 4096, 6144 bits (RSA256, RSA384, RSA512)	[RFC 7515], [RFC 7518], [RFC 7519]	token verification
cryptographic support for one-time password (TOTP verification)	HOTP	256 bits	[RFC 4226], [RFC 6238]	possession-based authentication of the Signer
random number generation	CTR_DRBG	x bytes	[SP800-90A]	genaration of keys, IVs, session IDs, salt

cryptographic operations	cryptographic algorithms	cryptographic key sizes	applicable standards	supported operations
key exchange	ECDH	224, 233, 256, 283, 384, 409, 521, 571 bits (elliptic curves: secp224k1, secp224r1, secp256k1, secp256r1, secp384r1, secp521r1, sect233k1, sect239k1, sect283k1, sect409k1, sect571k1, sect233r1, sect283r1, sect409r1, sect571r1)	[SP800-56A]	key exchange
key encapsulation	NTRU	length of shared key: 256	[NTRU]	key exchange
key encapsulation	ML-KEM (Kyber)	length of shared key: 256	[FIPS 203]	key exchange
hybrid encryption and decryption	(RSA, AES), (RSA, 3DES) (RSA, ARIA), (RSA, SEED)	see the following rows in this table: secure messaging – encryption and decryption (RSAES-PKCS1-v1_5) and encryption and decryption (AES, 3DES, ARIA or SEED)		
hybrid encryption and decryption	(EC, PBKDF2, AES), (EC, PBKDF2, 3DES) (EC, PBKDF2, ARIA) (EC, PBKDF2, SEED)	see the following rows in this table: key exchange (ECDH), key derivation and message encryption and decryption (AES, 3DES, ARIA or SEED)		

Table 1.3 Supported cryptographic operations and algorithms

Table 1.4 identifies the supported Elliptic Curves[4]:

[SEC 2] [RFC 4492]	[SP800-56A] [FIPS 186-5]	[X9.142]	[RFC 5639]	Prime/ Binary Fields	distributed private key is supported
secp224k1				Prime	yes
secp224r1	P-224			Prime	yes
			brainpoolP224r1	Prime	yes
			brainpoolP224t1	Prime	yes
sect233k1				Binary	no
sect233r1				Binary	no
sect239k1				Binary	no
		prime239v1		Prime	yes
		prime239v2		Prime	yes
		prime239v3		Prime	yes
		c2tnb239v1		Binary	no
		c2tnb239v2		Binary	no
		c2tnb239v3		Binary	no
secp256k1				Prime	yes

[4] Cryptographic operations using brainpool elliptic curves are implemented using OpenSSL module in non-FIPS Mode.

[SEC 2] [RFC 4492]	[SP800-56A] [FIPS 186-5]	[X9.142]	[RFC 5639]	Prime/ Binary Fields	distributed private key is supported
secp256r1	P-256	prime256v1		Prime	yes
			brainpoolP256r1	Prime	yes
			brainpoolP256t1	Prime	yes
		c2pnb272w1		Binary	no
sect283k1	K-283			Binary	no
sect283r1	B-283			Binary	no
		c2pnb304w1		Binary	no
			brainpoolP320r1	Prime	yes
			brainpoolP320t1	Prime	yes
		c2tnb359v1		Binary	no
		c2pnb368w1		Binary	no
secp384r1	P-384			Prime	yes
			brainpoolP384r1	Prime	yes
			brainpoolP384t1	Prime	yes
sect409k1	K-409			Binary	no
sect409r1	B-409			Binary	no
		c2tnb431r1		Binary	no
			brainpoolP512r1	Prime	yes
			brainpoolP512t1	Prime	yes
secp521r1	P-521			Prime	yes
sect571k1	K-571			Binary	no
sect571r1	B-571			Binary	no
	ED25519			Prime	no
	ED448			Prime	no

Table 1.4 Supported Elliptic Curves

Key backup

The CM provides a function to backup the TOE, thus the stored secret keys.

Only Administrators are able to perform the backup function. All backups are signed, Consequently, any backup preserves their integrity (including the binding of each set of attributes to its key). All backups are encrypted. Consequently, any backup preserves their confidentiality.

Key destruction

All secret keys and all authorization data are zeroised (with physically overwriting) at the end of their lifecycle or after they have been deallocated.

TSF data protection

The CM ensures the security of its TSF data, implementing self-tests, and providing secure failure and tamper protection capability.

Self-tests

The CM provides a suite of self-tests, which check and demonstrate the correct operation of the CM security functionality. The CM implements these self-tests:

- during initial start-up (including software/firmware integrity test, cryptographic algorithm tests and random number generator tests),
- periodically during normal operation (e.g. checking the environmental resources, checking whether the environmental conditions (including temperature and power) are outside normal operating

range),

- at the request of the Administrator (software/firmware integrity tests),
- at the conditions (e.g. pair-wise consistency tests during the asymmetric key pair generation)

Each MPCA performs the same self-tests, but at different times.

Secure failure

In case of critical failures, the CM enters a secure error state, in which it no more services its end users, but only performs infrastructural services. These critical errors include but are not limited to the following: self-test fails, environmental conditions are outside normal operating range, failures of critical TOE hardware components (including the RNG) occur.

Tamper protection

The CM implements a tamper detection security function:

- The MPCAs are protected by using uniquely identifiable tamper-evident seals and an appropriate physical design that allows the Administrator to verify the physical integrity of the MPCAs as part of a routine inspection procedure.
- This requires regular visual inspection of the MPCAs for signs of tamper at a frequency determined by the risk assessment of the specific operational environment.

The CM has a tamper resisting architecture:

- All shares of the secret keys and all sensitive key attributes stored permanently in the CM are encrypted with an infrastructural key.
- Authorisation data are not stored permanently in the TOE.

The CM implements a tamper response security mechanism:

- Tamper response is based on active protection of the MPCA. It is a combination of tamper sensors, temperature and voltage monitor.
- If any MPCA detects a physical tampering (eg. removing the cover of the closed physical enclosure) the CM enters a Tamper state.
- A result of the entering the Tamper state:
    - all processing of end users' requests is halted,
    - all authentication and authorization data, all key shares and all sensitive key attributes stored temporarily in RAM are immediately zeroized with physically overwriting,
    - the internal state of the DRNG is zeroized with the uninstantiate function.
- If the CM is in Tamper state, the CM does not perform any cryptographic operation and does not respond to any user request.

Audit

The CM audits all security related events. The audit records do not include any data which allow to retrieve sensitive data.

Every audit record includes the time of the event, subject identity (if applicable) and a human readable descriptive string about the related event. The CM detects unauthorised modification (including deletion and insertion) to the stored audit records in the audit trail.

Every block of audit record includes a serial number, a reliable time stamp (date and time of the event), an identifier of the related MPCA, and are signed with an infrastructural key.

The CM automatically transfers the blocks of audit records to an external audit server. If the transfer of an audit block has failed, the CM temporarily accumulates audit blocks locally in an audit directory, and periodically retries the transfer to the external audit server.

If the audit sub-system doesn't work for a reason, a special file (ErrorLog) is created and the audit records are appended to it while the system shuts down.

When local audit storage exhaustion is detected, the CM requires the local audit file to be successfully exported and deleted before allowing any other security related actions.

Only the Administrator is able to export and delete the local audit file and the ErrorLog**.**

Trusted communication

The CM implements and enforces:

- a secure channel based on TLS protocol, for communication with Administrators (through the SSA) and ECAs,
- a secure channel based on SSH protocol, for communication with Administrators (using the console command interface in the provided limited shell),
- a direct channel for communication with Administrators (using the console command interface with a physical keyboard),

The internal communication among different CM parts (among MPCAs) is also protected by TLS protocol.

MPCM is located within the physical boundary of the same hardware appliance then the communication between them is a trusted communication (the trusted path may be mapped to the physical configuration).

Optional using of external CMs

The CM functionality of the Trident allows to use external Cryptographic Modules (based on a configuration parameter). This configuration parameter can be set during production and cannot be modified. Using external Cryptographic Modules is not part of the TOE evaluated/certified configuration.

If a key initialised by creator during generation other than 'default', the CM functionality does not perform cryptographic operations, but invokes the external CM with appropriate parameters whenever a cryptographic operation is required.

This invocation is performed through a Local Client Applications (CMbr on *Figure 1.4*) using Standard PKCS#11 API.

1.4.2.2 SAM functionality

Roles and available functions

The SAM (i.e. SAM functionality of the Trident) maintains the following roles:

- Privileged User, who can perform SAM specific operations, through a local console or the externally available CMAPI, including the following:
    - Create_New_Signer (creating a new account with security attributes for a Signer),

    - Signer_Maintenance (e.g. deleting a Key_Id from the Signer's account),
    - Create_New_Privileged_User (creating a new account with security attributes for a Privileged User). Creating the initial (first) Privileged User requires entering an installation code,
    - SAM_Maintenance (creating and modifying the SAM configuration data record and SAM configuration file),
    - Backup and Restore functions (Restore function is under dual control),
    - Signer Key Pair Generation (have the CM generate a new asymmetric key pair and assigning it to a Signer's account).
- Signer, who communicates remotely with the SAM (invoking different SAP commands), and is able to perform the following operations:
    - Signer Key Pair Generation Request (requesting a new signing asymmetric key pair generation and assigning it to his/her account),
    - ChKeyPwd (establishing or modifies the key Authorisation Data for his/her key),
    - Signing (utilizing his/her signing key in the CM, transmitting the required data, including the unique user ID, two different authentication factors, the key ID, the key Authorisation Data and one or more DTBS/R),
    - Signer_Maintenance (deleting a Key_Id from his/her account and querying the security attributes of his/her account).

Authentication

For the Privileged Users, the SAM uses the same identification and authentication method as the CM: a unique user identifier and a static password and/or a TOTP or a JWT. For the Signers, the SAM requires both authentication factors: a password (knowledge-based factor) and a TOTP (possession-based factor) or a JWT.

The authentication may be carried out by a delegated party.

Cryptographic Support

The SAM does not perform cryptographic operations for its users: especially it does not generate/store/destruct, export/import, backup/restore, or use user key.

The SAM invokes the internal CM with appropriate parameters whenever a cryptographic operation for the Signer is required.

The SAM uses different infrastructural keys to protect its stored files and database records, and data transmitted or received via communication channels.

Audit

The SAM audits all security related events. The audit records do not include any data which allow to retrieve sensitive data.

The SAM’s audit functionality is the same as the CM’s.

Trusted communication

The SAM implements and enforces:

- a secure channel based on TLS protocol, for communication with Privileged Users (through the SSA),
- a secure channel based on SSH protocol, for communication with Privileged Users (using the console command interface in the provided limited shell),
- a secure channel based on the proprietary SAP protocol,
- a direct channel for communication with Privileged Users (using the console command interface with a physical keyboard).

The internal communication among different SAM parts (among MPCAs) is also protected by TLS protocol.

The communication between SAM and Signer based on a proprietary Signature Activation Protocol. The SAP is protected against replay, bypass and forgery attack, using a nonce, a time stamp and a shared secret. The SAP provides confidentiality and integrity protection for all transmitted data, including the authentication and authorization data and DTBS/R(s).

Using the SAM functionality is optional: the SAM functionality of the Trident can also be performed by an External Client Application, using CM APIs (see *Figure 1.1*).

1.4.2.3 Distributed functionality

In case of distributed configuration, the Trident consists of n (n=2, 3 or 4) separate TOE parts (MPCAs) to operate as a logical whole in order to fulfill the requirements of this Security Target. This security function based on the distributed structure of the Trident ensures the following:

- Distributed cryptography,
- Secret sharing,
- Consistency protection,
- Fault tolerance.

A TOE in standalone configuration can be extended to distributed configuration by adding and configuring one more MPCAs to the standalone one. A distributed configuration can also be extended by adding more MPCAs, until the maximum of 4 MPCAs is reached. Although unlimited MPCAs can be configured to work together, configuration of more than 4 MPCAs were not included in the TOE Evaluation.

Distributed cryptography

Generation of the RSA key pairs (and the pre-generated primes for them) and ECC key pairs for Key Users is not performed in a single MPCA, but in a distributed way. The n (n=2, 3 or 4) MPCAs jointly generate the RSA and ECC key pairs so that at the end of the generation:

- the public key part is publicly known, but
- none of the MPCAs holds the whole private key part, only a share of it.

Similarly, the n (n=2, 3 or 4) MPCAs jointly create the digital signatures/seals (or in case of RSA: decrypt the encrypted messages), using a multi-step signing/decrypting method. Each MPCA computes a partial cryptographic operation with own private key share so that at the end of the operation:

- the result is a standard digital signature/seal (or in case of RSA a decrypted message),

- after signature creation (or in case of RSA message decryption) the shares of the private key remain secret, none of the MPCAs revealed its private shares to the other MPCAs.

The end user's cryptographic keys can be generated in a distributed or in a non-distributed way.

The distributed key generation is implemented both ways, with and without a trusted dealer.

In case of RSA, distributed multi-prime key generation is also supported.

The Key Users can interact with any MPCA (permitted by the configuration of the IT environment, e.g., firewall rules) through the externally available APIs. The distributed operation of the Trident and internal communication among the MPCAs (in order to synchronize their databases) takes place behind the scenes.

Secret sharing

Based on distributed RSA and ECC key pairs generation and distributed cryptographic operation, the Trident achieves a new guarantee for ensuring the sole control of Key User’s private keys: a single MPCA never stores the whole private key.

Authentication of the end users is also performed in a distributed way, the n (n=2, 3 or 4) MPCAs jointly authenticate the end users. The n (n=2, 3 or 4) MPCAs store shared values for password and TOTP secrets.

Consistency protection

The Trident ensures that TSF data are consistent when they are replicated between TOE parts (MPCAs). When MPCAs are disconnected, the Trident ensures the consistency of the replicated TSF data upon reconnection before processing requests for any secure relevant management or user function. This security function is based on the nested transactions capability of the used database engine (LMDB).

Fault tolerance

In case of distributed configuration, the Trident ensures a fault tolerance capability: if some of the MPCAs becomes dysfunctional (a result of a fatal error or a network unavailability) the other MPCAs (if there are any) can ensure a limited functionality.

The available functions in this case are:

- the following distributed cryptographic services:
  - RSA signature/seal creation,
  - RSA decryption,
  - ECDSA signature/seal creation,
- the following non-distributed cryptographic services:
  - (RSA, ECDSA, Schnorr, SPHINCS+) signature/seal creation,
  - (RSA, ECDSA, Schnorr, SPHINCS+) signature/seal verification,
  - Random number generation,
  - RSA encryption/decryption,
  - AES and 3DES encryption/decryption,
  - Hybrid (RSA, AES), (RSA, 3DES), (EC, AES) and (EC, 3DES) encryption/decryption,

- Cryptographic hash function,
- Keyed-hash,
- Key derivation,
- TOTP verification,
- Cipher-based message authentication code operation,
- ECDH key exchange,
- Identification and authentication,
- Audit record protection.

1.4.2.4 High Availability functionality

In case of High Availability configuration, each primary (active) MPCA has a/more fully redundant secondary (passive) MPCA couple(s). One of the secondary MPCAs is only brought online when its associated primary node fails. This security function ensures the following:

- fault tolerance.

1.4.2.5 States and lifecycle stages of the Trident

Figure 1.9 illustrates the different states of an MPCA: Delivered (D), Operational-power_on (O_on), Operational-power_off (O_off), Error (E) and Tampered (T).

The supplier (developer/manufacturer) delivers the Trident (i. e. the one, two, three or the four MPCAs) to the customer in **Delivered state.** In this state, all software and hardware components of the MPCA(s) are installed, pre-configured and initialized. The physical enclosure is closed, and all MPCAs assure active tamper detecting and tamper resistance functionalities. In this state users cannot perform any functions of the Trident described in 1.3.3 and 1.4.2.

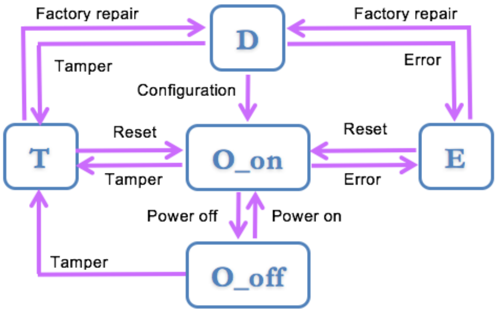

Figure 1.9 Diagram of the different states and state transitions of an MPCA

Powering off an MPCA triggers the transition from **Operational-power_on** state to **Operational-power_off state**, just like powering on launches the transition from Operational-power_off state to Operational-power_on state.

Detecting a fatal error (according to FPT_FLS.1) triggers the transition from Operational-power_on states to **Error state.** The Error state indicates an appliance malfunction that requires a security log analysis (to determine the reason of the error) and then resetting or repairing of the MPCA.

Detecting a tampering triggers the transition from Operational-power_off and Operational-power_on states to **Tampered state.** The Tamper state indicates the detection of a physical tampering that requires a deep and wide investigation (including security log analysis) to determine whether an error or a tampering has occurred. Depending on the conclusions, the result could be a resetting, a restoring or a repairing.

In Error and Tampered states users cannot perform any functions of the Trident, except that the Administrator can try to export the local audit and Errorlog file.

If the only MPCA is in Operational-power_on state, users can activate all functions of the Trident.

If all MPCAs are in Operational-power_on state, users can activate all functions of the Trident.

If less than all, but minimum 2 MPCAs are in Operational-power_on state, users can activate the limited functionality of the Trident, which contains almost all functions, except management and key generation functions (see “Fault tolerance” above).

In case of only one MPCA is in Operational-power_on state, only the non-distributed end user services function.

1.4.3 Features and Functions not included in the TOE Evaluation

The Trident is capable of a variety of functions and configurations which are not covered by the PPs that this ST claims conformance to, such as:

- building up the distributed system from more than four number of identical MPCAs (n= 5, 6, …),
- active-passive node clustering
- features and functions of an LCA other than the SAM,
- general cryptographic functions available through the SAP protocol,
- distributed authentication,
- using external Cryptographic Modules.

2 Conformance claims

2.1 CC conformance claim

This Security Target claims to be Common Criteria Part 2 extended and Common Criteria Part 3 conformant and written according to the Common Criteria version 3.1 R5 [CC1], [CC2] and [CC3].

2.2 PP claim

This Security Target conforms to

- Protection Profile [EN 419221-5] (PP for Trust Service Provider Cryptographic Modules - Part 5) and
- Protection Profile [EN 419241-2] (PP for QSCD for Server Signing).

Both PPs require strict conformance.

2.3 Package claim

This ST conforms to assurance package EAL4 augmented by AVA_VAN.5 and ALC_FLR.3 defined in [CC3].

2.4 Conformance rationale

This ST claims strict conformance to Protection Profiles [EN 419221-5] and [EN 419241-2].

[EN 419221-5] defines the security requirements for cryptographic modules which is intended to be suitable for use by trust service providers supporting electronic signature and electronic sealing operations, certificate issuance and revocation, time stamp operations, and authentication services, as identified in [eIDAS].

[EN 419241-2] defines the security requirements to reach compliance with

Annex II of [eIDAS] assuming use of a cryptographic module conforming to [EN 419221-5]. Consequently, being conformant to [EN 419221-5] and [EN 419241-2] at the same time guarantees the compliance with Annex II of [eIDAS] (REQUIREMENTS FOR QUALIFIED ELECTRONIC SIGNATURE CREATION DEVICES).

PPs [EN 419221-5] and [EN 419241-2] require strict conformance of the ST claiming conformance to these PPs.

The TOE (Trident) type covers the TOE types of the PPs [EN 419221-5] and [EN 419241-2]:

- The SAM module is a software component, which implements the Signature Activation Protocol (SAP).
- The SAM module deployed in a Cryptographic Module (CM).
- Together the SAM and CM are a QSCD.

To demonstrate that strict conformance is met, this rationale shows followings (see: [CC1], 287):

(1) The ST shall contain all threats of the PPs and may specify additional threats.

The Table 2.1 demonstrates that this ST contains all threats of the PPs [EN 419221-5] and [EN 419241-2], and specifies additional threats.

Threat	**This ST**	[EN 419 221-5]	[EN 419 241-2]
T.KeyDisclose	+	+	-
T.KeyDerive	+	+	-
T.KeyMod	+	+	-
T.KeyMisuse	+	+	-
T.KeyOveruse	+	+	-
T.DataDisclose	+	+	-
T.DataMod	+	+	-
T.Malfunction	+	+	-
T.ENROLMENT_SIGNER_IMPERSONATION	+	-	+
T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED	+	-	+
T.SVD_FORGERY	+	-	+
T.ADMIN_IMPERSONATION	+	-	+
T.MAINTENANCE_AUTHENTICATION_DISCLOSE	+	-	+
T.AUTHENTICATION_SIGNER_IMPERSONATION	+	-	+
T.SIGNER_AUTHENTICATION_DATA_MODIFIED	+	-	+
T.SAP_BYPASS	+	-	+
T.SAP_REPLAY	+	-	+
T.SAD_FORGERY	+	-	+
T.SIGNATURE_REQUEST_DISCLOSURE	+	-	+
T.DTBSR_FORGERY	+	-	+
T.SIGNATURE_FORGERY	+	-	+
T.PRIVILEGED_USER_INSERTION	+	-	+
T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION	+	-	+
T.AUTHORISATION_DATA_UPDATE	+	-	+
T. AUTHORISATION_DATA _DISCLOSE	+	-	+
T.CONTEXT_ALTERATION	+	-	+

T.AUDIT_ALTERATION	+	-	+
T.RANDOM	+	-	+
T.Inconsistency	+	-	-
T.Intercept	+	-	-
T.Breakdown	+	-	-
T.Update_Compromise	+	-	-

Table 2.1 Threats

(2) The ST shall contain all OSPs of the PPs and may specify additional OSPs.

The Table 2.2 demonstrates that the OSPs in this ST are a superset to the OSPs in the PPs to which conformance is claimed.

Organizational Security Policy	**This ST**	[EN 419221-5]	[EN 419241-2]
P.Algorithms	+	+	-
P.KeyControl	+	+	-
P.RNG	+	+	-
P.Audit	+	+	-
P.RANDOM	+	+	+[5]
P.CRYPTO	+	-	+[6]
P.BACKUP	+	-	-

Table 2.2 Organizational Security Policies

(3) The ST shall contain all assumptions as defined in the PPs, with two possible exceptions:

- an assumption (or a part of an assumption) specified in the PP may be omitted from the ST, if all security objectives for the operational environment defined in the PP addressing this assumption (or this part of an assumption) are replaced by security objectives for the TOE in the ST;
- a new assumption may be added in the ST to the set of assumptions defined in the PP, if this new assumption does not mitigate a threat (or part of a threat) meant to be addressed by security objectives for the TOE in the PP and if this assumption doesn't fulfil an OSP (or a part of an OSP) meant to be addressed by security objectives for the TOE in the PP;

Table 2.3 demonstrates that the assumptions in this ST are identical to the assumptions in the PPs to which conformance is claimed.

[5] This Organizational Security Policy is covered by P.RNG (OSP for CM)

[6] P.CRYPTO is an OSP from [EN 419241-2]. Since the SAM is implemented as a local application within the same physical boundary as the CM defined in [EN 419221-5] then objective OT.Algorithm enforces the P.CRYPTO (instead of the objective for the operational environment OE.CRYPTOMODULE_CERTIFIED).

Assumption	**This ST**	[EN 419221-5]	[EN 419241-2]
A.ExternalData	+	+	-
A.Env	+	+	-
A.DataContext	+	+	-
A.UAuth	+	+	-
A.AuditSupport	+	+	-
A.AppSupport	+	+	-
A.PRIVILEGED_USER	+	-	+
A.SIGNER_ENROLMENT	+	-	+
A.SIGNER_AUTHENTICATION_DATA_PROTECTION	+	-	+
A.SIGNER_DEVICE	+	-	+
A.CA	+	-	+
A.ACCESS_PROTECTED	+	-	+
A.SEC_REQ	+	-	+

Table 2.3. Assumptions

(4) The ST shall contain all security objectives for the TOE of the PPs but may specify additional security objectives for the TOE.

Table 2.4 demonstrates that this ST contains all security objectives for the TOE of the PPs [EN 419221-5] and [EN 419241-2], and specifies five additional security objectives for the TOE.

Security objectives for the TOE	**This ST**	[EN 419 221-5]	[EN 419 241-2]
OT.PlainKeyConf	+	+	-
OT.Algorithms	+	+	-
OT.KeyIntegrity	+	+	-
OT.Auth	+	+	-
OT.KeyUseConstraint	+	+	-
OT.KeyUseScope	+	+	-
OT.DataConf	+	+	-
OT.DataMod	+	+	-
OT.ImportExport	+	+	-
OT.Backup	+	+	-

Security objectives for the TOE	This ST	[EN 419 221-5]	[EN 419 241-2]
OT.RNG	+	+	-
OT.TamperDetect	+	+	-
OT.FailureDetect	+	+	-
OT.Audit	+	+	-
OT.SIGNER_PROTECTION	+	-	+
OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	+	-	+
OT.SIGNER_KEY_PAIR_GENERATION	+	-	+
OT.SVD	+	-	+
OT.PRIVILEGED_USER_MANAGEMENT	+	-	+
OT.PRIVILEGED_USER_AUTHENTICATION	+	-	+
OT.PRIVILEGED_USER _PROTECTION	+	-	+
OT.SIGNER_MANAGEMENT	+	-	+
OT.SAD_VERIFICATION	+	-	+
OT.SAP	+	-	+
OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION	+	-	+
OT.DTBSR_INTEGRITY	+	-	+
OT.SIGNATURE_INTEGRITY	+	-	+
OT.RANDOM	+	-	+[7]
OT.SYSTEM_PROTECTION	+	-	+
OT.AUDIT_PROTECTION	+	-	+
OT.SAM_Backup	+	-	-
OT.TSF_Consistency	+	-	-
OT.PROT_Comm	+	-	-
OT.Availability	+	-	-
OT.Updates	+	-	-

Table 2.4 Security objectives for the TOE

(5) The ST shall contain all security objectives for the operational environment as defined in the PP

[7] This security objective is covered by OT.RNG (security objective for CM).

with two exceptions:

- may specify that certain objectives for the operational environment in the PP are security objectives for the TOE in the ST. This is called re-assigning a security objective. If a security objective is re-assigned to the TOE, the security objectives rationale has to make clear which assumption or part of the assumption may not be necessary anymore;
- may specify additional objectives for the operational environment, if these new objectives do not mitigate a threat (or part of a threat) meant to be addressed by security objectives of the TOE in the PP and if these new objectives do not fulfil an OSP (or a part of an OSP) meant to be addressed by security objectives of the TOE in the PP.

Table 2.5 shows that the security objectives for the operational environment in this ST include all security objectives for the operational environment of the PPs [EN 419221-5] and [EN 419241-2].

Security objectives for the operational environment	**This ST**	[EN 419 221-5]	[EN 419 241-2]
OE.ExternalData	+	+	-
OE.Env	+	+	+
OE.DataContext	+	+	-
OE.Uauth	+	+	-
OE.AuditSupport	+	+	-
OE.AppSupport	+	+	-
OE.SVD_AUTHENTICITY	+	-	+
OE.CA_REQUEST_CERTIFICATE	+	-	+
OE.CERTIFICATE_VERFICATION	+	-	+
OE.SIGNER_AUTHENTICATION_DATA	+	-	+
OE.DELEGATED_AUTHENTICATION	+	-	+
OE.DEVICE	+	-	+
OE.CRYPTOMODULE_CERTIFIED	+	-	+[8]
OE.TW4S_CONFORMANT	+	-	+

Table 2.5 Security objectives for the operational environment

(6) The ST shall contain all security functional requirements (SFRs) and security assurance requirements (SARs) in the PP, but may claim additional or hierarchically stronger SFRs and SARs.

The SFRs specified in this ST include:

[8] OE.CRYPTOMODULE_CERTIFIED requirement for the SAM is accomplished because this ST claims to be strictly conformant also to the PP [EN 419 221-5]. (see Application Note 36)

- all SFRs specified in [EN 419221-5],
- all SFRs specified in [EN 419241-2], except for the following SFRs:
    - FCS_RNG.1. (Since the SAM is implemented as a local application within the same physical boundary as the CM, and CM includes FCS_RNG.1, according to the Application Note 39 in [EN 419241-2]) it is acceptable).
    - FPT_PHP.1 and FPT_PHP.3 (The SAM is implemented as a local application within the same physical boundary as the CM, and the CM already provides a tamper-resistant environment. According to the Application Note 69 in [EN 419241-2]) it is acceptable.)

Additional SFRs of this ST ensure:

- a separate backup and restore functions for SAM local client application (FDP_ACC.1/SAM Backup, FDP_ACF.1/SAM Backup)
- trusted path (a secure channel based on SSH protocol), for communication with Administrators, using the console command interface (FTP_TRP.1/Admin),
- mutual trusted acknowledgement between separate TOE parts (FPT_SSP.2),
- the consistency of TSF data replicated between separate TOE parts (FPT_TRC.1),
- the protection of communication channels between separate TOE parts (FPT_ITT.1),
- in case of distributed configuration a degraded fault tolerance capability if one of the MPCAs becomes dysfunctional (FRU_FLT.1),
- in case of High Availability configuration a complete fault tolerance capability if the active (primary) MPCA node fails (FRU_FLT.2),
- ML-KEM (Kyber-based) key generation and key-encapsulation mechanism (FCS_CKM.2),
- remote and local trusted update (FPT_TUD_EXT.1, FTP_TRP.1/Trusted Update, FMT_MOF.1 /ManualUpdate).

Additional SFR iterations of this ST are consequences of [EN 419221-5] PP's expectations (see [EN 419221-5] Application Notes 12 and 14):

- FCS_CKM.1/ key_gen_Non-multiparty
- FCS_CKM.1/ key_gen_Multiparty
- FCS_CKM.1/invoke_CM:key gen_Non-multiparty
- FCS_CKM.1/invoke_CM: key gen_Multiparty
- FCS_COP.1/ digsig_Non-multiparty
- FCS_COP.1/invoke_CM: dig sig_Non-multiparty
- FCS_COP.1/digsig_Multiparty
- FCS_COP.1/invoke_CM:dig sig_Multiparty
- FCS_COP.1/enc-dec_Non-multiparty

- FCS_COP.1/dec_Multiparty
- FCS_COP.1/hash
- FCS_COP.1/TOTP_verification
- FCS_COP.1/cmac operation
- FCS_CKM.2/Non-multiparty
- FCS_CKM.2/Multiparty

Additional SFR iterations of this ST are consequence of [EN 419241-2] PP’s expectations (see [EN 419221-5] Application Notes 18 and 19):

- FIA_AFL.1/CM_authentication and FIA_AFL.1/CM_authorisation instead of FIA_AFL.1
- FIA_UAU.6.1/AKeyAuth and FIA_UAU.6.1/GenKeyAuth instead of FIA_UAU.6.1/KeyAuth

Several SFRs are in both PPs (e.g. FAU_GEN.1, FAU_GEN.2, FIA_UAU.1). This ST distinguishes these SFRs using */CM and */SAM (e.g.: FAU_GEN.1/CM and FAU_GEN.1/SAM)

The SARs specified in this ST include all SARs of [EN 419221-5] and [EN 419241-2]:

- EAL4 augmented by AVA_VAN.5.

Additional SAR of this ST is:

- ALC_FLR.3

Therefore, this ST shows strict conformance to [EN 419221-5] and [EN 419241-2].

3 Security Problem Definition

3.1 General

CC defines assets as entities that the owner of the TOE presumably places value upon. The term “asset” is used to describe the threats in the TOE operational environment.

3.1.1 Assets of the Cryptographic Module (CM)

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

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

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

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

3.1.2 Assets of the Signature Activation Module (SAM)

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

Application Note 1 (Application Note 1 from EN 419241-2: Applied)

The integrity and confidentiality of the signing key value is the responsibility of the CM, and the SAM shall ensure that only the signer can use the signing key under his sole control.

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

Application Note 2

In the case of the Trident the SAM derives the R.Authorisation_Data from the SAD, and handes over to the CM without holding it.

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

The SAM uses the CM for signing key pair generation. As part of the signing key pair generation, CM provides the SAM with R.Signing_Key_Id and R.SVD. The SAM provides the R.SVD to the

SSA for further handling for the key pair to be certified.

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

Application Note 3

The confidentiality of the R.DTBS/R is not required by [eIDAS], but the Trident supports this.

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

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

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

Application Note 4

In case of the Trident the SAD is a combination of two signer's authentication factors, a unique key identifier, a given R.DTBS/R or a set of DTBS/Rs and the key's authorisation data.

The authentication factors and the authorisation data shall be protected in confidentiality.

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

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

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

Application Note 5 (Application Note 8 from EN 419241-2: Applied)

The R.Signer includes references to zero, one or several R.Signing_Key_Ids and R.SVD.

Application Note 6

In case of the Trident the R.Signer does not require encrypted data then the confidentiality requirement is considered fulfilled.

R.Reference_Signer_Authentication_Data: is the set of data used by SAM to authenticate the signer. It contains all the data (e.g. TOTP device serial number, phone numbers, protocol settings etc.) and keys (e.g. device keys, verification keys etc.) used by the SAM to authenticate the signer.

This may include an SVD or certificate to verify an assertion provided as a result of delegated authentication. The R.Reference_Signer_Authentication_Data shall be protected in integrity and confidentiality.

Application Note 7

In the Trident the Reference_Signer_Authentication_Data contains (among other data):

- two signer's authentication factors (a password and a shared secret) /if the signer authentication is carried out directly by the SAM/ or
- a JsonWebToken (JWT) issued by a delegated party (as an assertion that the signer has been authenticated) /if the signer authentication is carried out indirectly or partly indirectly by the SAM/.

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

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

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

Application Note 8

In the Trident the Reference_Signer_Authentication_Data contains (among other data) two Privileged User's authentication factors (a password and a shared secret).

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

3.1.3 Additional assets

There is one additional asset in relation to the distributed structure of the TOE:

R.MPCA_Id: The Trident consists of n (n=2, 3 or 4) identical parts (Multi-Party Cryptographic Appliance or MPCA). The R.MPCA_Id is the identifier of the MPCA. The binding of the R.MPCA_Id with MPCA shall be protected in integrity.

3.1.4 Subjects of the Cryptographic Module (CM)

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

Application Note 9

The Trident supports two types of client applications:

- the local client applications that communicates locally with the CM, (i.e. within the same hardware appliance)
- the external client applications that communicate remotely with the CM over a secure channel

S.User: an end user of the CM who can be associated with secret keys and authentication /authorisation data held by the CM. An end user communicates with the CM by using a client application (S.Application).

S.Admin: an administrator of the CM. Administrators are responsible for performing the CM initialisation, TOE configuration and other TOE administrative functions.

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

3.1.5 Subjects of the Signature Activation Module (SAM)

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

Privileged User: which performs the administrative functions of the SAM.

Application Note 10 (Application Note 14, 15 and 16 from EN 419241-2: Applied)

(14) The list of subjects described in [EN 419241-1] clause 6.2.1.2 SRG M.1.2 contains more roles as it covers the whole T4WS. This ST does not define more roles.

(15) The SSA plays a special role as it interacts directly with the TOE. Privileged Users can interact with the TOE directly or via the SSA. In case of the Trident Privileged Users can interact with the SAM directly (using USB interfaces for local console administration) and via the SSA (using network interfaces).

(16) The creation of signers, management of reference signer authentication data and signing key generation is expected to be carried out together with a registration authority (RA) providing a registration service using the SSA, as specified in e.g. [ETSI EN 319411-1].

3.1.6 Threat agents of the TOE

Threat agents: The attacker 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 other defined subjects (the defined subjects in section 3.1.4 are according to the CM and in this case the attacker will not have access to the authentication or authorisation data for the subject).

3.2 Threats

3.2.1 Threats for the Cryptographic Module (CM)

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 CM.

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 CM.

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 CM, including export and backups if supported. This includes replacement of a key as well as making changes to the value of a key, or changing its attributes such as required authorisation, usage constraints or identifier (changing the identifier to the identifier used for another key would allow unauthorised substitution of the original key with a key known to the attacker). The threat therefore includes the case where an attacker is able to break the binding between a key and its critical attributes[9].

T.KeyMisuse *Misuse of a key*

An attacker uses the CM to make unauthorised use of a secret key (R.SecretKey) that is managed by the CM (including the unauthorised use of a secret key for a cryptographic function that is not permitted for that key[10]), without necessarily obtaining access to the value of the key.

T.KeyOveruse *Overuse of a key*

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

T.DataDisclose *Disclosure of sensitive client application data*

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

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 CM or during transmission between physically separate parts of the CM, so that the result returned by the CM (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 CM may develop a fault that causes some other security property to be weakened or to fail. This may affect any of the assets and could result in any of the other threats being realised. Particular causes of faults to be considered are:

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

3.2.2 Threats for the Signature Activation Module (SAM)

3.2.2.1 Enrolment

The threats during enrolment are:

T.ENROLMENT_SIGNER_IMPERSONATION

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

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

The assets R.Signer and R.Reference_Signer_Authentication_Data are threatened.

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

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

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

T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED

(abbreviated as T.ENR_SIG_AUTH_DATA_DISCL)

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

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

The asset R.Reference_Signer_Authentication_Data is threatened. Such data disclosure may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

T.SVD_FORGERY

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

The asset R.SVD is threatened.

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

Application Note 11 (Application Note 17 from EN 419241-2: Applied)

There should be a secure transport of R.SVD from SAM to RA or CA. The SAM is expected to produce a CSR (Certification Signing Request). If the registration services of the TSP issuing the certificate requires a “proof of possession or control of the private key” associated with the SVD, as specified in [EN 319 411-1] clause 6.3.1 a), this threat can be countered without any specific measures within the TOE.

3.2.2.2 Signer Management

T.ADMIN_IMPERSONATION

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

T.MAINTENANCE_AUTHENTICATION_DISCLOSE

(abbreviated as T.MAINT_AUTH_DISCL)

Attacker discloses or changes (e. g. to a known value) R.Reference_Signer_Authentication_Data during update and is able to create a signature. The assets R.Reference_Signer_Authentication_Data and R.Signing_Key_Id are threatened. Such data disclosure may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

3.2.2.3 Usage

This section describes threats for signature operation including authentication.

T.AUTHENTICATION_SIGNER_IMPERSONATION

(abbreviated as T.AUTH_SIG_IMPERS)

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

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

T.SIGNER_AUTHENTICATION_DATA_MODIFIED

(abbreviated as T.SIG_AUTH_DATA_MOD)

An attacker is able to modify R.Reference_Signer_Authentication_Data inside the SAM or during maintenance.

The asset R.Reference_Signer_Authentification_Data is threatened. Such data modification may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

T.SAP_BYPASS

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

T.SAP_REPLAY

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

T.SAD_FORGERY

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

T.SIGNATURE_REQUEST_DISCLOSURE

(abbreviated as T.SIGN_REQ_DISCL)

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

T.DTBSR_FORGERY

An attacker modifies R.DTBS/R during transfer to SAM and is able to create a signature on this modified R.DTBS/R without the signer having authorised the operation on this R.DTBS/R. The asset R.DTBS/R is threatened.

T.SIGNATURE_FORGERY

An attacker modifies R.Signature during or after creation or during transfer outside the SAM.

The asset R.Signature is threatened.

Application Note 12 (Application Note 19 from EN 419241-2: Applied)

The modification of a signature can be detected by the SSA or any relying party by validation of the signature.

3.2.2.4 System

T.PRIVILEGED_USER_INSERTION

An attacker is able to create R.Privileged_User including R.Reference_Privileged_User_Authentication_Data and is able to log on to the SAM as a Privileged User. The assets R.Privileged_User and R.Reference_Privileged_User_Authentication_Data are threatened.

T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION

(abbreviated as T.REF_PRIV_U_AUTH_DATA_MOD)

An attacker modifies R.Reference_Privileged_User_Authentication_Data and is able to log on to the SAM as the Privileged User. The asset R.Reference_Privileged_User_Authentication_Data is threatened.

T.AUTHORISATION_DATA_UPDATE

Attacker impersonates Privileged User and updates R.Authorisation_Data and may be able to activate a signing key. The assets R.Authorisation_Data and R.Signing_Key_Id are threatened.

Application Note 13 (Application Note 20 from EN 419241-2: Applied)

In some applications, it may be sufficient for an attacker with access to R.Authorisation_Data and R.Signing_Key_Id to activate the signing key within the Cryptographic Module. Since the R.Signing_Key_Id is only to be protected in integrity and not in confidentiality, access to R.Authorisation_Data should only be allowed for authorised operators.

Application Note 14

In the case of the Trident Privileged User cannot update R.Authorisation_Data, then this threat is not relevant.

T. AUTHORISATION_DATA _DISCLOSE

(abbreviated as AUTHORISATION_DATA _DISCL)

Attacker discloses R.Authorisation_Data during update and is able to activate a signing key.

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

T.CONTEXT_ALTERATION

An attacker modifies system configuration R.TSF_DATA to perform an unauthorised operation.

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

T.AUDIT_ALTERATION

An attacker modifies system audit and is able hide trace of SAM modification or usage.

The assets R.SVD, R.SAD, R.Signer, R.Reference_Signer_Authentication_Data, R.DTBS/R, R.Signature, R.AUDIT and R.TSF_DATA are threatened.

T.RANDOM

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

3.2.3 Additional threats

There are some additional threats for the distributed configuration of the TOE:

T.Inconsistency *Inconsistency of TSF data*

The TSF data may become inconsistent if the internal channel between parts of the TOE (MPCAs) becomes inoperative (e.g. internal TOE network connections are broken or any MPCA becomes disabled).

T.Intercept *Intercept of the internal communication*

An attacker may acquire access to and/or modify sensitive information (R.SecretKey, R.ClientData, R.RAD, R.Authorisation_Data, R.SAD, R.Random) while these are being transmitted between TOE parts (MPCAs).

T.Breakdown *Breakdown in one of the MPCAs*

The TOE may not provide normal service to users due to external attacks or a fatal error in one of the TOE parts.

T.Update_Compromise *Compromised update of the software or firmware*
Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

3.3 Organizational Security Policies

The TOE shall comply with following Organizational Security Policies as security rules, procedures, practices, or guidelines imposed by an organization upon its operations.

3.3.1 Organizational Security Policies for the Cryptographic Module (CM)

P.Algorithms *Use of approved cryptographic algorithms*

The CM 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 15 (Application Note 1 from EN 419221-5: Applied)

The relevant authorities and endorsements are determined by the context of the client applications that use the CM. For digital signatures within the European Union this is as indicated in [eIDAS] and an exemplary list of algorithms and parameters is given in [TS 119312] or [SOGIS].

P.KeyControl *Support for control of keys*

The life cycle of the CM and any secret keys that it manages (where such keys are associated with specific entities, such as the signature creation data associated with a signatory or the seal creation data associated with a seal creator[11]), shall be implemented in such a way that the secret keys can be reliably protected by the legitimate owner against use by others, and in such a way that the use of the secret keys by the CM can be confined to a set of authorised cryptographic functions.

Application Note 16 (Application Note 2 from EN 419221-5: Applied)

[11] A seal creator may be a legal person (see [eIDAS]) 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.

This policy is intended to ensure that the CM can be used for qualified electronic seals and qualified electronic signatures as in [eIDAS], but recognises that not all keys are used for such purposes. Therefore, although the CM must 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 CM 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 CM is required to generate an audit trail of security-relevant events, recording the event details and the subject associated with the event.

Application Note 17 (Application Note 3 from EN 419221-5: Applied)

The CM is assumed to be part of a larger system that manages audit data. The CM therefore logs audit records, and it is assumed that these are collected, maintained and reviewed in the larger system. Hence there is no separate auditor role within the CM, but the role of System Auditor is assumed to exist in the larger system.

3.3.2 Organizational Security Policies for the Signature Activation Module (SAM)

P.RANDOM

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

Application Note 18

This Organizational Security Policy is covered by P.RNG (OSP for CM).

P.CRYPTO

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

Application Note 19 (Application Note 21 from EN 419241-2: Applied)

For cryptographic algorithms within the European Union this is as indicated in [eIDAS] and an exemplary list of algorithms and parameters is given in [TS 119312] or [SOGIS].

Application Note 20

Since the SAM is implemented as a local application within the same physical boundary as the CM defined in [EN 419221-5] then objective OT.Algorithm enforces the P.CRYPTO (instead of the objective for the operational environment OE.CRYPTOMODULE_CERTIFIED).

P.BACKUP

The SAM is required to provide backup functionality. The backup process shall preserve the confidentiality and integrity of the data during creation, transmission, storage and restoration of the backup data

3.4 Assumptions

3.4.1 Assumptions for the Cryptographic Module (CM)

A.ExternalData *Protection of data outside CM control*

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

In particular, any backups of the CM 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 CM. 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 CM 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 CM operates in a protected environment that limits physical access to the CM to authorised Administrators. The CM software and hardware environment (including client applications) is installed maintained by Administrators in a secure state that mitigates against the specific risks applicable to the deployment environment.

A.DataContext *Appropriate use of CM functions*

Any client application using the cryptographic functions of the CM 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 CM 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 CM according to the specific risks applicable to the deployment environment and the ways in which the CM is used.

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 CM cryptographic functions and parameters, and appropriate management and administration actions on the CM. This includes, for example, any relevant policies on algorithms, key generation methods, key lengths, key access, key import/export, key usage limitations, key activation, cryptoperiods and key renewal, and key/certificate revocation.

A.UAuth *Authentication of application users*

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

A.AuditSupport *Audit data review*

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

Application Note 21 (Application Note 4 from EN 419221-5: Applied)

As noted for P.Audit in section 3.3.1 the CM is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.

3.4.2 Assumptions for the Signature Activation Module (SAM)

A.PRIVILEGED_USER

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

A.SIGNER_ENROLMENT

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

A.SIGNER_AUTHENTICATION_DATA_PROTECTION (A.SIG_AUTH_DATA_PROT)

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

A.SIGNER_DEVICE

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

A.CA

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

A.ACCESS_PROTECTED

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

It is assumed that any audit generated by the SAM are only handled by authorised personal in a physical secured environment. The personal that carries these activities should act under established practices.

It is assumed that where copies of data protected by the SAM are managed outside of the SAM, client applications and other entities must provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment.

Application Note 22

There are no copies of data protected by the SAM, managed outside the SAM.

A.AUTH_DATA

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

A.CRYPTO

It is assumed that the SAM shall only use algorithms, algorithm parameters and key lengths endorsed by recognized authorities as appropriate by TSPs. This includes generation of random numbers, signing key pairs and signatures as well as the integrity and confidentiality of SAM assets.

Application Note 23 (Application Note 22 from EN 419241-2: Applied)

For cryptographic algorithms within the European Union this is as indicated in [eIDAS] and an exemplary list of algorithms and parameters is given in [TS 119312] or [SOGIS].

A.TSP_AUDITED

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

A.SEC_REQ

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

4 Security Objectives

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

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

4.1 Security Objectives for the TOE

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

4.1.1 Security Objectives for the Cryptographic Module (CM)

OT.PlainKeyConf *Protection of confidentiality of plaintext secret keys*

The plaintext value of secret keys is not made available outside the CM (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 CM cannot directly access the plaintext value of a secret key.

OT.Algorithms *Use of approved cryptographic algorithms*

The CM 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 24 (Application Note 5 from EN 419221-5: Applied)

See note under P.Algorithms (section 3.3.1) 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 CM 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 CM functions and data*

The CM carries out an authentication/authorisation check on all subjects before allowing them to use the CM. 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 CM
- users of CM cryptographic functions (client applications using secure channels)
- users of secret keys.

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

Application Note 25 (Application Note 6 from EN 419221-5: Applied)

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 CM. 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 CM rejects any attempt to use the key for a purpose that is not permitted. The CM 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 CM also prevents unauthorised use of any cryptographic functions that use a key.

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

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

Application Note 26 (Application Note 7 from EN 419221-5: Applied)

Such limits on the use of a key after initial authorisation are termed “re-authorisation conditions” in this PP. 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 must 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. see: FMT_MSA.3/Keys.

OT.DataConf *Protection of confidentiality of sensitive client application data*

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

Application Note 27 (Application Note 8 from EN 419221-5: Applied)

Protection of secret keys (as a specific type of sensitive data) is also subject to additional protection specified in other CM objectives. Any requirements for secure storage and control of access to other types of client application data within the CM 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 CM to perform cryptographic functions on data that represent a passphrase value and the passphrase value is to be stored on the CM, then the client application would need to use an appropriate encryption function before storing the data on the CM.

[12] 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 CM 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 CM.

Application Note 28 (Application Note 9 from EN 419221-5: Applied)

Any requirements for integrity protection of client application data within the CM 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 CM 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 must 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 must be encrypted). The CM 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 CM 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 by CM 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 CM shall provide features to protect its security functions against tampering. In particular the CM shall make any physical manipulation within the scope of the intended environment (adhering to OE.Env) detectable for the administrators of the CM.

OT.FailureDetect *Detection of CM hardware or software failures*

The CM 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 CM hardware components (including the RNG)
- Corruption of CM software.

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

OT.Audit *Generation of audit trail*

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

4.1.2 Security Objectives for the Signature Activation Module (SAM)

4.1.2.1 Enrolment

OT.SIGNER_PROTECTION

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

OT.REFERENCE_SIGNER_AUTHENTICATION_DATA

(abbreviated as OT.REF_SIG_AUTH_DATA)

The SAM shall be able to securely handle signature authentication data, R.Reference_Signer Authentication_Data, as part of R.Signer.

OT.SIGNER_KEY_PAIR_GENERATION

(abbreviated as OT.SIG_KEY_GEN)

The SAM shall be able to securely use the CM to generate signer signing key pairs and assign R.Signing_Key_Id and R.SVD to R.Signer.

OT.SVD

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

4.1.2.2 User Management

OT.PRIVILEGED_USER_MANAGEMENT

(abbreviated as OT.PRIV_U_MANAGEMENT)

The SAM shall ensure that any modification to R.Privileged_User and R.Reference_Privileged_User_Authentication_Data are performed under control of the Privileged User.

Application Note 29 (Application Note 23 from EN 419241-2: Applied)

The exception to this objective is when the initial (set of) Privileged Users are created as part of system initialisation.

OT.PRIVILEGED_USER_AUTHENTICATION

(abbreviated as OT.PRIV_U_AUTH)

The SAM shall ensure that an administrator with a Privileged User is authenticated before action on the SAM is performed.

OT.PRIVILEGED_USER_PROTECTION

(abbreviated as OT.PRIV_U_PROT)

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

OT.SIGNER_MANAGEMENT

The SAM shall ensure that any modification to R.Signer, R.Reference_Signer_Authentication_Data, R.Signing_Key_Id and R.SVD are performed under control of the signer or trusted administrator as Privileged User.

OT.SAM_BACKUP

Any method provided by the SAM for backing up user data, including R.Signing_Key_Id, R.Signer, R.Reference_Signer_Authentication_Data and R.Reference_Privileged_User_Authentication_Data

preserves the security of the data and is controlled by authorised Privileged Users. The secure backup process preserves the confidentiality and integrity of the data during creation, transmission, storage and restoration of the backup data.4.1.2.3 Usage

OT.SAD_VERIFICATION

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

Application Note 30 (Application Note 24 from EN 419241-2: Applied)

Where the SAM derives authorisation data from authentication data in the SAD and uses this to activate the signing key in the cryptographic module this function can depend on the controls provided by the cryptographic module.

Application Note 31 (Application Note 25 from EN 419241-2: Applied)

Requirements for authentication are described in [EN 419241-1] SRA_SAP.1.1.

OT.SAP

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

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

Application Note 32 (Application Note 26 from EN 419241-2: Applied)

The signer authentication is conducted according to [EN 419241-1] SCAL.2 for qualified signatures. The signer authentication is carried out in one of the following ways: (1) Directly by the SAM. In this case the SAM verifies the signer’s authentication factor(s). (2) Indirectly by the SAM. In this case an external authentication service as part of the TW4S or a delegated party that verifies the signer’s authentication factor(s) and issues an assertion that the signer has been authenticated. The SAM shall verify the assertion. (3) A combination of the two directly or indirectly schemes.

OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION

(abbreviated as OT.SIG_AUTH_DATA_PROT)

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

OT.DTBSR_INTEGRITY

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

OT.SIGNATURE_INTEGRITY

(abbreviated as OT.SIGN_INTEGRITY)

The SAM shall ensure that a signature can’t be modified inside the SAM.

OT.CRYPTO

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

4.1.2.3 System

OT.RANDOM

Random numbers generated by the TOE for use as keys, in protocols or seed data for another random number generator that is used for these purposes shall meet a defined quality metric in order to ensure that random numbers are not predictable and have sufficient entropy.

Application Note 33

This security objective is covered by OT.RNG (security objective for CM).

According to Application Note 39 in [EN 419241-2] the SFR FCS_RNG.1 (and OT.RNG) only apply, if the SAM is not implemented as a local application within the same physical boundary as the CM.

OT.SYSTEM_PROTECTION

The SAM shall ensure that modification to R.TSF_DATA is authorised by Privileged User and that unauthorised modification can be detected.

Application Note 34 (Application Note 27 from EN 419241-2: Applied)

The detection of unauthorised changes to R.TSF_DATA is only relevant if whole or part of it is stored outside the TOE. Since the Trident stores R.TSF_DATA, this objective is not relevant.

OT.AUDIT_PROTECTION

The SAM shall ensure that modifications to R.AUDIT can be detected.

4.1.3 Additional Security Objectives for the TOE

There are three additional Security Objectives for the distributed configuration of the TOE in relation to the distributed structure of the TOE:

OT.TSF_Consistency *Internal TSF consistency*

The TOE (CM+SAM) shall ensure the consistency of TSF data that are replicated between separate parts of the TOE.

OT.PROT_Comm *Protected communication between separate TOE parts*

The TOE (CM+SAM) shall provide protected communication channels between separate parts of the TOE.

OT.Availability *Partial Fault Tolerance*

The TOE (CM+SAM) shall provide normal service by maintaining the minimum security function at occurance of breakdown in one of the TOE parts by external attacks or a fatal error in one TOE part.

OT.Updates *Trusted Updates*

The TOE firmware and software is updated by an Administrator in response to the release of product updates due to new functionality. A secure update mechanism ensures the firmware and software are authorized through verification of their integrity and authenticity.

4.2 Security Objectives for the Operational Environment

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

4.2.1 SOs for the Operational Environment of the TOE (CM+SAM)

OE.Env *Protected operating environment*

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

The TOE (CM+SAM) shall operate in a protected environment that limits physical access to the TOE (CM+SAM) to authorised privileged users. The TOE (CM+SAM) 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).

4.2.2 SOs for the Operational Environment of the Cryptographic Module (CM)

OE.ExternalData *Protection of data outside TOE control*

Where copies of data protected by the CM are managed outside of the CM, 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 CM (such as audit data and encrypted keys).

In particular, any backups of the CM 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 CM. 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 CM 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.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 CM; 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 CM 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 CM, to confirm that it relates to the correct DTBS.

Client applications shall be responsible for any required logging of the uses made of the CM 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 CM shall correctly and securely gather identification and authentication/authorisation data from its users and securely transfer it to the CM (protecting the confidentiality of the authentication/authorisation data as required) when required to authorise the use of CM assets and services.

OE.AuditSupport *Audit data review*

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

Application Note 35 (Application Note 4 from EN 419221-5: Applied)

As noted for P.Audit, the CM 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 CM cryptographic functions and parameters, and appropriate management and administration actions on the CM. This includes, for example, any relevant policies on algorithms, key generation methods, key lengths, key access, key import/export, key usage limitations, key activation, cryptoperiods and key renewal, and key/certificate revocation.

4.2.3 SOs for the Operational Environment of the Signature Activation Module (SAM)

OE.SVD_AUTHENTICITY

The operational environment shall ensure the SVD integrity during transmit outside the SAM to the CA.

OE.CA_REQUEST_CERTIFICATE (abbreviated as OE.CA_REQ_CERT)

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

OE.CERTIFICATE_VERFICATION (abbreviated as OE.CERT_VERFICATION)

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

OE.SIGNER_AUTHENTICATION_DATA (abbreviated as OE.SIG_AUTH_DATA)

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

OE.DELEGATED_AUTHENTICATION

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

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

If the signer is only authenticated using a delegated party, the TSP shall ensure that the secret key material used to authenticate the delegated party to the TOE shall reside in a certified cryptographic module consistent with the requirement as defined in [EN 419241-1] SRG_KM.1.1.

The audit of the qualified TSP according to EN 419 241-1 shall provide evidence that any delegated party meets requirements from EN 419 241-1 SRA_SAP.1.1. and optionally SRG_KM.1.1 in case the signer is only authenticated using a delegated party.

Application Note 36

The Trident supports delegated authentication.

The signer authentication is carried out in one of the following ways:

(1) Directly by the SAM. In this case the SAM verifies the signer's authentication factors (password and TOTP).

(2) Indirectly by the SAM. In this case a delegated party verifies both of the signer's authentication factor and issues an assertion that the signer has been authenticated.

(3) Partly indirectly by the SAM. In this case a delegated party verifies one of the signer's authentication factor and issues an assertion that the signer has been authenticated. The SAM

verifies this assertion and the other signer's authentication factor (password).

OE.DEVICE

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

OE.CRYPTOMODULE_CERTIFIED (abbreviated as OE.CM_CERTIFIED)

If the SAM is implemented as a local application within the same physical boundary as the CM defined in [EN 419-221-5] then the SAM relies on the CM for providing a tamper-protected environment and for cryptographic functionality and random number generation. If the CM is implemented within a separate physical boundary then the SAM relies on the CM for cryptographic functionality and random number generation. The physical boundary shall physically protect the SAM conformant to FPT_PHP.1 and FPT_PHP.3 in [EN 419 221-5].

Application Note 37 (Application Note 26 from [EN 419241-2]: Applied)

OE.CRYPTOMODULE_CERTIFIED requirement for the SAM is accomplished because this ST claims to be strictly conformant also to the PP [EN 419221-5].

In case of an extended CM is used, OE.CRYPTOMODULE_CERTIFIED is an objective for the operational environment.

OE.TW4S_CONFORMANT

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

4.3 Security Objectives Rationale

4.3.1 Security objectives coverage (backtracking)

The following tables show how the security objectives and the security objectives for the operational environment cover the threats, organizational security policies and assumptions, for the CM (4.1) for the SAM (4.2) and for the distributed structure of the TOE (4.3).

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit	OE.ExternalData	OE.Env	OE.DataContext	OE.AppSupport	OE.Uauth	OE.AuditSupport
T.KeyDisclose	X		X				X		X	X		X			X	X				
T.KeyDerive		X									X									
T.KeyMod			X						X	X		X								
T.KeyMisuse				X	X															
T.KeyOveruse						X														
T.DataDisclose							X										X	X		
T.DataMod								X									X	X		
T.Malfunction													X							
P.Algorithms		X																		
P.CRYPTO[13]		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

Table 4.1 Mapping of security problem definition to security objectives for CM

[13] P.CRYPTO is an OSP from [EN 419241-2]. Since the SAM is implemented as a local application within the same physical boundary as the CM defined in [EN 419221-5] then objective OT.Algorithm enforces the P.CRYPTO (instead of the objective for the operational environment OE.CRYPTOMODULE_CERTIFIED).

	Enrolment				User management				Usage							System			Security Objectives for the Operational Environment							
	OT.SIGNER_PROTECT	OT.REF_SIG_AUTH_D	OT.SIG_KEY_GEN	OT.SVD	OT.PRIV_U_MANAGE MENT	OT.PRIV_U_AUTH	OT.PRIV_U_PROT	OT.SIGNER_MANAGE MENT	OT.SAM_BACKUP	OT.SAD_VERIFICATIO	OT.SAP	OT.SIG_AUTH_DATA_	OT.DTBSR_INTEGRIT	OT.SIGN_INTEGRITY	OT.CRYPTO	OT.RNG (for CM)	OT.SYSTEM_PROTEC TION	OT.AUDIT_PROTECTI	OE.ENV	OE.SVD_AUTHENTICI	OE.CA_REQ_CERT	OE.CERT_VERIFICAT	OE.SIG_AUTH_DATA	OE.DEVICE	OE.CM_CERTFIED	OE.TW4S_CONFORM
T.ENROLMENT_SIGNER_IMPERSONAL	X	X						X																		X
T.ENR_SIG_AUTH_DATA_DISCL	X	X																					X	X		
T.SVD_FORGERY			X	X											X					X	X					
T.ADMIN_IMPERSONATION						X																				
T.MAINT_AUTH_DISCL		X																								
T.AUTH_SIG_IMPERS										X																
T.SIG_AUTH_DATA_MOD		X									X	X														
T.SAP_BYPASS											X													X		
T.SAP_REPLAY											X													X		
T.SAD_FORGERY											X	X											X	X		
T.SIGN_REQ_DISCL											X															
T.DTBSR_FORGERY													X											X		
T.SIGNATURE_FORGERY														X	X											
T.PRIVILEGED_USER_INSERTION					X	X																				
T.REF_PRIV_U_AUTH_DATA_MOD					X	X	X																			
T.AUTHORISATION_DATA_UPDATE																	X									
T. AUTHORISATION_DATA _DISCL																	X									
T.CONTEXT_ALTERATION																	X									
T.AUDIT_ALTERATION																		X								
T.RANDOM																X										
P.CRYPTO															X											
P.RANDOM																X										
P.BACKUP									X																	
A.PRIVILEGED_USER																										X

	Enrolment				User management				Usage							System			Security Objectives for the Operational Environment							
	OT.SIGNER_PROTECT	OT.REF_SIG_AUTH_D	OT.SIG_KEY_GEN	OT.SVD	OT.PRIV_U_MANAGE MENT	OT.PRIV_U_AUTH	OT.PRIV_U_PROT	OT.SIGNER_MANAGE MENT	OT.SAM_BACKUP	OT.SAD_VERIFICATIO	OT.SAP	OT.SIG_AUTH_DATA_	OT.DTBSR_INTEGRIT	OT.SIGN_INTEGRITY	OT.CRYPTO	OT.RNG (for CM)	OT.SYSTEM_PROTEC TION	OT.AUDIT_PROTECTI	OE.ENV	OE.SVD_AUTHENTICI	OE.CA_REQ_CERT	OE.CERT_VERIFICAT	OE.SIG_AUTH_DATA	OE.DEVICE	OE.CM_CERTFIED	OE.TW4S_CONFORM
A.SIGNER_ENROLMENT																			X							
A.SIG_AUTH_DATA_PROT																							X			
A.SIGNER_DEVICE																								X		
A.CA																					X					
A.ACCESS_PROTECTED																			X							
A.AUTH_DATA																								X		
A.CRYPTO																									X	
A.TSP_AUDITED																			X							
A.SEC_REQ																										X

Table 4.2 Mapping of security problem definition to security objectives for SAM

	OT.TSF_ Consistency	OT.PROT_ Comm	OT.Availability	OT.Updates
T.Inconsistency	X			
T.Intercept		X		
T.Breakdown			X	
T.Update_Compromise				X

Table 4.3 Mapping of security problem definition to security objectives for the distributed structure

4.3.2 Security Objectives Sufficiency

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

4.3.2.1 Sufficiency for the Cryptographic Module (CM)

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 CM, as in OE.ExternalData, and by protecting the operation of the CM 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 CM 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 CM 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 CM allows a user to define.

T.DataDisclose is concerned with the transmission of data between client applications and the CM, or between separate parts of the CVM where the transmission passes through an insecure environment. This is addressed by OT.DataConf, which requires the CM 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 CM, and this is addressed by OT.DataMod which requires that the CM 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 CM to detect certain types of fault.

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.CRYPTO requires the use of algorithm, algorithm parameters and key lengths that are endorsed by appropriate authorities, and this is addressed by OT.Algorithms.

P.KeyControl requires that the CM 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 CM 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 CM
- OT.Auth, OT.KeyUseConstraint and OT.KeyUseScope ensure that the CM can provide welldefined 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 CM to client applications or for backup purposes, including the prevention of export of Assigned Keys.

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

Each of the Assumptions in section 3.4.1 is directly matched by a security objective for the operational environment in section 4.2.1 and 4.2.2. The wording of each objective for the operational environment includes the wording of each assumption, and no further rationale is therefore given here.

4.3.2.2 Sufficiency for the Signature Activation Module (SAM)

T.ENROLMENT_SIGNER_IMPERSONATION is covered by OT.SIGNER_PROTECTION requiring R.Signer to be protected in integrity and for sensitive parts in confidentiality.

It is also covered by OT.SIGNER_MANAGEMENT requiring the signer to be securely created.

It is also covered by OT.REFERENCE_SIGNER_AUTHENTICATION_DATA requiring the SAM to be able to assign signer authentication data to the signer.

It is also covered by OE.TW4S_CONFORMANT as that requires that signer enrolment to be handled in accordance with [Assurance] for level at least substantial.

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

It is also covered by OT.SIGNER_PROTECTION requiring that the attributes, including signer authentication data, be protected in integrity and if needed in confidentiality.

It is also covered by OE.SIGNER_AUTHENTICATION_DATA requiring the signer to keep his authentication data secret.

It is also covered by OE.DEVICE requiring the device used by the signer not to disclose authentication data.

T.SVD_FORGERY is covered by OT.SIGNER_KEY_PAIR_GENERATION requiring a Cryptographic Module to generate signer key pair.

It is also covered by OT.SVD requiring the public key to be protected while inside the SAM.

It is also covered by OT.CRYPTO requiring the usage of endorsed algorithms. It is also covered by OE.SVD_AUTHENTICITY requiring the environment to protect the SVD during transmit from the

SAM to the CA.

It is also covered by OE.CA_REQUEST_CERTIFICATE requiring the certification request to be protected in integrity.

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

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

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

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

T.SAP_BYPASS is covered by OT.SAP requiring that all steps, including SAD verification, of the SAP must completed.

T.SAP_REPLAY is covered by OT.SAP requiring that the signature activation protocol must be able to resist whole or part of it being replayed.

T.SAD_FORGERY is covered by OT.SAP requiring the SAM to be able to detect if the SAD has been modified during transmit to the SAM.

It is also covered by OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION requiring signature authentication data to be protected during transmit to the SAM.

It is also covered by OE.SIGNER_AUTHENTICATION_DATA requiring the signer to protect his authentication data.

It is also covered by OE.DEVICE requiring the device used by the signer to participate correctly in the SAP, in particular the device shall not disclose authentication data.

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

T.DTBSR_FORGERY is covered by OT.DTBSR_INTEGRITY requiring the DTBS/R to be to be protected in integrity during transmit to the SAM.

It is also covered by OE.DEVICE requiring the device to participate correctly in the SAP, including sending the SAD containing a link to the data to be signed.

T.SIGNATURE_FORGERY is covered by OT.SIGNATURE_INTEGRITY requiring that the signature is protected in integrity inside the SAM.

It is also covered by OT.CRYPTO requiring the usage of endorsed algorithms.

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

User to be authenticated..

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

It is also covered by OT.PRIVILEGED_USER_PROTECTION requiring the Privileged User to be protected in integrity.

T.AUTHORISATION_DATA_UPDATE is covered by OT.SYSTEM_PROETECTION requiring any unauthorised modification to SAM configuration to be detectable.

T.AUTHORISATION_DATA_DISCLOSE is covered by OT.SYSTEM_PROETECTION requiring any unauthorised modification to SAM configuration to be detectable.

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

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

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

P.CRYPTO is covered by OT.CRYPTO requiring the usage of endorsed algorithms

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

P.BACKUP is covered by OT.SAM_BACKUP requiring random numbers to meet a specified quality metric.

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

A.SIGNER_ENROLMENT is covered by OE.TW4S_CONFORMANT requiring the operation of the TW4S enrolment of users in a secure way.

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

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

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

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

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

A.CRYPTO is covered by OE.CRYPTOMODULE_CERTIFIED.

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

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

4.3.2.3 Sufficiency for the additional threats

T.Inconsistency addresses the threat arising from inconsistency of TSF data stored in different TOE parts. This threat is countered by OT.TSF_Consistency, which ensures the consistency of TSF data that are replicated between separate TOE parts.

T.Intercept addresses the threat arising from interception of secure data while they are being transmitted between TOE parts. This threat is countered by OT.PROT_Comm, which assures the protection of communication channels between separate TOE parts.

T.Breakdown is covered by OT.Availability, which requires a minimum service provision to be maintain in case of one of the MPCAs has broken down.

T.Update_Compromise is covered by OT.Updates, which requires a secure update mechanism.

5 Extended components definition

5.1 Generation of random numbers (FCS_RNG)

The additional family FCS_RNG (Generation of random numbers) of the Class FCS (Cryptographic Support) is defined in [EN 419221-5] and [EN 419241-2].

Family behaviour

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

Component levelling:

FCS_RNG: Generation of random numbers - 1

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 38 (Application Note 11/29 from [EN 419221-5] / [EN 419241-2]: Applied)
A physical random number generator (RNG) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (key strokes, mouse movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs where a hybrid physical RNG produces at least the amount of entropy the RNG output may contain and the internal state of a hybrid deterministic RNG output contains fresh entropy but less than the output of RNG may contain.

5.2 Basic TSF Self Testing (FPT_TST_EXT)

The additional family FPT_TST_EXT (Basic TSF Self Testing) of the Class FPT (Protection of the TSF) is defined in [EN 419221-5].

Application Note 39

The [EN 419221-5] use FPT_TST_EXT, but according to [CC2] 7.1.2.1 (49):

“The categorical information consists of a short name of seven characters, with the first three identical to the short name of the class followed by an underscore and the short name of the family as follows XXX_YYY.

This ST uses same format as the certified Protection Profile.

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

Family behaviour

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

Component levelling:

FPT_TST_EXT Basic TSF Self Testing - 1

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 additional self-tests run by the TSF*].

5.3 Trusted Update (FPT_TUD_EXT.1)

The additional family FPT_TUD_EXT.1 (Trusted update) of the Class FPT (Protection of the TSF) is defined in [cPP ND].

Family behaviour

Components in this family address the requirements for updating the TOE firmware and/or

software.

Component levelling:

FPT_TUD_EXT.1	Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.
FPT_TUD_EXT.2	Trusted update based on certificates applies when using certificates as part of trusted update and requires that the update does not install if a certificate is invalid.

Management: FPT_TUD_EXT.1

The following actions could be considered for the management functions in FMT:

a. Ability to update the TOE and to verify the updates

b. Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1/digsig_Non-multiparty) and [selection: no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]]

c. Ability to update the TOE, and to verify the updates using [selection: digital signature, published hash, no other mechanism] capability prior to installing those updates

Audit: FPT_TUD_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is

included in the PP/ST:

a. Initiation of the update process.

b. Any failure to verify the integrity of the update

FPT_TUD_EXT.1 (Trusted Update)

Hierarchical to: No other components

Dependencies: FCS_COP.1/digsig_Non-multiparty Cryptographic operation (for Cryptographic Signature and Verification), or FCS_COP.1/hash Cryptographic operation (for cryptographic hashing)

FPT_TUD_EXT.1.1

The TSF shall provide [assignment: Administrators] the ability to query the currently executing version of the TOE firmware/software and [selection: the most recently installed version of the TOE firmware/software; no other TOE firmware/software version].

FPT_TUD_EXT.1.2

The TSF shall provide [assignment: Administrators] the ability to manually initiate updates to TOE firmware/software and [selection: support automatic checking for updates, support automatic updates, no other update mechanism].

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.

6 Security requirements

6.1 Security functional requirements

6.1.1 Use of requirement specifications

Common Criteria allows several operations to be performed on functional requirements; refinement, selection, assignment, and iteration are defined in paragraph 2.1.4 of [CC2]. Each of these operations is used in this ST.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is either (i) denoted by the word "refinement" in **bold** text and the added or changed words are in bold text, or (ii) included in text as **bold** text and marked by a footnote. In cases where words from a CC requirement were deleted, a separate attachment indicates the words that were removed.

The s**election** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the PP authors or CC authors are denoted as underlined text and the original text of the component is given by a footnote. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made, [selection:], and are italicized. Selections filled in by the ST author are denoted as double underlined text and a foot note where the selection choices from the PP are listed.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the PP authors are denoted by showing as underlined text and the original text of the component is given by a footnote. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:], and are italicized. In some cases, the assignment made by the PP authors defines a selection to be performed by the ST author. Thus, this text is italicized like this. Assignments filled in by the ST author are denoted as double underlined text.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier.

For a distributed TOE, the functional security requirements need to be met by the TOE as a whole, but not all SFRs will necessarily be implemented by all TOE parts. The following categories are defined in order to specify when SFRs are to be implemented by one or all TOE parts:

- **All parts separately ('All')** – All TOE parts that comprise the distributed TOE must independently satisfy the requirement.
- **At least one part ('One')** – This requirement must be fulfilled by at least one part within the distributed TOE.
- **All parts together** ('**Distributed**') – This requirement must be fulfilled jointly by all TOE parts, in a distributed way.

In the case of the Trident:

- **Table 6.1.** specifies how each of the SFRs in this ST must be met, using the categories above. 'One' category means that this requirement must be fulfilled by the MPCA addressed by (local or external) client application.

Description	CM	SAM	Additional SFRs	Distri-buted structure
Security audit data generation (FAU)				
Audit data generation	FAU_GEN.1/CM	FAU_GEN.1/SAM		All
User identity association	FAU_GEN.2/CM	FAU_GEN.2/SAM		All
Guarantees of audit data availability	FAU_STG.2	-		All
Cryptographic support (FCS)				
Cryptographic key generation	FCS_CKM.1/key_gen_Non-multiparty FCS_CKM.1/key_gen_Multiparty	FCS_CKM.1/invoke_CM:key gen_Non-multiparty FCS_CKM.1/invoke_CM:key gen_Multiparty		All Distributed
Cryptographic key distribution	-	-	FCS_CKM.2/ Non-multiparty FCS_CKM.2/ Multiparty	All All
Cryptographic key destruction	FCS_CKM.4/CM	FCS_CKM.4/SAM		All
Cryptographic operation	FCS_COP.1/digsig_Non-multiparty FCS_COP.1/digsig_Multiparty FCS_COP.1/enc-dec_Non-multiparty FCS_COP.1/dec_Multiparty FCS_COP.1/hash FCS_COP.1/TOTP_verification FCS_COP.1/cmac operation	FCS_COP.1/invoke_CM:digsig_Non-multiparty FCS_COP.1/invoke_CM:digsig_Multiparty - - FCS_COP.1/SAM_hash FCS_COP.1/SAM_TOTP_verification - FCS_COP.1/SAM_key_derivation		One Distributed One Distributed One One One One
Generation of random numbers	FCS_RNG.1	-		One
User data protection (FDP)				

Description	CM	SAM	Additional SFRs	Distributed structure
Subset access control	FDP_ACC.1/KeyUsage	-		All
	FDP_ACC.1/CM_Backup	-		All
	-	FDP_ACC.1/Privileged User Creation		All
	-	FDP_ACC.1/Signer Creation		All
	-	FDP_ACC.1/Signer Key Pair Generation		All
	-	FDP_ACC.1/Signer Maintenance		All
	-	FDP_ACC.1/Signer Key Pair Deletion		All
	-	FDP_ACC.1/Supply DTBS/R		All
	-	FDP_ACC.1/Signing		All
	-	FDP_ACC.1/SAM Maintenance		All
		FDP_ACC.1/SAM Backup		All
Security attribute-based access control	FDP_ACF.1/KeyUsage	-		All
	FDP_ACF.1/CM_Backup	-		All
	-	FDP_ACF.1/Privileged User Creation		All
	-	FDP_ACF.1/Signer Creation		All
	-	FDP_ACF.1/Signer Key Pair Generation		All
	-	FDP_ACF.1/Signer Maintenance		All
	-	FDP_ACF.1/Signer Key Pair Deletion		All
	-	FDP_ACF.1/Supply DTBS/R		All
	-	FDP_ACF.1/Signing		All
	-	FDP_ACF.1/SAM Maintenance		All
		FDP_ACF.1/SAM Backup		All
Subset information flow control	FDP_IFC.1/KeyBasics	-		All
	-	FDP_IFC.1/Signer		All
	-	FDP_IFC.1/Privileged User		All
Simple security attributes	FDP_IFF.1/KeyBasics	-		All
	-	FDP_IFF.1/Signer		All
	-	FDP_IFF.1/Privileged User		All
Export of user data with security attributes	-	FDP_ETC.2/Signer		All
	-	FDP_ETC.2/Privileged User		All
Import of user data with security attributes		FDP_ITC.2/Signer		All
		FDP_ITC.2/Privileged User		All
Stored data integrity monitoring and action	FDP_SDI.2	-		All
Subset residual information protection	FDP_RIP.1	-		All

Description	CM	SAM	Additional SFRs	Distri-buted structure
Basic data exchange confidentiality	-	FDP_UCT.1		All
Data exchange integrity	-	FDP_UIT.1		All
Identification and authentication (FIA)				
Authentication failure handling	FIA_AFL.1/CM_authentication	-		All
	FIA_AFL.1/CM_authorisation	-		All
	-	FIA_AFL.1/SAM		All
Timing of identification	FIA_UID.1/CM	-		One
		FIA_UID.2/SAM		One
Timing of authentication	FIA_UAU.1/CM			One
	-	FIA_UAU.1/SAM		One
Multiple authentication mechanisms	-	FIA_UAU.5/Signer		One
	-	FIA_UAU.5/Privileged User		One
Re-authenticating	FIA_UAU.6/KeyAuth	-		One
	FIA_UAU.6/GenKeyAuth	-		One
User attribute definition	-	FIA_ATD.1		All
User-subject binding	-	FIA_USB.1		All
Security management (FMT)				
Management of security attributes	FMT_MSA.1/GenKeys	-		All
	FMT_MSA.1/AKeys	-		All
	-	FMT_MSA.1/Signer		All
	-	FMT_MSA.1/Privileged User		All
Secure security attributes	-	FMT_MSA.2		All
Static attribute initialization	FMT_MSA.3/Keys	-		All
	-	FMT_MSA.3/Signer		All
	-	FMT_MSA.3/Privileged User		All
Management of TSF data	FMT_MTD.1/Unblock	-		All
	FMT_MTD.1/AuditLog	-		All
	-	FMT_MTD.1/SAM		All

Description	CM	SAM	Additional SFRs	Distri-buted structure
Security management functions	FMT_SMF.1/CM	FMT_SMF.1/SAM		All
Security roles	FMT_SMR.1/CM	FMT_SMR.2/SAM		All
			FMT_MOF.1/ ManualUpdate	One
Protection of the TSF (FPT)				
Reliable time stamps	FPT_STM.1/CM	FPT_STM.1/SAM		All
Failure with preservation of secure state	FPT_FLS.1	-		All
Passive detection of physical attack	FPT_PHP.1	-		All
Resistance to physical attack	FPT_PHP.3	-		All
Basic TSF Self Testing	FPT_TST_EXT.1	-		All
Replay detection	-	FPT_RPL.1		One
Inter-TSF basic TSF data consistency		FPT_TDC.1		All
Internal TSF consistency			FPT_TRC.1	All
Mutual trusted acknowledgement			FPT_SSP.2	All
Basic Internal TSF Data Transfer Protection			FPT_ITT.1	All
Trusted Update			FPT_TUD_EXT.1	One
Resource utilisation (FRU)				
Degraded fault tolerance			FRU_FLT.1	All
Limited fault tolerance			FRU_FLT.2	One
Trusted path/channels (FTP)				

Description	CM	SAM	Additional SFRs	Distri-buted structure
Trusted path	FTP_TRP.1/Local	-	-	One
	FTP_TRP.1/Admin	-	-	One
	FTP_TRP.1/External	-	-	One
	-	FTP_TRP.1/SSA	-	One
	-	FTP_TRP.1/SIC	-	One
			FTP_TRP.1/ Trusted Update	One
Inter-TSF trusted channel		FTP_ITC.1/CM		One

Table 6.1 Functional Security Requirements for the distributed structure of the Trident

6.1.2 SFRs of the Cryptographic Module (CM)

6.1.2.1 Security audit data generation (FAU)

FAU_GEN.1/CM (Audit data generation)

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1/CM

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[14] level of audit;

c) Startup of the TOE;

d) Shutdown of the TOE;

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

f) Cryptographic key destruction (FCS_CKM.4/CM);

g) Failure of the random number generator (FCS_RNG.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/CM_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);

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

m) Failures to establish secure channels (FTP_TRP.1/Local, **FTP_TRP.1/Admin**[15], 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) Failures to establish secure channels among different TOE parts,

t) Pre-generation of prime numbers for the RSA key-pairs,

u) **Initiation of the update process,**

v) **Any failure to verify the integrity of the update,**

w) Cryptographic key distribution (FCS_CKM.2)[16].

FAU_GEN.1.2/CM

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 ST: identifier of the related MPCA, human readable descriptive string about the related event[17].

FAU_GEN.2/CM (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/CM

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

[15] [refinement]

[16] [assignment: other specifically defined auditable events]

[17] [assignment: other audit relevant information]

The TSF shall be able to detect[18] unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.2.3

The TSF shall ensure that all[19] stored audit records will be maintained when the following conditions occur: audit storage exhaustion[20].

6.1.2.2 Cryptographic support (FCS)

FCS_CKM.1/key gen_Non-multiparty (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/key gen_Non-multiparty

The TSF shall generate cryptographic keys in accordance with **the**[21] specified cryptographic key generation algorithms **listed below**[22] and specified cryptographic key sizes **specified for each algorithm**[23] that meet the following **standards noted for each algorithm**[24].

key generation algorithm	cryptographic key sizes (bits)	standard
RSA Key Gen - probable prime	2048, 3072, 4096, 6144	[FIPS 186-5]
ECDSA keyGen with Curve: P-224, P-256, P-384, P-521 secretGenerationMode: testing candidates	224, 256, 384, 521	[FIPS 186-5]
ECDSA keyGen with Curve: brainpoolP224r1, brainpoolP256r1, brainpoolP320r1 brainpoolP384r1, brainpoolP512r1	224, 256, 320, 384, 512	[SP800-186]
EdDSA keyGen with Curve: ED25519, ED448	256, 456	[SP800-186]
EdDSA keyGen with Curve: brainpoolP224r1, brainpoolP256r1, brainpoolP320r1 brainpoolP384r1, brainpoolP512r1	224, 256, 320, 384, 512	[SP800-186]
ML-DSA.KeyGen_internal **ML-DSA-44, ML-DSA-65, ML-DSA-87**	private key: 2560, 4032, 4896 public key: 1312, 1952, 2592	[FIPS 204]
key generation using an approved random number generator, for **AES keys** **3DES keys** **ARIA keys**	 256 168 128, 192, 256	[SP800-57]

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

[19] [assignment: metric for saving audit records]

[20] [selection: audit storage exhaustion, failure, attack]

[21] [refinement]

[22] [assignment: cryptographic key generation algorithm]

[23] [assignment: cryptographic key sizes]

[24] [assignment: list of standards]

SEED keys **TOTP shared secret**	128 256	
SPHINCS+ key pairs generation **(SK.seed, PK.seed) and (SK.prf, PK.prf)**	512, 1024	[NIST IR.8240] [SPHINCS+] [FIPS 205]
ML-KEM (Kyber) key encapsulation mechanism **ML-KEM-512, ML-KEM-768, ML-KEM-1024** **Kyber512, Kyber768, Kyber1024** **Kyber512-90s, "Kyber768-90s, Kyber1024-90s**	length of shared key: 256	[FIPS 203] [Kyber], [NIST IR.8413] https://github.com/pq-crystals/kyber
NTRU key encapsulation mechanism	length of shared key: 256	[NTRU], https://github.com/jschanck/ntru https://ntru.org
PRF for **TLS master secrets**	384	[RFC 5246]
key derivation: **Balloon**	length of password	[Balloon]
key derivation: **PBKDF2**	length of password (characters) min: 8	[SP800-132]
key derivation: **HKDF** Mode: ANSIX9.63 Supported hash algorithms: SHA2-(256, 384, 512), SHA3-(256, 384, 512)	128, 192, 256	[SP800-135r1]

FCS_CKM.1/key gen_Multiparty (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/Key generation_Multiparty

The TSF shall generate cryptographic keys in accordance with **the**[25] specified cryptographic key generation algorithm**s listed below**[26] and specified cryptographic key sizes **specified for each algorithm**[27] that meet the following **standards noted for each algorithm**[28].

key generation algorithm	cryptographic key sizes (bits)	standard
RSA Key Gen - probable prime	2048, 3072, 4096, 6144	[FIPS 186-5]
ECDSA keyGen with Curve: P-224, P-256, P-384, P-521 secretGenerationMode: extra bits	224, 256, 384, 521	[FIPS 186-5]
ECDSA keyGen with Curve: brainpoolP224r1, brainpoolP256r1, brainpoolP320r1 brainpoolP384r1, brainpoolP512r1	224, 256, 384, 320, 512	[SP800-186]
key derivation: **PBKDF**	length of password (characters)	[SP800-132]

[25] [refinement]

[26] [assignment: cryptographic key generation algorithm]

[27] [assignment: cryptographic key sizes]

[28] [assignment: list of standards]

	min: 8	
key derivation: **KDF** Mode: ANSIX9.63 Supported hash algorithms: SHA2-(256, 384, 512), SHA3-(256, 384, 512)	128, 192, 256	[SP800-135r1]

FCS_CKM.4/CM (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/CM

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method zeroization[29] that meets the following: [FIPS 140-3], and [ISO19790], section 7.9.7[30].

FCS_COP.1/dig sig_Non_multiparty (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/dig sig_Non-multiparty

The TSF shall perform generation and verification of digital signature and seal[31] in accordance with **the**[32] specified cryptographic algorithm**s listed below**[33] and cryptographic key sizes **specified for each algorithm**[34] that meet the following: **standards noted for each algorithm**[35]:

cryptographic algorithm	cryptographic key sizes (bits)	standard
RSA SigGen and **RSA SigVer RSASSA-PKCS1-v1_5** with SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512)	2048, 3072, 4096	[FIPS 186-5]
RSA SigGen and **RSA SigVer RSASSA-PSS** with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256) and with maskFunction: MGF1, SHAKE-(128, 256)	2048, 3072, 4096	[FIPS 186-5]
ECDSA sigGen and **ECDSA sigVer** with Curve: P-224, P-256, P-384, P-521 and with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256)	224, 256, 384, 521	[FIPS 186-5]
ECDSA sigGen and **ECDSA sigVer**	224, 256, 384, 512	[FIPS 186-5]

[29] [assignment: cryptographic key destruction method]

[30] [assignment: list of standards]

[31] [assignment: list of cryptographic operations]

[32] [refinement]

[33] [assignment: cryptographic algorithm]

[34] [assignment: cryptographic key sizes]

[35] [assignment: list of standards]

with Curve: brainpoolP224r1, brainpoolP256r1, brainpoolP320r1 brainpoolP384r1, brainpoolP512r1 and with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256)		
ECDSA sigGen and **ECDSA sigVer** with Curve: K-(233, 283, 409, 571), B-(233, 283, 409, 571), brainpoolP(224, 256, 320, 384, 512)t1, prime239v(1,2,3), c2tnb239v(1,2,3), c2pnb(272, 304, 368)w1, c2tnb359v1, c2tnb431r1 and with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256)	224, 233, 239, 256, 272, 283, 304, 320, 359, 368, 384, 409, 431, 512, 571	[FIPS 186-5]
EdDSA sigGen and **EdDSA sigVer** with Curve: ED25519 and with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256) and with Curve: ED448 and with Hash: SHAKE-(128, 256)	256 and 456	[FIPS 186-5] [RFC 8032]
EdDSA sigGen and **EdDSA sigVer** with Curve: brainpoolP224r1, brainpoolP256r1, brainpoolP320r1 brainpoolP384r1, brainpoolP512r1 and with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256)	224, 256, 384, 512	[FIPS 186-5]
SPHINCS+ SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128s, SLH-DSA-SHA2-128f, SLH-DSA-SHAKE-128f SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192s,SLH-DSA-SHA2-192f SLH-DSA-SHAKE-192f SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256s, SLH-DSA-SHA2-256f SLH-DSA-SHAKE-256f	128 192 256	[FIPS 205]
Schnorr with Curve: P-(224, 256, 384, 521), K-(233, 283, 409, 571), B-(233, 283, 409, 571), brainpoolP(224, 256, 320, 384, 512)r1, brainpoolP(224, 256, 320, 384, 512)t1, prime239v(1,2,3), c2tnb239v(1,2,3), c2pnb(272, 304, 368)w1, c2tnb359v1, c2tnb431r1 and with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256)	224, 233, 239, 256, 272, 283, 304, 320, 359, 368, 384, 409, 431, 512, 521, 571	[Schnorr]
ML-DSA **ML-DSA-44,** **ML-DSA-65,** **ML-DSA-87**	(private, public) (2560, 1312) (4032, 1952) (4896, 2592)	[FIPS 204]

FCS_COP.1/dig sig_Multiparty (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/dig sig_Multiparty

The TSF shall perform generation of digital signature and seal[36] in accordance with **the**[37] specified

[36] [assignment: list of cryptographic operations]

[37] [refinement]

cryptographic algorithms **listed below**[38] and cryptographic key sizes **specified for each algorithm**[39] that meet the following: **standards noted for each algorithm**[40]:

cryptographic algorithm	cryptographic key sizes (bits)	standard
RSA SigGen RSASSA-PKCS1-v1_5 with SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512)	2048, 3072, 4096	[FIPS 186-5]
RSA SigGen RSASSA-PSS with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256) and with maskFunction: MGF1, SHAKE-(128, 256)	2048, 3072, 4096	[FIPS 186-5]
ECDSA sigGen with Curve: P-224, P-256, P-384, P-521 and with Hash: SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512), SHAKE-(128, 256)	224, 256, 384, 521	[FIPS 186-5]

FCS_COP.1/enc-dec _Non-multiparty (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/enc-dec_Non-multiparty

The TSF shall perform encryption and decryption[41] in accordance with **the**[42] specified cryptographic algorithms **listed below**[43] and cryptographic key sizes **specified for each algorithm**[44] that meet the following: **standards noted for each algorithm**[45]:

cryptographic algorithm	cryptographic key sizes (bits)	standard
AES supported modes: CBC, CCM, CFB1, CFB8, CFB, CTR, ECB, GCM, OFB, XTS	128, 192, 256	[FIPS 197] [SP800-38A] [SP800-38D] [SP800-38E]
3DES supported modes: ECB, CBC, CFB1, CFB8, CFB, OFB	192[46]	[SP800-67r2] [SP800-38A]
ARIA supported modes: ECB, CBC, CFB1, CFB8, CFB128, OFB, CTR, GCM	128, 192, 256	[RFC 5794] [SP800-38A] [SP800-38D]
SEED	128	[RFC 4269]

[38] [assignment: cryptographic algorithm]

[39] [assignment: cryptographic key sizes]

[40] [assignment: list of standards]

[41] [assignment: list of cryptographic operations]

[42] [refinement]

[43] [assignment: cryptographic algorithm]

[44] [assignment: cryptographic key sizes]

[45] [assignment: list of standards]

[46] Only decryption is supported

supported modes: ECB, CBC, CFB, OFB		
RSA RSAES-PKCS1-v1_5	encrypt: 2048[47], decrypt: 2048, 3072, 4096, 6144	[PKCS#1]

FCS_COP.1/dec_Multiparty (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/dec_Multiparty

The TSF shall perform decryption[48] in accordance with **the**[49] specified cryptographic algorithm**s listed below**[50] and cryptographic key sizes **specified for each algorithm**[51] that meet the following: **standards noted for each algorithm**[52]:

cryptographic algorithm	cryptographic key sizes (bits)	standard
RSA RSAES-PKCS1-v1_5	2048, 3072, 4096, 6144	[PKCS#1]

FCS_COP.1/hash (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/hash

The TSF shall perform cryptographic hash function[53] in accordance with a specified cryptographic algorithm SHA-256, SHA-384, SHA-512, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256 [54] and cryptographic key sizes none[55] that meet the following: [TS 119312], [FIPS 180-4] and [FIPS 202][56].

[47] Only for internal use

[48] [assignment: list of cryptographic operations]

[49] [refinement]

[50] [assignment: cryptographic algorithm]

[51] [assignment: cryptographic key sizes]

[52] [assignment: list of standards]

[53] [assignment: list of cryptographic operations]

[54] [assignment: cryptographic algorithm]

[55] [assignment: cryptographic key sizes]

[56] [assignment: list of standards]

FCS_COP.1/TOTP_verification (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/TOTP_verification

The TSF shall perform TOTP verification[57] in accordance with a specified cryptographic algorithm HOTP[58] and cryptographic key sizes 256 bits[59] that meet the following: [RFC 4226] and [RFC 6238][60].

FCS_COP.1/cmac operation (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/cmac operation

The TSF shall perform message authentication code operation[61] in accordance with **the[62]** specified cryptographic algorithm**s listed below**[63] and cryptographic key sizes **specified for each algorithm**[64] that meet the following **standards noted for each algorithm**[65]**:**

cryptographic algorithm	cryptographic key sizes (bits)	standard
AES-CCM	128, 192, 256	[SP800-38C]
AES-CMAC	128, 192, 256	[SP800-38B]
HMAC Hash function: SHA-1, SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512) Secret Key: K	length of K (bits) min: 8 max: 524288	[FIPS 198-1]
3DES-CMAC	192	[SP800-67] [SP800-38B]
ARIA-CMAC	128, 192, 256	[RFC 5794] [SP800-38B]

[57] [assignment: list of cryptographic operations]

[58] [assignment: cryptographic algorithm]

[59] [assignment: cryptographic key sizes]

[60] [assignment: list of standards]

[61] [assignment: list of cryptographic operations]

[62] [refinement]

[63] [assignment: cryptographic algorithm]

[64] [assignment: cryptographic key sizes]

[65] [assignment: list of standards]

FCS_RNG.1 (Generation of random numbers)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1

The TSF shall provide a **CTR_DRBG**[66] hybrid deterministic[67] random number generator that implements:

(DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2[68] as random source.

(DRG.4.2) The RNG provides forward secrecy.

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

(DRG.4.4) The RNG provides enhanced forward secrecy after 100 days or after 2^34 strings of bit length 128 whichever occurs first.

(DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2[69].

FCS_RNG.1.2[70]

The TSF shall provide octets of bits[71] that meet:

(DRG.4.6) The RNG generates output for which 2^34 strings of bit length 128 are mutually different with probability 2^-16 probability.

(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 of [AIS31][72].

6.1.2.3 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

[66] that meet the following: [SP800-90A]

[67][selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic]

[68] As defined by [AIS31]

[69][assignment: list of security capabilities]

[70] The quality metric required in FCS_RNG.1.2 is detailed in the German Scheme (see [AIS31]).

[71][selection: bits, octets of bits, numbers [assignment: format of the numbers]]

[72][assignment: a defined quality metric]

The TSF shall enforce the Key Basics SFP[73] on

1. subjects: all,
2. information: keys,
3. operations: all[74].

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[75] 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[76].

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[77].

[73] [assignment: information flow control SFP]

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

[75] [assignment: information flow control SFP]

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

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

FDP_IFF.1.3/KeyBasics

The TSF shall enforce the **following additional information flow control rules[78]:**none[79]

FDP_IFF.1.4/KeyBasics

The TSF shall explicitly authorise an information flow based on the following rules: none[80]

FDP_IFF.1.5/KeyBasics

The TSF shall explicitly deny an information flow based on the following rules:

1. No subject shall be allowed to access the plaintext value of any secret key directly.
2. No subject shall be allowed to export a secret key in plaintext.
3. No subject shall be allowed to export an Assigned Key.
4. No subject shall be allowed to export a secret key without submitting the correct authorisation data for the key.
5. No subject shall be allowed to access intermediate values in any operation that uses a secret key.
6. A key with an Export Flag value marking it as non-exportable shall not be exported[81]

Application Note 40 (Application Note 21 from [EN 419221-5]: Applied)

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.

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.

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

[78] [refinement]

[79] [assignment: additional information flow control SFP rules]

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

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

The TSF shall enforce the KeyUsage_SFP[82] on

1. subjects: all,
2. objects: keys,
3. operations: all[83].

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

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/KeyUsage

The TSF shall enforce the KeyUsage SFP[84] 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[85].

Application Note 41 (Application Note 22 from [EN 419221-5]: Applied)

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/AKeyAuth for Assigned keys and in FIA_UAU.6/GenKeyAuth for non-Assigned keys.

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

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[86].

FDP_ACF.1.3/KeyUsage

The TSF shall explicitly authorize access of subjects to objects based on the following additional

[82] [assignment: access control SFP]

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

[84] [assignment: access control SFP]

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

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

rules: none[87].

FDP_ACF.1.4/KeyUsage

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[88].

FDP_ACC.1/CM_Backup (Subset access control)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/CM_Backup

The TSF shall enforce the Backup SFP[89] on

1. subjects: all,
2. objects: keys,
3. operations: backup, restore[90].

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

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/CM_Backup

The TSF shall enforce the Backup SFP[91] to objects based on the following:

1. whether the subject is an administrator[92].

FDP_ACF.1.2/CM_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

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

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

[89] [assignment: access control SFP]

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

[91] [assignment: access control SFP]

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

binding of each set of attributes to its key[93].

FDP_ACF.1.3/CM_Backup

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[94].

FDP_ACF.1.4/CM_Backup

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[95].

FDP_SDI.2 (Stored data integrity monitoring and action)

Hierarchical to: FDP_SDI.1 Stored data integrity monitoring.

Dependencies: No dependencies.

FDP_SDI.2.1

The TSF shall monitor user data stored in containers controlled by the TSF for integrity errors[96] on all **keys (including security attributes)[97]**, based on the following attributes: integrity protection data[98].

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[99].

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 de-allocation of the resource from[100] the following objects:

1. authorisation data,
2. keys[101].

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

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

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

[96] [assignment: integrity errors]

[97] refinement: objects

[98] [assignment: user data attributes]

[99] [assignment: action to be taken]

[100] [selection: allocation of the resource to, deallocation of the resource from]

[101] [assignment: list of objects]

6.1.2.4 Identification and authentication (FIA)

FIA_UID.1/CM (Timing of identification)

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1/CM

The TSF shall allow:

1. Self test according to FPT_TST_EXT.1[102],
2. Establishing trusted paths among different TOE parts (MPCAs),
3. Establishing a trusted path between External Client Application and the TOE[103].

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

FIA_UID.1.2/CM

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

FIA_UAU.1/CM (Timing of authentication)

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1/CM

The TSF shall allow:

1. Self-test according to FPT_TST_EXT.1[104],
2. Identification of the user by means of TSF required by FIA_UID.1[105],
3. Establishing trusted paths among different TOE parts (MPCAs),
4. Establishing a trusted path between External Client Application and the TOE[106]

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

FIA_UAU.1.2/CM

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

FIA_AFL.1/CM_authentication (Authentication failure handling)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/CM_authentication

[102] [assignment: list of TSF-mediated actions]

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

[104] [assignment: list of TSF-mediated actions]

[105] [assignment: list of TSF-mediated actions]

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

The TSF shall detect when an administrator configurable positive integer within (3, 20) values[107] unsuccessful authentication attempts occur related to consecutive failed authentication attempts[108].

FIA_AFL.1.2/CM_authentication

When the defined number of unsuccessful authentication attempts has been met[109] the TSF shall block access to[110] any TSF-mediated function until[111] unblocked by Administrator[112].

FIA_AFL.1/CM_authorisation (Authentication failure handling)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/CM_authorisationThe TSF shall detect when an administrator configurable positive integer within (3, 20) values[113] unsuccessful **authorisation[114]** attempts occur related to consecutive failed authorisation attempts[115].

FIA_AFL.1.2/CM_authorisation

When the defined number of unsuccessful **authorisation[116]** attempts has been met[117] the TSF shall block access to[118] the related key until[119] unblocked by Administrator[120].

FIA_UAU.6/AKeyAuth (Re-authenticating)

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1/AKeyAuth

The TSF shall **authorise and re-authorise**[121] the user **for access to a secret key[122]** under the conditions:

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

[108][assignment: list of authentication events]

[109][selection: met, surpassed]

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

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

[112][assignment: list of actions]

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

[114][refinement: authentication]

[115][assignment: list of authentication events]

[116][refinement: authentication]

[117][selection: met, surpassed]

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

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

[120][assignment: list of actions]

[121][refinement: re-authenticate]

[122][refinement]

1. Authorisation in order to be granted initial access to the key[123]; and
2. Re-authorisation of all **Assigned[124]** keys under the following conditions:
    - after expiry of the time period (as specified in the key's attributes) for which the secret key was last authorised;
    - after the number of uses of the secret key (as specified in the key's attributes) for which the secret key was last authorised has already been made; and
    - after explicit rescinding of previous authorisation for access to the secret key[125] [126].

FIA_UAU.6/GenKeyAuth (Re-authenticating)

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1/GenKeyAuth

The TSF shall **authorise and re-authorise**[127] the user **for access to a secret key[128]** under the conditions:

1. Authorisation in order to be granted initial access to the key[129]; and
2. Re-authorisation of all **non-Assigned[130]** keys under the following conditions:
    - after expiry of an administrator configurable time period for which the secret key was last authorized (in case of this value equals to 0, there is no expiry at all);
    - after an administrator configurable number of uses of the secret key for which the secret key was last authorised has already been made; (in case of this value equals to 0, there is

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

[124] [refinement]

[125] [EN 419221-5]: [selection:

- Re-authorisation of [assignment: identification of secret keys that are subjects 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 authorized,
    - after the number of uses of the secret key (as specified in the secret key's attributes) for which the secret key was last authorised has already been made;
    - after explicit rescinding of previous authorization for access to the secret key].
- [assignment: list of other conditions under which authorisation and re-authorisation for access to secret keys is required]
- Authorisation on every subsequent access to the key].

[126] CC: [assignment: list of conditions under which re-authentication is required]

[127] [refinement: re-authenticate]

[128] [refinement]

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

[130] [refinement]

no expiry at all) [131] [132].

6.1.2.5 Security management (FMT)

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

Assigned Keys represent a type of key that can be more easily mapped to requirements for sole control because changes to some of their attributes are more tightly controlled (see FMT_MSA.1/AKeys, and the description of attributes below) and, since they are intended for use within the TOE, because they cannot be imported or exported[133]. In particular, an Administrator cannot avoid the need to provide the current authorization data in order to use such a key, nor can an Administrator change the authorization data (which would then allow use of the key by the Administrator). This enables a key to be generated and then to be made an Assigned Key at the point where it is assigned to an individual signatory or, in the case of a key used for the creation of electronic seals, to a group of key users.

FMT_SMR.1/CM (Security roles)

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FMT_SMR.1.1/CM

The TSF shall maintain the roles Administrator, Local Client Application, External Client Application, Key User[134].

FMT_SMR.1.2/CM

The TSF shall be able to associate users with roles.

Application Note 42 (Application Note 36 from [EN 419221-5]: Applied)
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

[131] [EN 419221-5]: [selection:

- Re-authorisation of [assignment: identification of secret keys that are subjects 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 authorized,
  - after the number of uses of the secret key (as specified in the secret key's attributes) for which the secret key was last authorised has already been made;
  - after explicit rescinding of previous authorization for access to the secret key].
- [assignment: list of other conditions under which authorisation and re-authorisation for access to secret keys is required]
- Authorisation on every subsequent access to the key].

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

[133] 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 1.4.2.1.

[134] CC: [assignment: the authorised identified roles], PP: [Administrator, [selection: Local Client Application, External Client Application], Key User, [assignment: list of additional authorised identified roles]]

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.

FMT_SMF.1/CM **(Security management functions)**

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1/CM

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[135],
5. key import function[136],
6. key export function[137],
7. **User management**,
8. **Configuration management**
9. **Ability to update the TOE and to verify the updates[138]**.[139]

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[140] the TSF data in the Table 6.2[141] to Administrator[142].

[135] [EN 419221-5]:

(4) [selection: backup and restore functions, no backup and restore functions]

[136] [EN 419221-5]:

(5) [selection: key import function, no key import function],.

[137] [EN 419221-5]:

(6) [selection: key export function, no key export function].

[138] [refinement]

[139][assignment: list of management functions to be provided by the TSF]

[140][selection: change default, query, modify, delete, clear, [assignment: other operations]]

[141][assignment: list of TSF data]

[142][assignment: the authorized identified roles]

TSF data	user	key
user accounts (as in FIA_UAU.1) blocked by authentication failures	Administrator Key User	
keys (as in FIA_UAU.6/AKeyAuth) blocked by authorisation failures		Assigned Key
keys (as in FIA_UAU.6/GenKeyAuth) blocked by authorisation failures		General Key
keys (as in FIA_UAU.6/AKeyAuth) blocked by re-authorisation failures		Assigned Key
keys (as in FIA_UAU.6/GenKeyAuth) blocked by re-authorisation failures		General Key

Table 6.2 TSF data related to the unblocking

FMT_MTD.1/AuditLog (Management of TSF data)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/AuditLog

The TSF shall restrict the ability to control export and deletion of[143] the audit log records[144] to the Administrator role[145].

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[146] to restrict the ability to modify[147] the security attributes Uprotected Flag, Authorisation Data and Operational Flag[148] to:

- Key User modifies his/her Uprotected Flag with (first used) chgkeypwd CMAPI command,
- Key User modifies his/her Authorisation Data with chgpwd CMAPI command,

[143] [selection: change default, query, modify, delete, clear, [assignment: other operations]]

[144] [assignment: list of TSF data]

[145] [assignment: the authorized identified roles]

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

[147] [selection: change default, query, modify, delete, [assignment: other operations]]

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

- Key User modifies his/her Operational Flag with setkeyopstate CMAPI command[149].

FMT_MSA.1/AKeys (Management of security attributes)

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/AKeys

The TSF shall enforce the Key Usage SFP[150] to restrict the ability to modify[151] the security attributes Uprotected Flag, Authorisation Data and Operational Flag[152] to:

- Key User modifies his/her Uprotected Flag with (first used) chgkeypwd CMAPI command,
- Key User modifies his/her Authorisation Data with chgpwd CMAPI command,
- Key User modifies his/her Operational Flag with setkeyopstate CMAPI command[153].

FMT_MSA.3/Keys (Static attribute initialization)

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[154] to provide restrictive[155] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Keys

The TSF shall allow Administrator[156] to specify alternative initial values to override the default values when an object or information is created.

Application Note 43

The Administrator can specify alternative initial values for the following security attributes:

1. Key Usage ("Signing" or "General")

[149] [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]]

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

[151] [selection: change default, query, modify, delete, [assignment: other operations]]

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

[153] [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]

[154] [assignment: access control SFP, information flow control SFP]

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

[156] [assignment: the authorized identified roles, according to the constraints in the Key Attributes Initialisation Table]

Key Attribute (MSA.1)	Assigned Key	General Key
Key ID	Initialised by generation process	Initialised by generation process
Owner ID	Initialised by generation process	Initialised by generation process
Key Type	Initialised by generation process	Initialised by generation process
Authorisation Data	Initialised by authenticated Key User (the owner of the key)	Initialised by authenticated Key User (the owner of the key)
Re-authorisation conditions	Initialised by generation process	Initialised by generation process
Key Usage	Initialised by creator during generation	Initialised by creator during generation
Assigned Flag	Initialised by generation process (Assigned)	Initialised by generation process (Non-assigned)
Uprotected Flag	Initialised by generation process	Initialised by generation process
Operational Flag	Initialised by generation process	Initialised by generation process
Integrity Protection Data	Initialised automatically by TSF	Initialised automatically by TSF

Table 6.3 Key Attributes Initialisation Table

Key Attribute (MSA.1)	Assigned Key	General Key
Key ID	Cannot be modified	Cannot be modified
Owner ID	Cannot be modified	Cannot be modified
Key Type	Cannot be modified	Cannot be modified
Authorisation Data	Modified only when modification operation includes successful validation of current (pre-modification) authorisation data	Modified only when modification operation includes successful validation of current (pre-modification) authorisation data
Re-authorisation conditions	Cannot be modified	Cannot be modified
Key Usage	Cannot be modified	Cannot be modified
Assigned Flag	Cannot be modified	Cannot be modified
Uprotected Flag	Modified only when the Key User establishes his/her Authorisation Data	Modified only when the Key User establishes his/her Authorisation Data
Operational Flag	Can be modified only by Key User	Can be modified only by Key User
Integrity Protection Data	Cannot be modified by users (maintained automatically by TSF)	Cannot be modified by users (maintained automatically by TSF)

Table 6.4 Key Attributes Modification Table

Application Note 44

Key ID (key identifier) uniquely identifies the key within the system of which the CM is a part.

Owner ID identifies the Key User who owns the key.

Key Type identifies whether the key is AES, 3DES, ARIA, SEED, RSA or EC key.

Authorisation data: value of data that allows a secret key to be used for cryptographic operations. The CM does not store the value of the Authorisation data, but uses it for encrypt/decrypt (share of) the key.

Re-authorisation conditions: the constraints on uses of the key that can be made before reauthorisation is required according to FIA_UAU.6/AKeyAuth for Assigned keys and FIA_UAU.6/GenKeyAuth for non-Assigned keys, and which determine whether a subject is currently authorised to use a key.

Key Usage: the cryptographic functions that are allowed to use the key 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 ST as 'exportable (meaning export is allowed) and 'non-exportable' (meaning export is not allowed)

Assigned flag indicates whether the key has currently been assigned. For an Assigned Key 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 'assigned' and 'non-assigned'.

Uprotected Flag indicates whether the stored key is protected only with an infrastructural key, or additionally with a password established by the key's owner. This flag is initialised by key generation process, setting its value to "no". When the Key User (key's owner) establishes his/her Authorisation Data, the value of this flag is set to "yes".

Operational Flag indicates whether the key is in operational state. This flag is initialised by key generation process to "non-operational". A key can be used for cryptographic operations only in "operational" state. Only the Key User (key's owner) is able to change the value of this flag from "non-operational" to "operational" and vice versa.

Integrity Protection Data is a digital signature created by an infrastructural key for key data record which contains the key and its attributes.

6.1.2.6 Protection of the TSF (FPT)

FPT_STM.1/CM (Reliable time stamps)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1/CM

The TSF shall be able to provide reliable time stamps.

FPT_TST_EXT.1 (Basic TSF Self Testing)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests during initial start-up (or power-on), periodically during normal operation, at the request of the authorised user, and at the conditions specified below[157] [158] to demonstrate the correct operation of the TSF:

- At initial start-up (or power-on):
  - Software/firmware integrity tests
  - Cryptographic algorithm tests (known answer tests)
  - Random number generator tests[159]
  - RSA pair-wise consistency tests for infrastructural keys
  - Checking the environmental resources (e.g. available storage capacity, network)
  - Configuration file integrity test
  - Checking the database consistency among different TOE parts (in case of distributed configuration)
  - Checking the expiration date of stored certificates
- Periodically during normal operation (when frequency of the test depends on an administrator configurable value):
  - RSA pair-wise consistency tests for infrastructural keys
  - Checking whether the environmental conditions are outside normal operating range (including temperature and power)
  - Checking the database consistency among different TOE parts (in case of distributed configuration)
- At the condition:
  - pair-wise consistency tests for signer keys (during the asymmetric key pair generation),
  - Random number generator tests (in every 10 day)
  - Checking the environmental resources (e.g. available storage capacity, network) (in every hour)
  - health checks for random number generators (after every 2^20 generate operations)
  - Examining the state of the CM for a potential tamper event
  - Database records integrity tests (during every read operation)
  - Checking the expiration date of stored certificates (in every hour)[160].

FPT_PHP.1 (Passive detection of physical attack)

Hierarchical to: No other components.

[157] [EN 419221-5] [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]]

[158] ST: [assignment: conditions under which self-tests should occur]

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

[160][assignment: list of additional self-tests run by the TSF]

Dependencies: No dependencies.

FPT_PHP.1.1

The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.1.2

The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

FPT_PHP.3 (Resistance to physical attack)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1

The TSF shall resist removing the cover[161] to the MPCA[162] by responding automatically such that the SFRs are always enforced.

Application Note 45 (Application Notes 33 and 34 from [EN 419221-5]: Applied)

The level of protection in FPT_PHP.1 and FPT_PHP.3 is equivalent to the level of assessment for this aspect of tamper detection and response required for ISO/IEC 19790:2012 for Security Level 3.

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 the RNG occur,
4. Corruption of TOE software occurs[163],
5. Integrity error in blocks of audit records occurs,
6. Database inconsistency occurs[164].

6.1.2.7 Trusted path/channels (FTP)

FTP_TRP.1/Local (Trusted Path)

Hierarchical to: No other components.

[161] [assignment: physical tampering scenarios]

[162] [assignment: list of TSF devices/elements]

[163] [assignment: list of types of failures in the TSF]

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

Dependencies: No dependencies.

FTP_TRP.1.1/Local

The TSF shall provide a communication path between itself and local[165] **client applications[166]** that is logically distinct from other communication paths and provides assured **authentication**[167] of its end points and protection of the communicated data from modification and disclosure[168].

FTP_TRP.1.2/Local

The TSF shall permit local client applications[169] to initiate communication via the trusted path.

FTP_TRP.1.3/Local

The TSF shall require the use of the trusted path for: all CMAPI commands[170].

Application Note 46 (Application Note 29 from [EN 419221-5]: Applied)

Since in the Trident CM local client applications are located within the physical boundary of the same hardware appliance, the trusted path may be mapped to the physical configuration. Consequently, this SFR is trivially satisfied because of the physical security assumed in the appliance environment.

FTP_TRP.1/Admin (Trusted Path)

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1/Admin

The TSF shall provide a communication path between itself and local[171] **Administrator through a trusted IT product**[172] that is logically distinct from other communication paths and provides assured **authentication**[173] of its end points and protection of the communicated data from modification and disclosure[174].

FTP_TRP.1.2/Admin

The TSF shall permit local[175] **Administrator through a trusted IT product**[176] to initiate communication via the trusted path.

[165] [selection: remote, local]

[166] users

[167] identification

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

[169] [selection: the TSF, local users, remote users]

[170] [assignment: services for which trusted path is required].

[171] [selection: remote, local]

[172] [refinement: users]

[173] [refinement: identification]

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

[175] [selection: the TSF, local users, remote users]

[176] [refinement: users]

FTP_TRP.1.3/Admin

The TSF shall require the use of the trusted path for:

1. User management,
2. Configuration management[177].

FTP_TRP.1/External (Trusted Path)

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1/External

The TSF shall provide a communication path between itself and remote[178] **external client applications**[179] that is logically distinct from other communication paths and provides assured **authentication**[180] of its end points and protection of the communicated data from modification and disclosure[181].

FTP_TRP.1.2/External

The TSF shall permit remote[182] **external client applications**[183] to initiate communication via the trusted path.

FTP_TRP.1.3/External

The TSF shall require the use of the trusted path for: all CMAPI commands[184].

6.1.3 SFRs of the Signature Activation Module (SAM)

The following 3 tables describe the subjects, object and operations supported by the SAM.

Subject	Description
R.Signer	Represents within the TOE, the end user that wants to create a digital signature
R.Privileged_User	Represents within the TOE, a privileged user that can administer the TOE and a few operations relevant for R.Signer

Table 6.5 Subjects of the SAM

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

[178] [selection: remote, local]

[179] [refinement: users]

[180] [refinement: identification]

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

[182] [selection: the TSF, local users, remote users]

[183] [refinement: users]

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

Object	Description
R.Reference_Privileged_User_Authentication_Data	Data used by the TOE to authenticate a Privileged_User
R.Reference_Signer_Authentication_Data	Data used by the TOE to authenticate a Signer
R.SVD	The public part of a R.Signer signature key pair
R.Signing_Key_Id	An identifier representing the private part of a R.Signer signature key pair
R.DTBS/R	Data to be signed representation
R.Authorisation_Data	Data used by the Cryptographic Module to activate the private part of a R.Signaer signature key pair
R.Signature	The result of a signature operation
R.TSF_DATA	TOE Configuration Data

Table 6.6 Objects of the SAM

Subject	Operation	Object	Description
R.Privileged_User	Create_New_Privileged_User	R.Privileged_User R.Reference_Privileged_User_Authentication_Data	A new privileged user can be created which covers the object representing the new privileged user as well as the object used to authenticate the newly created privileged user.
R.Privileged_User	Create_New_Signer	R.Signer R.Reference_Signer_Authentication_Data	A new signer can be created which covers the object representing the new signer as well as the object used to authenticate the newly created signer.
R.Privileged_User R.Signer	Generate_Signer_Key_Pair	R.Signer R.SVD R.Signing_Key_Id	A key pair can be generated and assigned to a signer.
R.Privileged User R.Signer	Signer_Maintenance	R.Signer R.SVD R.Signing_Key_Id	A key pair can be deleted from a signer.
R.Privileged User	Supply_DTBS/R	R.Signer R.DTBS/R	Data to be signed by a signer can be supplied by a privileged user.
R.Signer	Signing	R.Authorisation_Data R.Signer R.Signing_Key_Id R.DTBS/R R.Signature	A signer can sign data to be signed resulting in a signature.
R.Privileged User	TOE_Maintenance	R.TSF_DATA	The TOE configuration can be maintained by a privileged user.

Table 6.7 Operations supported by the SAM

6.1.3.1 Security audit data generation (FAU)

FAU_GEN.1/SAM (Audit data generation)

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1/SAM

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[185] level of audit; and

c) Privileged User management;

d) Privileged User authentication

e) Signer management;

f) Signer authentication **(directly or partly directly by the SAM)**[186];

g) Signing key generation;

h) Signing key destruction;

i) Signing key activation and usage including the hash of the DTBS and R.Signature;

j) Change of **SAM**[187] configuration[188];

k) Certification request generation;

l) Failures to establish secure channels between different TOE parts (MPCAs);

m) Backup and restore (FDP_ACF.1/SAM Backup): use of any backup function, use of any restore function, unsuccessful restore because of detection of modification of the backup data[189].

FAU_GEN.1.2/SAM

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 ST: type of action performed (success or failure), identity of the role which performs the operation[190], identifier of the related MPCA, human readable descriptive string about the related event[191].

Application Note 47

Audit trail does not include any data which allow to retrieve sensitive data like R.SAD, R.Reference_Signer_Authentication_Data and R.Authorisation_Data.

FAU_GEN.2/SAM (User identity association)

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

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

[186] [refinement]

[187] [refinement: TOE]

[188] [assignment: other specifically defined auditable events]

[189] [assignment: other specifically defined auditable events]

[190] CC: [assignment: other audit relevant information]

[191] [EN 419241-2]: [assignment: other audit relevant information]

FAU_GEN.2.1/SAM

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.

6.1.3.2 Cryptographic support (FCS)

FCS_CKM.1/invoke_CM:key gen_Non-multiparty (Cryptographic key generation)

See: FCS_CKM.1/key gen_Non-multiparty

FCS_CKM.1/invoke_CM:key gen_Multiparty (Cryptographic key generation)

See: FCS_CKM.1/key gen_Multiparty

Application Note 48

Although the SAM does not generate the above keys and key pairs itself, the SFRs above expresses the requirement for SAM to invoke the CM with the appropriate parameters whenever key generation is required.

FCS_CKM.4/SAM (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/SAM

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method zeroization[192] that meets the following: [FIPS 140-3], and [ISO19790], section 7.9.7[193].

Application Note 49

Although the SAM does not destruct keys itself (besides the shared secret used for TOTP validation), this SFR expresses the requirement for SAM to invoke the CM with the appropriate parameters whenever key destruction is required.

FCS_COP.1/invoke_CM:dig sig_Non_multiparty (Cryptographic operation)

See: FCS_COP.1/dig sig_Non_multiparty

FCS_COP.1/invoke_CM:dig sig_Multiparty (Cryptographic operation)

See: FCS_COP.1/dig sig_Multiparty

Application Note 50

Although the SAM does not create (or validate) digital signature (or seal) itself, the SFR above expresses the requirement for SAM to invoke the CM with the appropriate parameters whenever creation (or validation) of a digital signature (or a seal) is required.

[192] [assignment: cryptographic key destruction method]

[193] [assignment: list of standards]

FCS_COP.1/SAM_hash (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/SAM_hash

The TSF shall perform cryptographic hash function[194] in accordance with a specified cryptographic algorithm SHA-256, SHA-384 and SHA-512[195] and cryptographic key sizes none[196] that meet the following: [TS 119312] and [FIPS 180-4][197].

FCS_COP.1/SAM_key_derivation (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/SAM_key_derivation

The TSF shall perform key derivation[198] in accordance with a specified cryptographic algorithm PBKDF2[199] and cryptographic key sizes length of password[200] that meet the following: [PKCS#5][201].

FCS_COP.1/SAM_TOTP_verification (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]

[194] [assignment: list of cryptographic operations]

[195] [assignment: cryptographic algorithm]

[196] [assignment: cryptographic key sizes]

[197] [assignment: list of standards]

[198] [assignment: list of cryptographic operations]

[199] [assignment: cryptographic algorithm]

[200] [assignment: cryptographic key sizes]

[201] [assignment: list of standards]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SAM_TOTP_verification

The TSF shall perform TOTP verification[202] in accordance with a specified cryptographic algorithm HOTP[203] and cryptographic key sizes 256 bits[204] that meet the following: [RFC 4226] and [RFC 6238][205].

Application Note 51

The SAM performs TOTP verification itself, (for the Signer's possession-based authentication).

Application Note 52

Since the SAM is implemented as a local application within the same physical boundary as the CM, SFR FCS_RNG.1 does not apply for the SAM (see Application Note 39 in [EN 419241-2]).

6.1.3.3 User data protection (FDP)

FDP_ACC.1/Privileged User Creation (Subset access control)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Privileged User Creation

The TSF shall enforce the Privileged User Creation SFP[206] on

1. subjects: Privileged User,
2. objects: new security attributes for the Privileged User to be created,
3. operations: Create_New_Privileged_User:

The **SAM**[207] creates R.Privileged_User and R.Reference_Privileged_User_Authentication_Data with information transmitted by Privileged User[208].

Application Note 53

The initial Privileged User is created with a special command (mpc_initmpcm), which requires a master password, defined during installation phase. Later all Privileged User are able to create a new Privileged User.

FDP_ACF.1/Privileged User Creation (Security attribute based access control)

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

[202] [assignment: list of cryptographic operations]

[203] [assignment: cryptographic algorithm]

[204] [assignment: cryptographic key sizes]

[205] [assignment: list of standards]

[206] [assignment: access control SFP]

[207] [refinement: TOE]

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

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/Privileged User Creation

The TSF shall enforce the Privileged User Creation SFP[209] to objects based on the following:

1. Whether the subject is a Privileged User authorized to create a new Privileged User[210].

FDP_ACF.1.2/Privileged User Creation

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

1. Only a Privileged User who has been authorised for creation of new users can carry out the Create_New_Privileged_User operation[211].

FDP_ACF.1.3/Privileged User Creation

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[212].

FDP_ACF.1.4/Privileged User Creation

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[213].

FDP_ACC.1/Signer Creation **(Subset access control)**

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Signer Creation

The TSF shall enforce the Signer Creation SFP[214] on

1. subjects: Privileged User,
2. objects: new security attributes for the Signer to be created,
3. operations: Create_New_Signer:

   The **SAM**[215] creates R.Signer and R.Reference_Signer_Authentication_Data with information transmitted by Privileged User[216].

FDP_ACF.1/Signer Creation **(Security attribute based access control)**

Hierarchical to: No other components.

[209] [assignment: access control SFP]

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

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

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

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

[214] [assignment: access control SFP]

[215] [refinement: TOE]

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

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/Signer Creation

The TSF shall enforce the Signer Creation SFP[217] to objects based on the following:

1. Whether the subject is a Privileged User authorized to create a new Signer[218].

FDP_ACF.1.2/Signer Creation

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

1. Only a Privileged User who has been authorised for creation of new users can carry out the Create_New_Signer operation[219].

FDP_ACF.1.3/Signer Creation

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[220].

FDP_ACF.1.4/Signer Creation

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[221].

FDP_ACC.1/Signer Maintenance (Subset access control)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Signer Maintenance

The TSF shall enforce the Signer Maintenance SFP[222] on

1. subjects: Privileged User, and Signer
2. objects: The security attributes R.Reference_Signer_Authentication_Data of R.Signer,
3. operations: Signer Maintenance:

The Privileged User or Signer instructs the **SAM**[223] to update R.Reference_Signer_Authentication_Data of R.Signer [224].

[217] [assignment: access control SFP]

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

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

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

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

[222] [assignment: access control SFP]

[223] [refinement: TOE]

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

FDP_ACF.1/Signer Maintenance (Security attribute based access control)

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/Signer Maintenance

The TSF shall enforce the Signer Maintenance SFP [225] to objects based on the following:

1. Whether the subject is a Privileged User or Signer authorised to maintain the Signer security attributes [226].

FDP_ACF.1.2/Signer Maintenance

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

1. Only a Privileged User or Signer who has been authorised to maintain a Signer can carry out the Signer_Maintenance operation [227].

FDP_ACF.1.3/Signer Maintenance

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

1. The Signer must be the owner of the R.Signer object to be maintained. [228].

FDP_ACF.1.4/Signer Maintenance

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

(1) If the Signer does not own the R.Signer object, it can't be maintained[229].

Application Note 54

The initial R.Reference_Signer_Authentication_Data is created by Privileged User during the Create_New_Signer operation.

Later only Signer is able to modify his own R.Reference_Signer_Authentication_Data.

FDP_ACC.1/Signer Key Pair Generation (Subset access control)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Signer Key Pair Generation

[225] [assignment: access control SFP]

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

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

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

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

The TSF shall enforce the Signer Key Pair Generation SFP[230] on

1. subjects: Privileged User and Signer,
2. objects: the security attributes R.SVD and R.Signing_Key_Id as part of R.Signer,
3. operations: Generate_Signer_Key_Pair:

   The Privileged User or Signer instructs the **SAM**[231] to request the CM to generate a signing key pair R.Signing_Key_Id and R.SVD and assign them to the R.Signer[232].

Application Note 55

The R.Authorisation_Data is created by the key owner Signer.

The signing keys (precisely the private part of the Signer Key Pair) can be stored

a) in the TOE, or

b) externally to the TOE in a special blob (xkbl, extended key blob). In this case the signing keys are encrypted using AES-256 in GCM method, and encryption keys are not leaving the TOE. Therefore, both the integrity and confidentiality of the signer keys are protected during external storage.

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

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/Signer Key Pair Generation

The TSF shall enforce the Signer Key Pair Generation SFP[233] to objects based on the following:

1. whether the subject is a Privileged User or Signer authorised to generate a key pair[234].

FDP_ACF.1.2/Signer Key Pair Generation

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

1. Only a Privileged User or Signer who has been authorised to generate the key pair can carry out the Generate_Signer_Key_Pair operation[235].

FDP_ACF.1.3/Signer Key Pair Generation

[230] [assignment: access control SFP]

[231] [refinement: TOE]

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

[233] [assignment: access control SFP]

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

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

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

1. The Signer must be the owner of the R.Signer object where the key pair is to be generated[236].

FDP_ACF.1.4/Signer Key Pair Generation

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

1. If the Signer does not own the R.Signer object, key pair shall not be generated[237].

FDP_ACC.1/Signer Key Pair Deletion (Subset access control)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ Signer Key Pair Deletion

The TSF shall enforce the Signer Key Pair Deletion SFP[238] on

1. subjects: Privileged User and Signer,
2. objects: the security attributes R.Signing_Key_Id and R.SVD of R.Signer,
3. operations: Signer_Key Pair Deletion:

   The Privileged User or Signer instructs the **SAM**[239] to delete the R.Signing_Key_Id and R.SVD from R.Signer[240].

Application Note 56

Deletion of R.Signing_Key_Id also requires that the signing key is deleted by the CM. This SFR is limited to covering deletion of the R.Signing_Key_Id and R.SVD of R.Signer performed using one of the interfaces provided by the TOE (SAM).

FDP_ACF.1/Signer Key Pair Deletion (Security attribute based access control)

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/Signer Key Pair Deletion

The TSF shall enforce the Signer Key Pair DeletionSFP[241] to objects based on the following:

1. Whether the subject is a Privileged User or Signer authorised to delete the Signer security

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

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

[238] [assignment: access control SFP]

[239] [refinement: TOE]

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

[241] [assignment: access control SFP]

attributes[242].

FDP_ACF.1.2/Signer Key Pair Deletion

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

1. Only a Privileged User or Signer who has been authorised to delete a key pair can carry out the Signer_Key Pair Deletion operation[243].

FDP_ACF.1.3/Signer Key Pair Deletion

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

1. The Signer must be the owner of the R.Signer object containing the key pair to be deleted[244].

FDP_ACF.1.4/Signer Key Pair Deletion

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

1. If the Signer does not own the R.Signer object, the key pair can't be deleted[245].

FDP_ACC.1/Supply DTBS/R **(Subset access control)**

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Supply DTBS/R

The TSF shall enforce the Supply DTBS/R policy[246] on

1. subjects: Privileged User,
2. objects: the security attributes R.DTBS/R of R.Signer,
3. operations: Supply_DTBS/R:

   The Privileged User instructs the **SAM**[247]. to link the supplied DTBS/R to the next signature operation for R.Signer[248].

FDP_ACF.1/Supply DTBS/R **(Security attribute based access control)**

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

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

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

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

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

[246] [assignment: access control SFP]

[247] [refinement: TOE]

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

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/Supply DTBS/R

The TSF shall enforce the Supply DTBS/R policy[249] to objects based on the following:

1. Whether the subject is a Privileged User authorised to supply a DTBS/R[250].

FDP_ACF.1.2/Supply DTBS/R

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

1. Only a Privileged User or Signer who has been authorised to supply a DTBS/R can carry out the Supply_DTBS/R operation[251].

FDP_ACF.1.3/Supply DTBS/R

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[252].

FDP_ACF.1.4/Supply DTBS/R

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[253].

FDP_ACC.1/Signing **(Subset access control)**

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Signing

The TSF shall enforce the Signing policy[254] on

1. subjects: Signer,
2. objects: R.Authorisation_Data, security attributes R.Signing_Key_Id and R.DTBS/R of R.Signer and R.Signature.,
3. operations: Signing:

   The Signer instructs the **SAM**[255] to perform a signature operation containing the following steps:

   - The **SAM**[256] establish R.Authorisation_Data for the R.Signing_Key_Id.

[249] [assignment: access control SFP]

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

[251] [rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

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

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

[254] [assignment: access control SFP]

[255] [refinement: TOE]

[256] [refinement: TOE]

- The **SAM**[257] uses the R.Autorisation_Data and R.Signing_Key_Id to activate a signing key in the CM and signs the R.DTBS/R resulting in R.Signature.
- The **SAM**[258] deactivates the signing key when the signature operation is completed.[259]

Application Note 57 (Application Note 53 from [EN 419241-2]: Applied)

Signing key deactivating means that the signer shall authorise any subsequent use of it.

Application Note 58

[Trident-ARC] and [Trident-TDS] describe how R.Authorisation_Data is used to activate signing keys in the CM and how the DTBS/R(s) is supplied to the SAM.

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

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/Signing

The TSF shall enforce the Signing policy[260] to objects based on the following:

1. Whether the subject is a Signer authorised to create a signature[261].

FDP_ACF.1.2/Signing

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

1. The R.SAD is verified in integrity.
2. The R.SAD is verified that it binds together the Signer authentication, a set of R.DTBS/R and R.Signing_Key_Id.
3. The R.DTBS/R used for signature operations is bound to the R.SAD.
4. The Signer identified in the SAD is authenticated according to the rules specified in FIA_UAU.5/Signer.
5. Only an R.Signing_Key_Id as bound in the SAD, and which is part of the R.Signer security attributes, can be used to create a signature[262].

FDP_ACF.1.3/Signing

The TSF shall explicitly authorize access of subjects to objects based on the following additional

[257] [refinement: TOE]

[258] [refinement: TOE]

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

[260] [assignment: access control SFP]

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

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

rules:

1. The Signer must be the owner of the R.Signer object used to generate the signature[263].

FDP_ACF.1.4/Signing

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

1. If the Signer does not own the R.Signer object, it can’t be used to create a signature[264].

FDP_ACC.1/SAM Maintenance (Subset access control)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/SAM Maintenance

The TSF shall enforce the **SAM**[265] Maintenance SFP[266] on

1. subjects: Privileged User,
2. objects: R.TSF_DATA,
3. operations: SAM_Maintenance:

   The Privileged User transmits information to the **SAM**[267] to manage R.TSF_DATA[268].

FDP_ACF.1/SAM Maintenance (Security attribute based access control)

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/SAM Maintenance

The TSF shall enforce the **SAM**[269] Maintenance SFP[270] to objects based on the following:

1. Whether the subject is a Privileged User authorised to maintain the **SAM**[271] configuration data.[272].

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

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

[265] [refinement: TOE]

[266] [assignment: access control SFP]

[267] [refinement: TOE]

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

[269] [refinement: TOE]

[270] [assignment: access control SFP]

[271] [refinement: TOE]

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

FDP_ACF.1.2/SAM Maintenance

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

1. Only a Privileged User who has been authorised to maintain the **SAM**[273] can carry out the **SAM**[274] Maintenance operation[275].

FDP_ACF.1.3/SAM Maintenance

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[276].

FDP_ACF.1.4/SAM Maintenance

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[277].

FDP_ACC.1/SAM_Backup (Subset access control)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/SAM_Backup

The TSF shall enforce the Backup SFP[278] on

1. subjects: all,
2. objects: keys,
3. operations: backup, restore[279].

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

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/SAM_Backup

The TSF shall enforce the Backup SFP[280] to objects based on the following:

[273] [refinement: TOE]

[274] [refinement: TOE]

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

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

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

[278] [assignment: access control SFP]

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

[280] [assignment: access control SFP]

1. whether the subject is a Privileged User[281].

FDP_ACF.1.2/SAM_Backup

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

1. Only authorised Privileged Users 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 a Privileged User,
3. Any backup and restore shall preserve the confidentiality and integrity of user's security attributes[282].

FDP_ACF.1.3/SAM_Backup

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[283].

FDP_ACF.1.4/SAM_Backup

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[284].

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

Hierarchical to: No other components.

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

FDP_ETC.2.1/Signer

The TSF shall enforce the Signer Creation SFP, Signer Key Pair Generation SFP, Signer Key Pair Deletion SFP, Signer Maintenance SFP, Supply DTBS/R SFP, Signing SFP[285] and **Backup SFP**[286] [287] when exporting user data, controlled under the SFP(s), outside of the TSF.

FDP_ETC.2.2/Signer

The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3/Signer

The TSF shall ensure that the security attributes, when exported outside the TSF, are unambiguously

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

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

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

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

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

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

[287] [refinement]

associated with the exported user data.

FDP_ETC.2.4/Signer

The TSF shall enforce the following rules when user data is exported from the TSF: none[288].

FDP_IFC.1/Signer **(Subset information flow control)**

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/Signer

The TSF shall enforce the Signer Flow SFP[289] on Privileged User and Signer accessing Signer security attributes for all operations[290].

FDP_IFF.1/Signer **(Simple security attributes)**

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/Signer

The TSF shall enforce the Signer Flow SFP[291] based on the following types of subject and information security attributes:

1. Privileged User and Signer accessing the Signer security attributes[292].

FDP_IFF.1.2/Signer

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

1. The **SAM**[293] shall be initialized with FDP_ACC.1/SAM Maintenance,
2. To allow a Signer to sign, the Signer shall be created in the **SAM**[294] by FDP_ACC.1/Signer Creation followed by FDP_ACC.1/Signer key Pair Generation,
3. After Signer is created the following operations can be done: FDP_ACC.1/Signer Key Pair Generation, FDP_ACC.1/Signer Key Pair Deletion, FDP_ACC.1/Supply DTBS/R, FDP_ACC.1/Signer Maintenance, FDP_ACC.1/Signing[295] and **FDP_ACC.1/**

[288][assignment: additional exportation control rules]

[289] [assignment: information flow control SFP]

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

[291] [assignment: information flow control SFP]

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

[293] [refinement: TOE]

[294] [refinement: TOE]

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

SAM_Backup[296] [297].

FDP_IFF.1.3/Signer

The TSF shall enforce the **following additional information flow control rules[298]:** none[299]

FDP_IFF.1.4/Signer

The TSF shall explicitly authorise an information flow based on the following rules: none[300]

FDP_IFF.1.5/Signer

The TSF shall explicitly deny an information flow based on the following rules: none[301].

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

Hierarchical to: No other components.

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

FDP_ETC.2.1/Privileged User

The TSF shall enforce the Privileged User Creation policy[302] when exporting user data, controlled under the SFP(s), outside of the TSF.

FDP_ETC.2.2/Privileged User

The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3/Privileged User

The TSF shall ensure that the security attributes, when exported outside the TSF, are unambiguously associated with the exported user data.

FDP_ETC.2.4/Privileged User

The TSF shall enforce the following rules when user data is exported from the TSF: none[303].

FDP_IFC.1/Privileged User (Subset information flow control)

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/Privileged User

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

[297] [refinement]

[298] [refinement]

[299] [assignment: additional information flow control SFP rules]

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

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

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

[303] [assignment: additional exportation control rules]

The TSF shall enforce the Privileged User Flow SFP[304] on Privileged User,

1. information: Privileged User accessing Privileged User security attributes for all operations[305].

FDP_IFF.1/Privileged User (Simple security attributes)

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/Privileged User

The TSF shall enforce the Privileged User Flow SFP[306] based on the following types of subject and information security attributes:

1. Privileged User accessing the Privileged User security attributes[307].

FDP_IFF.1.2/Privileged User

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

1. The **SAM**[308] shall be initialized with FDP_ACC.1/SAM Maintenance[309].

FDP_IFF.1.3/Privileged User

The TSF shall enforce the**:** none[310]

FDP_IFF.1.4/Privileged User

The TSF shall explicitly authorise an information flow based on the following rules: none[311]

FDP_IFF.1.5/Privileged User

The TSF shall explicitly deny an information flow based on the following rules: none[312].

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

Hierarchical to: No other components.

[304] [assignment: information flow control SFP]

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

[306] [assignment: information flow control SFP]

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

[308] [refinement: TOE]

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

[310] [assignment: additional information flow control SFP rules]

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

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

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

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

FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1/Signer

The TSF shall enforce the Signer Creation SFP, Signer Key Pair Generation SFP, Signer Key Pair Deletion SFP, Signer Maintenance SFP, Supply DTBS/R SFP, Signing SFP[313] and **SAM_Backup SFP**[314] when importing user data, controlled under the SFP(s), outside of the TOE.

FDP_ITC.2.2/Signer

The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/Signer

The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4/Signer

The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/Signer

The TSF shall enforce the following rules when user data is imported from the TSF: none[315].

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

Hierarchical to: No other components.

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

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

FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1/Privileged User

The TSF shall enforce the Privileged User Creation policy[316] when importing user data, controlled under the SFP(s), outside of the TOE.

FDP_ITC.2.2/Privileged User

The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/Privileged User

The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4/Privileged User

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

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

[315] [assignment: additional importation control rules]

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

The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/Privileged User

The TSF shall enforce the following rules when user data is imported from the TSF: none[317].

FDP_UCT.1 (Basic data exchange confidentiality)

Hierarchical to: No other components.

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

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

FDP_UCT.1.1

The TSF shall enforce the Signer Flow SFP and Privileged User Flow SFP[318] to be able to transmit and receive[319] user data in a manner protected from unauthorised disclosure.

FDP_UIT.1 (Data exchange integrity)

Hierarchical to: No other components.

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

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

FDP_UIT.1.1

The TSF shall enforce the Signer Flow SFP and Privileged User Flow SFP[320] to transmit and receive[321] user data in a manner protected from modification and insertion[322] errors for R.Signer and R.Privileged_User and for R.SAD also[323] from modification and replay errors[324].

FDP_UIT.1.2

The TSF shall be able to determine on receipt of user data, whether modification, deletion and insertion[325] for R.Signer and R.Privileged_User and for R.SAD[326] whether modification and replay[327] has occurred.

[317] [assignment: additional importation control rules]

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

[319] [selection: transmit, receive]

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

[321] [selection: transmit, receive]

[322] [selection: modification, deletion, insertion, replay]

[323] The TSF shall enforce the [assignment: access control SFP(s) and/or information flow control SFP(s)] to [selection: transmit, receive] user data in a manner protected from [selection: modification, deletion, insertion, replay] errors.

[324] [selection: modification, deletion, insertion, replay]

[325] [selection: modification, deletion, insertion, replay]

[326] The TSF shall be able to determine on receipt of user data, whether [selection: modification, deletion, insertion, replay] has occurred.

[327] [selection: modification, deletion, insertion, replay]

Application Note 59 (Application Note 59 from [EN 419241-2]: Applied)

Insertion of objects would mean that authorised creation of Signer and Privileged User could be possible.

6.1.3.4 Identification and authentication (FIA)

FIA_UID.2/SAM (User identification before any action)

Hierarchical to: FIA_UID.1 Timing of identification.

Dependencies: No dependencies.

FIA_UID.2.1/SAM

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

FIA_UAU.1/SAM (Timing of authentication)

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1/SAM

The TSF shall allow:

1. Identification of the Privileged User by means of TSF required by FIA_UID.2
2. Establishing a trusted path between remote Signer and the TOE by means of TSF required by FTP_TRP.1[328]

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

FIA_UAU.1.2/SAM

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

FIA_AFL.1/SAM (Authentication failure handling)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/SAM

The TSF shall detect when a **TOE Maintenance**[329] configurable positive integer within (3,20) values[330] unsuccessful authentication occurs related to Privileged User and Signer authentication[331].

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

[329] [refinement: an administrator]

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

[331] [assignment: list of authentication events]

FIA_AFL.1.2/SAM

When the defined number of unsuccessful authentication attempts has been met[332], the TSF shall suspend the Privileged User and when it is a Signer, suspend the usage of R.Signing_Key_Id[333].

FIA_UAU.5/Signer (Multiple authentication mechanisms)

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1/Signer

The TSF shall provide a password based authentication and a second authentication, based on Time-Based One-Time Password Algorithm according to [RFC 6238][334] to support user authentication.

FIA_UAU.5.2/Signer

The TSF shall authenticate any **Signer[335]**'s claimed identity according to the **following[336]:**

- If the signer authentication is carried out directly by the SAM:
  - Signer provides his/her password (as the knowledge-based authentication factor) and the TOTP (as the possession-based authentication factor)[337].
- If the signer authentication is carried out indirectly by the SAM:
  - Delegated party provides a JsonWebToken (JWT) according to [RFC 7519] as an assertion that the Signer has been authenticated.
- If the signer authentication is carried out partly indirectly by the SAM:
  - Signer provides his/her password, and delegated party provides a JsonWebToken (JWT) according to [RFC 7519] as an assertion that the Signer has been authenticated.

Application Note 60 (Application Note 62 from [EN 419241-2]: Applied)

This SFR only apply for Signer authentication for maintaining signer (FDP_ACC.1/Signer Maintenance) and for signing (FDP_ACC.1/Signing).

Application Note 61

The Trident supports delegated authentication, when a delegated party verifies one or two of the signer’s authentication factor.

[332] [selection: met, surpassed]

[333] [assignment: list of actions]

[334] CC: [assignment: list of multiple authentication mechanisms], PP: [selection: [assignment: list of direct authentication mechanisms conformant to [EN 419 241-1] SRA_SAP.1.1, [assignment: list of delegated authentication mechanisms conformant to [EN 419 241-1] SRA_SAP.1.1]]

[335] [refinement: user]

[336] [refinement]

[337] CC: [assignment: rules describing how the multiple authentication mechanisms provide authentication], PP: • [assignment: If the TOE supports delegated authentication then: the rules describing how this is verified by TSF], • [assignment: If the TOE is supports direct authentication of the Signer, rules describing how the direct authentication mechanisms provide authentication].

FIA_UAU.5/Privileged user **(Multiple authentication mechanisms)**

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1/Privileged User

The TSF shall provide a password based authentication and a second authentication, based on Time-Based One-Time Password Algorithm according to [RFC 6238][338] to support **Privileged user[339]** authentication.

FIA_UAU.5.2/Privileged User

The TSF shall authenticate any **Privileged User**[340]'s claimed identity according to the **following**[341]**:**

- Privileged User provides his/her password (as the knowledge-based authentication factor),
- Privileged User provides the TOTP (as the possession-based authentication factor)[342].

FIA_ATD.1 **(User attribute definition)**

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: the security attribute as defined in FIA_USB.1[343].

FIA_USB.1 **(User-subject binding)**

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition.

FIA_USB.1.1

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

1. R.Reference_Signer_Authentication_Data
2. R.Signing_Key_Id
3. R.SVD
4. R.Signer

[338] [assignment: list of multiple authentication mechanisms]

[339] [refinement: user]

[340] [refinement: user]

[341] [refinement]

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

[343] [assignment: list of security attributes]

5. Role
6. EntityType

to Signer

1. R.Reference_Privileged_User_Authentication_Data
2. R.Privileged_User
3. Role

to Privileged User.[344].

FIA_USB.1.2

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

1. Whether the subject is a Privileged User authorized to create a new Signer.
2. Whether the subject is a Privileged User authorized to create a new Privileged User
3. none[345].

FIA_USB.1.3

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

1. Whether the subject is a Privileged User authorized to modify an R.Signer object.
2. Whether the subject is a Signer authorized to modify his own R.Signer object,
3. none.[346]

Application Note 62 (Application Note 63 from [EN 419241-2]: Applied)

In FIA_USB.1.1 several attributes including R.Signing_Key_ID and R.SVD may initially be empty.

6.1.3.5 Security management (FMT)

FMT_MSA.1/Signer (Management of security attributes)

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/Signer

The TSF shall enforce:

1. Signer Creation SFP[347] to restrict the ability to create[348] the security attributes listed in

[344] [assignment: list of user security attributes]

[345] [assignment: rules for the initial association of attributes]

[346] [assignment: rules for the changing of attributes]

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

[348] [selection: change default, query, modify, delete, [assignment: other operations]]

FIA_USB.1 for Signer[349] to authorised Privileged User[350].

2. Generate Signer Key Pair SFP[351] to restrict the ability to generate[352] the security attributes R.SVD and R.Signing_Key_Id[353] to authorised Privileged User and Signer[354].
3. Signer Key Pair Deletion SFP[355] to restrict the ability to destruct[356] the security attributes R.SVD and R.Signing_Key_Id[357] as part of R.Signer to authorised Signer[358].
4. Supply DTBS/R SFP[359] to restrict the ability to create[360] the security attribute R.DTBS/R as part of R.Signer[361] to Privileged User[362].
5. Signing SFP[363] to restrict the ability to create[364] the security attribute R.DTBS/R as part of R.Signer[365] to authorised Signer[366].
6. Signing SFP[367] to restrict the ability to query[368] the security attributes listed in FIA_USB.1[369] to authorised Signer[370].
7. Signer Maintenance SFP[371] to restrict the ability to change[372] the security attributes R.Reference_Signer_Authentication_Data as part of R.Signer[373] to authorised Privileged

[349] [assignment: list of security attributes]

[350] [assignment: the authorized identified roles]

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

[352] [selection: change default, query, modify, delete, [assignment: other operations]]

[353] [assignment: list of security attributes]

[354] [assignment: the authorized identified roles]

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

[356] [selection: change default, query, modify, delete, [assignment: other operations]]

[357] [assignment: list of security attributes]

[358] [assignment: the authorized identified roles]

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

[360] [selection: change default, query, modify, delete, [assignment: other operations]]

[361] [assignment: list of security attributes]

[362] [assignment: the authorized identified roles]

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

[364] [selection: change default, query, modify, delete, [assignment: other operations]]

[365] [assignment: list of security attributes]

[366] [assignment: the authorized identified roles]

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

[368] [selection: change default, query, modify, delete, [assignment: other operations]]

[369] [assignment: list of security attributes]

[370] [assignment: the authorized identified roles]

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

[372] [selection: change default, query, modify, delete, [assignment: other operations]]

[373] [assignment: list of security attributes]

User and Signer[374].

FMT_MSA.1/Privileged User (Management of security attributes)

Hierarchical to: No other components.

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

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/Privileged User

The TSF shall enforce:

1. Privileged User Creation SFP[375] to restrict the ability to create and query[376] the security attributes listed in FIA_USB.1 for Privileged User[377] to authorised Privileged User[378].

FMT_MSA.2 (Secure security attributes)

Hierarchical to: No other components.

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

FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.2.1

The TSF shall ensure that only secure values are accepted for all security attributes listed in FIA_USB.1[379].

FMT_MSA.3/Signer (Static attribute initialization)

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1/Signer

The TSF shall enforce Signer Creation SFP[380] to provide restrictive[381] default values for security attributes that are used to enforce the SFP.

[374] [assignment: the authorized identified roles]

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

[376] [selection: change default, query, modify, delete, [assignment: other operations]]

[377] [assignment: list of security attributes]

[378] [assignment: the authorized identified roles]

[379] [assignment: list of security attributes]

[380] [assignment: access control SFP, information flow control SFP]

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

FMT_MSA.3.2/Signer

The TSF shall allow the Privileged User[382] to specify alternative initial values to override the default values when an object or information is created.

Application Note 63

The Privileged User can specify alternative initial values for the following security attributes:

1. for R.Reference_Signer_Authentication_Data:
   - authfactor ("PWD + TOTP")
   - Initial userPWD (a string to be changed by the Signer)
   - salt for one-way transformation of the userPW (320 random bits)
   - TOTP secret (256 random bits)
2. for R.Signer:
   - uid (user name in the SAM)
3. Role ("Signer")
4. EntityType ("User" or "Org")

FMT_MSA.3/Privileged User (Static attribute initialization)

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1/Privileged User

The TSF shall enforce Privileged User Creation SFP[383] to provide restrictive[384] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Privileged User

The TSF shall allow the Privileged User[385] to specify alternative initial values to override the default values when an object or information is created.

Application Note 64

The Privileged User can specify alternative initial values for the following security attributes:

1. for R.Reference_Privileged_User_Authentication_Data
   - authfactor ("PWD+TOTP")
   - Initial userPWD (a string to be changed by the Privileged User)

[382] [assignment: the authorized identified roles]

[383] [assignment: access control SFP, information flow control SFP]

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

[385] [assignment: the authorized identified roles]

   - salt for one-way transformation of the userPW (320 random bits)
   - TOTP secret (256 random bits)
2. for R.Privileged_User
   - uid (user name in the SAM)
3. Role (“SAMadmin”)

FMT_MTD.1/SAM (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/SAM

The TSF shall restrict the ability to modify[386] the R.TSF_DATA[387] to Privileged User[388].

FMT_SMF.1/SAM (Security management functions)

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1/SAM

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

1. Signer management,
2. Privileged User management,
3. Configuration management[389],
4. Backup and restore functions[390].

FMT_SMR.2/SAM (Restrictions on security roles)

Hierarchical to: FMT_SMR.1 Security roles

Dependencies: FIA_UID.1 Timing of identification.

FMT_SMR.2.1/SAM

The TSF shall maintain the roles Signer and Privileged User, none[391].

[386] [selection: change default, query, modify, delete, clear, [assignment: other operations]]

[387] [assignment: list of TSF data]

[388] [assignment: the authorized identified roles]

[389] [assignment: list of security management functions to be provided by the TSF]

[390] [assignment: additional list of management functions to be provided by the TSF]

[391] CC: [assignment: authorised identified roles], PP: Signer and Privileged User, [assignment: authorised identified roles

FMT_SMR.2.2/SAM

The TSF shall be able to associate users with roles.

FMT_SMR.2.3/SAM

The TSF shall ensure that the conditions Signer can’t be a Privileged User[392] are satisfied.

6.1.3.6 Protection of the TSF (FPT)

FPT_RPL.1 **(Replay detection)**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_RPL.1.1

The TSF shall detect replay for the following entities: R.SAD[393].

FPT_RPL.1.2

The TSF shall perform reject the signature operation[394] when replay is detected.

FPT_STM.1/SAM **(Reliable time stamps)**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1/SAM

The TSF shall be able to provide reliable time stamps.

Application Note 65

The SAM receives a reliable time source from its environment (from the CM, through the OS).

Application Note 66

Since the SAM is implemented as a local application within the same physical boundary as the CM,

FPT_PHP.1 and FPT_PHP.3 do not apply for the SAM, because the FPT_PHP.1 and FPT_PHP.3 defined in [EN 419221-5] for the CM already provide a tamper-resistant environment.

FPT_TDC.1 **(Inter-TSF basic TSF data consistency)**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1

The TSF shall provide the capability to consistently interpret

1. R.Signer,

[392] [assignment: conditions for the different roles]

[393] [assignment: list of identified entities]

[394] [assignment: list of specific actions]

2. R.Reference_Signer_Authentication_Data,
3. R.SAD,
4. R.DTBS/R,
5. R.SVD
6. R.Privileged_User
7. R.Reference_Privileged_User_Authentication_Data
8. R.TSF_DATA [395]

when shared between the TSF and another trusted IT product.

FPT_TDC.1.2

The TSF shall use data integrity either on data or on communication channel[396] when interpreting the TSF data from another trusted IT product.

Application Note 67

Since the Trident does not store data outside its physical boundary, then FPT_TDC.1 is trivially satisfied.

6.1.3.7 Trusted path/channels (FTP)

FTP_ITC.1/CM **(Inter-TSF trusted channel)**

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1/CM

The TSF shall provide a communication channel between itself and **cryptographic module certified according to [EN 419 221-5]**[397] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/CM

The TSF shall permit TSF and a cryptographic module certified according to [EN 419 221-5][398] to initiate communication via the trusted channel.

FTP_ITC.1.3/CM

The TSF shall initiate communication via the trusted channel for:

1. Management functions, as specified in FMT_SMF.1[399]

Application Note 68

Since the SAM is implemented as a local application within the same physical boundary as the CM,

[395] [assignment: list of TSF data types]

[396] [assignment: list of interpretation rules to be applied by the TSF]

[397] [refinement: another trusted IT product]

[398] [selection: the TSF, another trusted IT product]

[399] [assignment: list of functions for which a trusted channel is required]

and the CM already provides a tamper-resistant environment, then FTP_ITC.1/CM is trivially satisfied.

FTP_TRP.1/SSA (Inter-TSF Trusted Path)

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1/SSA

The TSF shall provide a communication path between itself and local[400] **Privileged Users through SSA**[401] that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification[402].

FTP_TRP.1.2/SSA

The TSF shall permit local[403] **Privileged User through a trusted IT product**[404] to initiate communication via the trusted path.

FTP_TRP.1.3/SSA

The TSF shall require the use of the trusted path for:

1. FDP_ACC.1/Privileged User Creation,
2. FDP_ACC.1/Signer Creation,
3. FDP_ACC.1/Signer Maintenance
4. FDP_ACC.1/Signer Key Pair Generation,
5. FDP_ACC.1/Signer Key Pair Deletion,
6. FDP_ACC.1/Supply DTBS/R,
7. FDP_ACC.1/SAM Maintenance,
8. FDP_ACC.1/SAM Backup[405].

Application Note 69

Since the Trident does not support “Supply DTBS/R by the Privileged User” then (6) in

FTP_TRP.1.3/SSA is trivially satisfied.

FTP_TRP.1/SIC (Inter-TSF Trusted Path)

Hierarchical to: No other components.

[400] [selection: remote, local]

[401] [refinement: users]

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

[403] [selection: the TSF, local users, remote users]

[404] [refinement: SSA]

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

Dependencies: No dependencies.

FTP_TRP.1.1/SIC

The TSF shall provide a communication path between itself and remote[406] **Signers through the SIC**[407] that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification[408].

FTP_TRP.1.2/SIC

The TSF shall permit remote[409] **Signers through the SIC**[410] to initiate communication via the trusted path.

FTP_TRP.1.3/SIC

The TSF shall require the use of the trusted path for:

1. FDP_ACC.1/Signer Maintenance
2. FDP_ACC.1/Signer Key Pair Generation
3. FDP_ACC.1/Signer Key Pair Deletion
4. FDP_ACC.1/Signing[411].

Application Note 70 (Application Note 74 from [EN 419241-2]: Applied)
The SAM is not expected to verify the SIC as a communication end point and it may rely on the signer authentication.

6.1.4 Additional SFRs

In case of distributed configuration, there are a few additional SFRs in relation to the distributed structure of the TOE: FPT_ITT.1, FPT_SSP.2, FPT_TRC.1, and FRU_FLT.1.

In case of High Availability configuration, there is an additional SFR in relation to the active/passive (primary/secondary) node structure of the TOE: FRU_FLT.2.

There are three additional SFRs for trusted updates: FPT_TUD_EXT.1, FTP_TRP.1/Trusted Update and FMT_MOF.1/ManualUpdate.

Finally, there is an additional SFR for cryptographic key distribution (FCS_CKM.2)

6.1.4.1 Protection of the TSF (FPT)

FPT_ITT.1 **(Basic Internal TSF Data Transfer Protection)**

Hierarchical to: No other components.

Dependencies: No dependencies.

[406] [selection: remote, local]

[407] [refinement: users]

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

[409] [selection: the TSF, local users, remote users]

[410] [refinement: users]

[411] CC: [selection: initial user authentication, [assignment: other services for which trusted path is required]], PP: [selection: (1) FDP_ACC.1/Signer Key Pair Generation (2) FDP_ACC.1/Signer Maintenance (3) FDP_ACC.1/Signing (4) [assignment: other services for which trusted path is required]].

FPT_ITT.1.1

The TSF shall protect TSF data from disclosure and modification[412] when it is transmitted between separate parts of the TOE, **using the following mechanisms: TLS as defined in [RFC 5246].**

FPT_SSP.2 (Mutual trusted acknowledgement)

Hierarchical to: FPT_SSP.1 Simple trusted acknowledgement

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

FPT_SSP.2.1

The TSF shall acknowledge, when requested by another part of the TSF, the receipt of an unmodified TSF data transmission.

FPT_SSP.2.2 The TSF shall ensure that the relevant parts of the TSF know the correct status of transmitted data among its different parts, using acknowledgements.

FPT_TRC.1 (Internal TSF consistency)

Hierarchical to: No other components.

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

FPT_TRC.1.1

The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE.

FPT_TRC.1.2

When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for[413]**:**

1. The following management functions from FMT_SMF.1/CM:
   - Unblock of access due to authentication or authorisation failures,
   - User management,
   - Configuration management.
2. The following management functions in FMT_SMF.1/SAM,
   - Signer management,
   - Privileged User management,
   - Configuration management,
3. All cryptographic operations.

FPT_TUD_EXT.1 (Trusted Update)

Hierarchical to: No other components

Dependencies: FCS_COP.1/digsig_Non-multiparty Cryptographic operation (for cryptographic verification), or FCS_COP.1/hash Cryptographic operation (for cryptographic hashing)

[412] [selection: disclosure, modification]

[413] [assignment: list of functions dependent on TSF data replication consistency]

FPT_TUD_EXT.1.1

The TSF shall provide Administrators[414] the ability to query the currently executing version of the TOE firmware/software and no other TOE firmware/software version[415].

FPT_TUD_EXT.1.2

The TSF shall provide Administrators[416] the ability to manually initiate updates to TOE firmware/software and no other update mechanism[417].

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a published hash[418] prior to installing those updates.

6.1.4.2 Resource utilisation (FRU)

FRU_FLT.1 (Degraded fault tolerance)

Hierarchical to: No other components.

Dependencies: FPT_FLS.1 Failure with preservation of secure state

FRU_FLT.1.1 The TSF shall ensure the operation of the cryptographic services, listed in the following table[419] when the following failures occur:

- in case of distributed configuration: fatal error or a long-term network unavailability in k out of the n MPCAs /with possible (k,n) values in the following table/[420]:

non-distributed cryptographic services			
services	related SFRs of the CM functionality	related SFRs of the SAM functionality	(k,n)
signature/seal creation	FCS_COP.1/dig sig_Non_multiparty	FCS_COP.1/invoke_CM:dig sig_Non_multiparty	(1,2) (1,3) (1,4) (2,3) (2,4) (3,4)
encryption/decryption	FCS_COP.1/enc-dec _Non-multiparty	-	
Random number generation	FCS_RNG.1	-	
Cryptographic hash function	FCS_COP.1/hash	FCS_COP.1/SAM_hash	
TOTP verification	FCS_COP.1/TOTP_verification	FCS_COP.1/SAM_TOTP_verification	
message authentication code operation	FCS_COP.1/cmac operation	-	
Identification and authentication	FIA_UID.1/CM, FIA_UAU.1/CM, FIA_AFL.1/CM_authentication, FIA_AFL.1/CM_authorisation, FIA_UAU.6/AKeyAuth, FIA_UAU.6/GenKeyAuth	FIA_UID.2/SAM, FIA_UAU.1/SAM, FIA_AFL.1/SAM, FIA_UAU.5/Signer, FIA_UAU.5/Privileged user	

[414] [assignment: Administrators]

[415] [selection: the most recently installed version of the TOE firmware/software; no other TOE firmware/software version]

[416] [assignment: Administrators]

[417] [selection: support automatic checking for updates, support automatic updates, no other update mechanism]

[418] [selection: digital signature mechanism, published hash]

[419] [assignment: list of TOE capabilities]

[420] [assignment: list of type of failures]

Audit record protection	FAU_STG.2	-	
distributed cryptographic services			
services	SFRs of the CM	SFRs of the SAM	(k,n)
signature/seal creation	FCS_COP.1/dig_sig_Multiparty	FCS_COP.1/invoke_CM:dig_sig_Multiparty	(2,3) (2,4) (3,4)
RSA decryption	FCS_COP.1/dec_Multiparty	-	(2,3) (2,4) (3,4)

FRU_FLT.2 (Limited fault tolerance)

Hierarchical to: FRU_FLT.1 Degraded fault tolerance

Dependencies: FPT_FLS.1 Failure with preservation of secure state

FRU_FLT.2.1 The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur:

in case of High Availability configuration: fatal error in the active (online) MPCA node[421]

6.1.4.3 Trusted path/channels (FTP)

FTP_TRP.1/Trusted Update (Trusted Path)

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1/Trusted Update

The TSF shall provide a communication path between itself and remote[422] **supplier (developer/manufacturer)**[423] that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification and disclosure[424].

FTP_TRP.1.2/Trusted Update

The TSF shall permit the TSF[425] to initiate communication via the trusted path.

FTP_TRP.1.3/Trusted Update

The TSF shall require the use of the trusted path for trusted software/firmware update[426]

6.1.4.4 Security management (FMT)

FMT_MOF.1 /ManualUpdate (Management of security functions behaviour)

[421] [assignment: list of type of failures]

[422] [selection: remote, local]

[423] [refinement: users]

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

[425] [selection: the TSF, local users, remote users]

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

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1/ManualUpdate

The TSF shall restrict the ability to enable[427] the functions perform manual updates[428] to Administrator[429].

6.1.4.5 Cryptographic support (FCS)

FCS_CKM.2/Non-multiparty Cryptographic key distribution

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_CKM.2.1 /Non-multiparty

The TSF shall distribute cryptographic keys in accordance with **the** specified cryptographic key distribution methods **listed below**[430] that meets the following[431]: **standards noted for each distribution method**.

cryptographic key distribution method	standard
key agreement: **KAS ECC CDH Component** Curve: P-224, P-256, P-384, P-521 ED25519, ED448	[SP800-56Ar3] §5.7.1.2, §6.3.2
key agreement: **ECDH** Curve: K-(233, 283, 409, 571), B-(233, 283, 409, 571), brainpoolP(224, 256, 320, 384, 512)r1, brainpoolP(224, 256, 320, 384, 512)t1 prime239v(1,2,3), c2tnb239v(1,2,3), c2pnb(272, 304, 368)w1, c2tnb359v1, c2tnb431r1	[SP800-56Ar3], §5.7.1.2, §6.3.2
key transport: **KTS-IFC** with Hash: SHA-1, SHA2-(224, 256, 384, 512), SHA3-(224, 256, 384, 512) keyGenerationMethods: rsakpg2-basic scheme: KTS-OAEP-basic	[SP800-56Br2]

FCS_CKM.2/Multiparty Cryptographic key distribution

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

[427] [selection: determine the behaviour of, disable, enable, modify the behaviour of]

[428] [assignment: list of functions]

[429] [assignment: the authorised identified roles]

[430] [assignment: cryptographic key distribution method]

[431] [assignment: list of standards]

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1/Multiparty

The TSF shall distribute cryptographic keys in accordance with **the** specified cryptographic key distribution method**s listed below**[432] that meets the following[433]: **standards noted for each distribution method**.

cryptographic key distribution method	standard
key agreement: **KAS ECC CDH Component** Curve: P-224, P-256, P-384, P-521	[SP800-56Ar3] §5.7.1.2, §6.3.2

[432] [assignment: cryptographic key distribution method]

[433] [assignment: list of standards]

6.2 Security assurance requirements

Class Assurance	Assurance components
ADV: Development	ADV_ARC.1 Architectural Design with domain separation and non-bypassability
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
	ALC_FLR.3 Systematic flaw remediation
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.5 Advanced methodical vulnerability analysis

Table 6.8 Assurance requirements: EAL4 augmented by AVA_VAN.5 and ALC_FLR.3

6.3 Security requirements rationale

6.3.1 Security requirements coverage

6.3.1.1 Coverage for the Cryptography Module (CM)

	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/*[434]		X												
FCS_CKM.2/*[435]		X												
FCS_CKM.4/CM	X													
FCS_COP.1/*[436]		X												
FCS_RNG.1											X			
FIA_UID.1/CM				X										
FIA_UAU.1/CM				X										
FIA_AFL.1/CM_authentication				X										
FIA_AFL.1/CM_authorisation				X										
FIA_UAU.6/AKeyAuth				X		X								
FIA_UAU.6/GenKeyAuth				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/CM_Backup										X				
FDP_ACF.1/CM_Backup										X				

[434] FCS_CKM.1/* : FCS_CKM.1/key gen_Non-multiparty, FCS_CKM.1/key gen_Multiparty

[435] FCS_COP.2/*: FCS_COP.2/Non-multiparty, FCS_COP.2/Multiparty

[436] FCS_COP.1/* : FCS_COP.1/dig sig_Multiparty, FCS_COP.1/dig sig_Non_multiparty, FCS_COP.1/enc-dec _Non-multiparty, FCS_COP.1/dec_Multiparty, FCS_COP.1/hash, FCS_COP.1/TOTP_verification, FCS_COP.1/cmac operation

	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
FDP_SDI.2			X											
FDP_RIP.1	X				X									
FTP_TRP.1/Local			X	X			X	X	X					
FTP_TRP.1/Admin			X	X			X	X	X					
FTP_TRP.1/External			X	X			X	X	X					
FPT_STM.1/CM														X
FPT_TST_EXT.1													X	
FPT_PHP.1												X		
FPT_PHP.3												X		
FPT_FLS.1													X	
FMT_SMR.1/CM				X										X
FMT_SMF.1/CM				X										X
FMT_MTD.1/Unblock				X										
FMT_MTD.1/AuditLog														X
FMT_MSA.1/GenKeys					X									
FMT_MSA.1/AKeys					X									
FMT_MSA.3/Keys					X									
FAU_GEN.1/CM														X
FAU_GEN.2/CM														X
FAU_STG.2														X

Table 6.9 CM Security Objectives mapping to SFRs

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/CM 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/*, FCS_CKM.1/* and FCS_CKM.2/*[437] (as a dependency of FCS_CKM.1/key_gen_Non-multiparty)

OT.KeyIntegrity is addressed primarily by FDP_SDI.2 which requires integrity protection of keys and their attributes by the CM. 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 integrityprotected channel as part of FTP_TRP.1/Local, FTP_TRP.1/Admin and FTP_TRP.1/External.

OT.Auth is addressed by FIA_UID.1, FIA_UAU.1 and FIA_AFL.1/* for administrator authentication (with FMT_MTD.1/Unblock and its dependencies on FMT_SMR.1 and FMT_SMF.1 ensuring that appropriate roles and unblocking for 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, FTP_TRP.1/Admin and FTP_TRP.1/External. Authorisation for the use of secret keys is addressed by FIA_UAU.6/AKeyAuth and FIA_UAU.6/GenKeyAuth.

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 authorisation 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/AKeyAuth and FIA_UAU.6/GenKeyAuth.

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

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

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

[437] FCS_COP.2/*: FCS_COP.2/Non-multiparty, FCS_COP.2/Multiparty

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 time stamps 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 CM is required by FMT_SMR.1.

6.3.1.2 Coverage for the Signature Activation Module (SAM)

	OT.SIGNER_PROTECTION	OT.REF-SIG_AUTH_DATA	OT.SIG_KEY_GEN	OT.SVD	OT.PRIV_U_MANAGEMENT	OT.PRIV-U-AUTH	OT.PRIV_U_PROT	OT.SIGNER-MANAGEMENT	OT.SYSTEM-PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIG_AUTH_DATA_PROT	OT.DTBSR_INTEGRITY	OT.SIGN_INTEGRITY	OT.CRYPTO	OT.RANDOM	OT.SAM_BACKUP
FAU_GEN.1/SAM										X								
FAU_GEN.2/SAM										X								
FCS_CKM.1/*[438]			X													X		
FCS_CKM.1/**[439]																X		
FCS_CKM.4/SAM			X															
FCS_COP.1/*[440]																X		
FCS_COP.1/**[441]															X	X		
FCS_RNG.1[442]			X															
FDP_ACC.1/Privileged User Creation					X													
FDP_ACF.1/Privileged User Creation					X													
FDP_ACC.1/Signer Creation		X						X										
FDP_ACF.1/Signer Creation		X						X										

[438] FCS_CKM.1/* : FCS_CKM.1/invoke_CM:key gen_Non-multiparty, FCS_CKM.1/invoke_CM:key gen_Multiparty

[439] FCS_CKM.1/** : FCS_CKM.1/invoke_CM:TOTP_shared_secret,

[440] FCS_COP.1/* : FCS_COP.1/SAM_hash, FCS_COP.1/SAM_key_derivation, FCS_COP.1/SAM_TOTP_verification

[441] FCS_COP.1/** : FCS_COP.1/invoke_CM:dig sig_Non-multiparty, FCS_COP.1/invoke_CM:dig sig_Multiparty

[442] FCS_RNG.1 is a SFR of the CM functionality. /According to Application Note 39 in [EN 419241-2], the SFR FCS_RNG.1 only apply for SAM functionality, if the SAM is not implemented as a local application within the same physical boundary as the cryptographic module./

	OT.SIGNER_PROTECTION	OT.REF-SIG_AUTH_DATA	OT.SIG_KEY_GEN	OT.SVD	OT.PRIV_U_MANAGEMENT	OT.PRIV-U-AUTH	OT.PRIV_U_PROT	OT.SIGNER-MANAGEMENT	OT.SYSTEM-PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIG_AUTH_DATA_PROT	OT.DTBSR_INTEGRITY	OT.SIGN_INTEGRITY	OT.CRYPTO	OT.RANDOM	OT.SAM_BACKUP
FDP_ACC.1/Signer Maintenance		X																
FDP_ACF.1/Signer Maintenance		X																
FDP_ACC.1/Signer Key Pair Generation			X	X														
FDP_ACF.1/Signer Key Pair Generation			X	X														
FDP_ACC.1/Signer Key Pair Deletion								X										
FDP_ACF.1/Signer Key Pair Deletion								X										
FDP_ACC.1/Supply DTBS/R														X				
FDP_ACF.1/Supply DTBS/R														X				
FDP_ACC.1/Signing											X				X			
FDP_ACF.1/Signing											X				X			
FDP_ACC.1/SAM Maintenance									X									
FDP_ACF.1/SAM Maintenance									X									
FDP_ACC.1/SAM Backup																		X
FDP_ACF.1/SAM Backup																		X
FDP_ETC.2/Signer	X																	
FDP_IFC.1/Signer	X																	
FDP_IFF.1/Signer	X																	
FDP_ETC.2/Privileged User					X		X											
FDP_IFC.1/Privileged User					X		X											
FDP_IFF.1/Privileged User					X		X											
FDP_ITC.2/Signer	X																	
FDP_ITC.2/Privileged User					X		X											
FDP_UCT.1	X																	

	OT.SIGNER_PROTECTION	OT.REF-SIG_AUTH_DATA	OT.SIG_KEY_GEN	OT.SVD	OT.PRIV_U_MANAGEMENT	OT.PRIV-U-AUTH	OT.PRIV_U_PROT	OT.SIGNER-MANAGEMENT	OT.SYSTEM-PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIG_AUTH_DATA_PROT	OT.DTBSR_INTEGRITY	OT.SIGN_INTEGRITY	OT.CRYPTO	OT.RANDOM	OT.SAM_BACKUP
FDP_UIT.1	X																	
FIA_AFL.1/SAM						X					X							
FIA_ATD.1	X				X		X											
FIA_UAU.1/SAM						X					X							
FIA_UAU.5/Signer											X							
FIA_UAU.5/Privileged User						X												
FIA_UID.2/SAM					X		X	X										
FIA_USB.1	X		X		X		X											
FMT_MSA.1/Signer								X										
FMT_MSA.1/Privileged User					X			X										
FMT_MSA.2					X			X										
FMT_MSA.3/Signer								X										
FMT_MSA.3/Privileged User					X			X										
FMT_MTD.1/SAM									X									
FMT_SMF.1/SAM									X									
FMT_SMR.2/SAM									X									
FPT_RPL.1												X						
FPT_STM.1/SAM										X								
FPT_TDC.1	X				X													
FTP_TRP.1/SSA									X					X				
FTP_TRP.1/SIC												X	X	X				
FTP_ITC.1/CM			X												X			

Table 6.10 SAM Security Objectives mapping to SFRs

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

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

OT.SIGNER_KEY_PAIR_GENERATION is handled by the requirements for key generation and cryptographic algorithms in FCS_CKM.1 and FCS_COP.1. FCS_RNG.1 provides a random source for key generation. FCS_CKM.4 describes the requirements for key destruction. FDP_ACC.1/Signer Key Pair Generation and FDP_ACF.1/Signer Key Pair Generation describes access control for creating a key pair. FIA_USB.1 describes that R.Signing_Key_Id is associated with Signer. FTP_ITC.1/CM can be used to communicate securely with a CM.

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

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

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

OT.PRIVILEGED_USER _PROTECTION is handled by FDP_ETC.2/Privileged User, FDP_IFC.1/Privileged User, FDP_IFF.1/Privileged User, FDP_ITC.2/Privileged User, FDP_UCT.1, FDP_UIT.1 and FPT_TDC.1. The actual description of the data is described in FIA_ATD.1 and FIA_USB.1.

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

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

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

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

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

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

OT.DTBSR_INTEGRITY is covered by FTP_TRP.1/SSA and FTP_TRP.1/SIC requiring data transmission to be protected in integrity.

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

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

OT.RANDOM is covered by OT.RNG (security objective for CM).

OT.SAM_BACKUP is handled by FDP_ACC.1/SAM_Backup and FDP_ACF.1/SAM_Backup.

6.3.1.3 *Coverage for the additional Security Objectives*

	OT.TSF_ Consistency	**OT.PROT_ Comm**	**OT.Availability**	**OT.Updates**
FPT_SSP.2	X			
FPT_TRC.1	X			
FPT_ITT.1		X		
FRU_FLT.1			X	
FRU_FLT.2			X	
FPT_TUD_EXT.1				X
FTP_TRP.1/Trusted Update				X
FMT_MOF.1/ManualUpdate				X

Table 6.11 Additional Security Objectives mapping to SFRs

OT.TSF_Consistency is addressed by FPT_SSP.2, which requires mutual trusted acknowledgement during the communication between separate TOE parts and FPT_TRC.1 which requires the consistency of the TSF data when they are replicated between separate TOE parts.

OT.PROT_Comm is addressed by FPT_ITT.1 which requires protection of user and TSF data protection against disclosure and modification when they are transmitted between separate parts of the TOE.

OT.Availability

- in case of distributed configuration is addressed by FRU_FLT.1 which requires operation of core security function and ensures minimum service provision even during a breakdown of some TOE parts.
- in case of High Availability configuration is addressed by FRU_FLT.2 which requires operation of all security function and ensures complete service provision even during a breakdown of a TOE part (the active MPCA node).

OT.Updates is addressed by FPT_TUD_EXT.1, which requires means to authenticate firmware/software updates, FTP_TRP.1/Trusted Update. which requires trusted path for software/firmware update, and FMT_MOF.1/ManualUpdate, which restricts the ability to enable the functions to perform manual updates to Administrator.

6.3.2 Satisfaction of SFR dependencies

6.3.2.1 Satisfaction of dependencies for the Cryptographic Module (CM)

The dependencies between SFRs are addressed as shown in Table 6.9 Where a dependency is not met in the manner defined in [CC2] then a rationale is provided for why the dependency is unnecessary or else met in some other way.

SFR	Dependencies	Fulfilled by
FCS_CKM.1/*	**[FCS_CKM.2 or FCS_COP.1]** **FCS_CKM.4**	**FCS_COP.1/*** **FCS_CKM.4/CM**
FCS_CKM.2	**[FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1]** **FCS_CKM.4**	**FCS_CKM.1** **FCS_CKM.4**
FCS_CKM.4/CM	**[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]**	**FCS_CKM.1/***
FCS_COP.1/*	**[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]** **FCS_CKM.4**	**FCS_CKM.1/*** **FCS_CKM.4/CM**
FCS_RNG.1	**No dependencies**	**n/a**
FIA_UID.1/CM	**No dependencies**	**n/a**
FIA_UAU.1/CM	**FIA_UID.1**	**FIA_UID.1/CM**
FIA_AFL.1/*	**FIA_UAU.1**	**FIA_UAU.1/CM**
FIA_UAU.6/AKeyAuth	**No dependencies**	**n/a**
FIA_UAU.6/GenKeyAuth	**No dependencies**	**n/a**
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**

SFR	Dependencies	Fulfilled by
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/CM_Backup	**FDP_ACF.1**	**FDP_ACF.1/CM_Backup**
FDP_ACF.1/CM_Backup	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/CM_Backup** The dependency on FMT_MSA.3 is not relevant in this case since the attribute used in FDP_ACF.1/CM_Backup is determined by the ability of the user to authenticate as an administrator according to FIA_UAU.1.
FDP_SDI.2	**No dependencies**	**n/a**
FDP_RIP.1	**No dependencies**	**n/a**
FTP_TRP.1/Local	**No dependencies**	**n/a**
FTP_TRP.1/Admin	**No dependencies**	**n/a**
FTP_TRP.1/External	**No dependencies**	**n/a**
FPT_STM.1/CM	**No dependencies**	**n/a**
FPT_TST_EXT.1	**No dependencies**	**n/a**
FPT_FLS.1	**No dependencies**	**n/a**
FPT_PHP.1	**No dependencies**	**n/a**
FPT_PHP.3	**No dependencies**	**n/a**
FMT_SMR.1/CM	**FIA_UID.1**	**FIA_UID.1/CM**
FMT_SMF.1/CM	**No dependencies**	**n/a**
FMT_MTD.1/Unblock	**FMT_SMR.1** **FMT_SMF.1**	**FMT_SMR.1/CM** **FMT_SMF.1/CM**
FMT_MTD.1/AuditLog	**FMT_SMR.1** **FMT_SMF.1**	**FMT_SMR.1/CM** **FMT_SMF.1/CM**
FMT_MSA.1/GenKeys	**[FDP_ACC.1 or FDP_IFC.1]** **FMT_SMR.1** **FMT_SMF.1**	**FDP_ACC.1/KeyUsage and** **FDP_IFC.1/KeyBasics** **FMT_SMR.1/CM** **FMT_SMF.1/CM**

SFR	Dependencies	Fulfilled by
FMT_MSA.1/AKeys	**[FDP_ACC.1 or FDP_IFC.1]** **FMT_SMR.1** **FMT_SMF.1**	**FDP_ACC.1/KeyUsage and FDP_IFC.1/KeyBasics** **FMT_SMR.1/CM** **FMT_SMF.1/CM**
FMT_MSA.3/Keys	**FMT_MSA.1** **FMT_SMR.1**	**FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys** **FMT_SMR.1/CM**
FAU_GEN.1/CM	**FPT_STM.1**	**FPT_STM.1/CM**
FAU_GEN.2/CM	**FAU_GEN.1** **FIA_UID.1**	**FAU_GEN.1/CM** **FIA_UID.1/CM**
FAU_STG.2	**FAU_GEN.1**	**FAU_GEN.1/CM**

Table 6.12 Satisfaction of dependencies for CM

6.3.2.2 Satisfaction of dependencies for the Signature Activation Module (SAM)

SFR	Dependencies	Fulfilled by
FAU_GEN.1/SAM	**FPT_STM.1**	**FPT_STM.1/SAM**
FAU_GEN.2/SAM	**FAU_GEN.1** **FIA_UID.1**	**FAU_GEN.1/SAM** **FIA_UID.2/SAM**
FCS_CKM.1/*	**[FCS_CKM.2 or FCS_COP.1]** **FCS_CKM.4**	**FCS_COP.1/*** **FCS_CKM.4/SAM**
FCS_CKM.4/SAM	**[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]**	**FCS_CKM.1/invoke_CM:*_key_gen**
FCS_COP.1/*	**[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]** **FCS_CKM.4**	**FCS_CKM.1/*** **FCS_CKM.4/SAM**
FDP_ACC.1/Privileged User Creation	**FDP_ACF.1**	**FDP_ACF.1/Privileged User Creation**
FDP_ACF.1/Privileged User Creation	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/Privileged User Creation** **FMT_MSA.3/Privileged User**
FDP_ACC.1/Signer Creation	**FDP_ACF.1**	**FDP_ACF.1/Signer Creation**
FDP_ACF.1/Signer Creation	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/Signer Creation** **FMT_MSA.3/Signer**

SFR	Dependencies	Fulfilled by
FDP_ACC.1/Signer Maintenance	**FDP_ACF.1**	**FDP_ACF.1/Signer Maintenance**
FDP_ACF.1/Signer Maintenance	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/Signer Maintenance** **FMT_MSA.3/Signer**
FDP_ACC.1/Signer Key Pair Generation	**FDP_ACF.1**	**FDP_ACF.1/Signer Key Pair Generation**
FDP_ACF.1/ Signer Key Pair Generation	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/Signer Key Pair Generation** **FMT_MSA.3/Signer**
FDP_ACC.1/Signer Key Pair Deletion	**FDP_ACF.1**	**FDP_ACF.1/Signer Key Pair Deletion**
FDP_ACF.1/Signer Key Pair Deletion	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/Signer Key Pair Deletion** **FMT_MSA.3/Signer**
FDP_ACC.1/Supply DTBS/R	**FDP_ACF.1**	**FDP_ACF.1/Supply DTBS/R**
FDP_ACF.1/Supply DTBS/R	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/Supply DTBS/R** **FMT_MSA.3/Privileged User**
FDP_ACC.1/Signing	**FDP_ACF.1**	**FDP_ACF.1/Signing**
FDP_ACF.1/Signing	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/Signing** **FMT_MSA.3/Signer**
FDP_ACC.1/SAM Maintenance	**FDP_ACF.1**	**FDP_ACF.1/SAM Maintenance**
FDP_ACF.1/SAM Maintenance	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/SAM Maintenance** **FMT_MSA.3/Privileged User**
FDP_ACC.1/SAM Backup	**FDP_ACF.1**	**FDP_ACF.1/SAM Backup**
FDP_ACF.1/SAM Backup	**FDP_ACC.1** **FMT_MSA.3**	**FDP_ACC.1/SAM Backup** **FMT_MSA.3/Privileged User**
FDP_IFC.1/Signer	**FDP_IFF.1**	**FDP_IFF.1/Signer**
FDP_IFF.1/Signer	**FDP_IFC.1** **FMT_MSA.3**	**FDP_IFC.1/Signer** **FMT_MSA.3/Signer**
FDP_IFC.1/Privileged User	**FDP_IFF.1**	**FDP_IFF.1/Privileged User**
FDP_IFF.1/Privileged User	**FDP_IFC.1** **FMT_MSA.3**	**FDP_IFC.1/Privileged User** **FMT_MSA.3/Privileged User**
FDP_ETC.2/Signer	**[FDP_ACC.1 or FDP_IFC.1]**	**FDP_IFC.1/Signer**

SFR	Dependencies	Fulfilled by
FDP_ETC.2/Privileged User	**[FDP_ACC.1 or FDP_IFC.1]**	**FDP_IFC.1/Privileged User**
FDP_ITC.2/Signer	**[FDP_ACC.1 or FDP_IFC.1]** **[FTP_ITC.1 or FTP_TRP.1]** **FPT_TDC.1**	**FDP_IFC.1/Signer** **FTP_TRP.1/SSA and FTP_TRP.1/SIC** **FPT_TDC.1**
FDP_ITC.2/Privileged User	**[FDP_ACC.1 or FDP_IFC.1]** **[FTP_ITC.1 or FTP_TRP.1]** **FPT_TDC.1**	**FDP_IFC.1/Privileged User** **FTP_TRP.1/SSA** **FPT_TDC.1**
FDP_UCT.1	**[FTP_ITC.1 or FTP_TRP.1]** **[FDP_ACC.1 or FDP_IFC.1]**	**FTP_TRP.1/SIC and FTP_TRP.1/SSA** **FDP_IFC.1/Signer and FDP_IFC.1/Privileged User**
FDP_UIT.1	**[FTP_ITC.1 or FTP_TRP.1]** **[FDP_ACC.1 or FDP_IFC.1]**	**FTP_TRP.1/SIC and FTP_TRP.1/SSA** **FDP_IFC.1/Signer and FDP_IFC.1/Privileged User**
FIA_ATD.1	**No dependencies**	**n/a**
FIA_USB.1	**FIA_ATD**	**FIA_ATD.1**
FIA_UID.2/SAM	**No dependencies**	**n/a**
FIA_UAU.1/SAM	**FIA_UID.1**	**FIA_UID.2/SAM**
FIA_AFL.1/SAM	**FIA_UAU.1**	**FIA_UAU.1/SAM**
FIA_UAU.5/Signer	**No dependencies**	**n/a**
FIA_UAU.5/Privileged User	**No dependencies**	**n/a**
FMT_MSA.1/Signer	**[FDP_ACC.1 or FDP_IFC.1]** **FMT_SMR.1** **FMT_SMF.1**	**FDP_ACF.1/Signer Creation,** **FDP_ACF.1/Signer Key Pair Generation,** **FDP_ACF.1/Signer Maintenance,** **FDP_ACF.1/Supply DTBS/R and** **FDP_ACF.1/Signing** **FMT_SMR.1/SAM** **FMT_SMF.1/SAM**
FMT_MSA.1/Privileged User	**[FDP_ACC.1 or FDP_IFC.1]** **FMT_SMR.1** **FMT_SMF.1**	**FDP_ACF.1/Privileged User Creation** **FMT_SMR.1/SAM** **FMT_SMF.1/SAM**

SFR	Dependencies	Fulfilled by
FMT_MSA.2	**[FDP_ACC.1 or FDP_IFC.1]** **FMT_MSA.1** **FMT_SMR.1**	**FDP_ACF.1/Signer Creation,** **FDP_ACF.1/Signer Key Pair Generation,** **FDP_ACF.1/Signer Maintenance,** **FDP_ACF.1/Supply DTBS/R,** **FDP_ACF.1/Signing,** **FDP_ACF.1/Privileged User Creation,** **FDP_IFC.1/Signer and** **FDP_IFC.1/Privileged User** **FMT_MSA.1 /Signer and** **FMT_MSA.1/Privileged User** **FMT_SMR.1/SAM**
FMT_MSA.3/Signer	**FMT_MSA.1** **FMT_SMR.1**	**FMT_MSA.1/Signer** **FMT_SMR.1/SAM**
FMT_MSA.3/Privileged User	**FMT_MSA.1** **FMT_SMR.1**	**FMT_MSA.1/Privileged User** **FMT_SMR.1/SAM**
FMT_MTD.1/SAM	**FMT_SMR.1** **FMT_SMF.1**	**FMT_SMR.1/SAM** **FMT_SMF.1/SAM**
FMT_SMR.2/SAM	**FIA_UID.1**	**FIA_UID.2/SAM**
FMT_SMF.1/SAM	**No dependencies**	**n/a**
FPT_STM.1/SAM	**No dependencies**	**n/a**
FPT_RPL.1	**No dependencies**	**n/a**
FPT_TDC.1	**No dependencies**	**n/a**
FTP_ITC.1/CM	**No dependencies**	**n/a**
FTP_TRP.1/SSA	**No dependencies**	**n/a**
FTP_TRP.1/SIC	**No dependencies**	**n/a**

Table 6.13 Satisfaction of dependencies for SAM

6.3.2.3 Satisfaction of dependencies for the additional SFRs

SFR	Dependencies	Satisfied by
FPT_SSP.2	**FPT_ITT.1**	**FPT_ITT.1**
FPT_TRC.1	**FPT_ITT.1**	**FPT_ITT.1**

SFR	Dependencies	Satisfied by
FPT_ITT.1	**No dependencies**	**n/a**
FRU_FLT.1	**FPT_FLS.1**	**FPT_FLS.1**
FRU_FLT.2	**FPT_FLS.1**	**FPT_FLS.1**
FMT_MOF.1/ManualUpdate	**FMT_SMR.1,** **FMT_SMF.1**	**FMT_SMR.1,** **FMT_SMF.1**
FPT_TUD_EXT.1	**FCS_COP.1/digsig_Non-multiparty or FCS_COP.1/hash**	**FCS_COP.1/hash**
FTP_TRP.1/Trusted Update	**No dependencies**	**n/a**

Table 6.14 Satisfaction of dependencies for additional SFRs

6.3.3 Satisfaction of SAR dependencies

SAR	Dependencies	Satisfied by
EAL4 package	(dependencies of EAL4 package are not reproduced here)	By construction, all dependencies are satisfied in a CC EAL package
ALC_FLR.3	**No dependencies**	n/a
AVA_VAN.5	**ADV_ARC.1** **ADV_FSP.4** **ADV_TDS.3** **ADV_IMP.1** **AGD_OPE.1** **AGD_PRE.1** **ATE_DPT.1**	**ADV_ARC.1** **ADV_FSP.4** **ADV_TDS.3** **ADV_IMP.1** **AGD_OPE.1** **AGD_PRE.1** **ATE_DPT.1** (all are included in EAL4 package)

Table 6.15 Satisfaction of dependencies for assurance requirements

6.3.4 Rationale for chosen security assurance requirements

The assurance level for this ST is EAL4 augmented by AVA_VAN.5 and ALC_FLR.3. This ST conforms to Protection Profiles [EN 419221-5] and [EN 419241-2]. Both PPs [EN 419221-5] and [EN 419241-2] require strict conformance of the ST claiming conformance to these PPs. The assurance level for the PPs above is EAL4 augmented by AVA_VAN.5. Additional SAR of this ST is ALC_FLR.3.

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

ALC_FLR.3 has been included in addition to EAL4 to cause the evaluation of the TOE’s flaw remedation procedures which Trident users can rely on following the release of the TOE.

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.

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.

7 TOE summary specification

To fulfill the Security Functional Requirements, the Trident comprises the following Security Functions (SFs):

1. Roles and Authentication and Authorisation (SF_IA_CM and SF_IA_SAM)
2. Security management (SF_Management_CM and SF_Management_SAM)
3. Key Security (SF_Crypto_CM and SF_Crypto_SAM)
4. Access and information flow control (SF_Control_CM and SF_Control_SAM)
5. TSF data protection (SF_FPT_CM and SF_FPT_SAM)
6. Audit (SF_Audit_CM and SF_Audit_SAM)
7. Communication protection (SF_Comm_CM and SF_Comm_SAM)
8. Distributed structure (SF_Distributed_TOE)
9. High Availability structure (SF_HA_TOE)
10. Trusted Update (SF_Trusted Update)

In SF1-SF7 (named SF_*_CM) is related to the CM functionality, while the SF_*_SAM named SFs are related to the SAM functionality. SF8 details the special TOE capabilities based on its distributed structure.

7.1 Security Functionality

7.1.1 Roles, Authentication and Authorisation (SF_IA_CM and SF_IA_SAM)

SF_IA_CM

Roles

The CM maintains the Administrator, Key User, LCA and ECA roles, associating users with roles.

Application Note 36 from [EN 419221-5]:

> 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.

Trident support both types of Client Applications. The Key User acts through one of the client applications.

(Related SFRs are the following: FMT_SMR.1/CM)

Authentication and Authorisation

The CM uses a common method for identification and authentication in case of each role:

a unique user identifier + (static password or/and TOTP secret).

Before using a secret key an authorisation or a re-authorisation is required.

The CM blocks the account/key after a predefined number of consecutive failed authentication/ authorisation attempts.

(FIA_UID.1/CM; FIA_UAU.1/CM; FIA_UAU.6/AKeyAuth; FIA_UAU.6/GenKeyAuth; FIA_AFL.1/CM_authentication; FIA_AFL.1/CM_authorisation)

SF_IA_SAM

Roles

The SAM maintains the Privileged User and Signer roles associating users with roles.

The SAM ensures that all user have only one role, consequently a signer can't be a privileged user.

(FMT_SMR.2/SAM)

Authentication

For the Privileged Users, the SAM uses the same identification and authentication method as the CM: a unique user identifier + (static password or/and TOTP).

For the Signer the SAM requires two different authentication factors, a password (as the knowledge-based factor) and a TOTP (as the possession-based factor).

The identification and authentication method is: a unique user identifier + static password + TOTP.

The SAM blocks the account after a predefined number of consecutive failed authentication attempts. When a signer account has been locked the SAM also suspends the usage of all signing keys of the Signer.

The SAM maintains accounts (with different security attributes) belonging to individual users.

(*FIA_UID.2/SAM; FIA_UAU.1/SAM; FIA_UAU.5/Signer; FIA_UAU.5/Privileged User; FIA_AFL.1/SAM; FIA_ATD.1; FIA_USB.1)*

7.1.2 Security management (SF_Management_CM and SF_Management_SAM)

SF_Management_CM

The Administrator is able to (*FMT_SMF.1/CM*):

- unblock a blocked user account or a blocked key (*FMT_MTD.1/Unblock*),
- specify alternative initial value for the "Key Usage" security attribute, setting its value to "General" or to "Signing" (*FMT_MSA.3/Keys*)
- export and delete the local audit and Errorlog file (*FMT_MTD.1/AuditLog*),
- backup and restore of the CM's TSF state (*FDP_ACC.1/CM_Backup; FDP_ACF.1/CM_Backup*).

The Key User is able to modify the following attributes of his/her key (*FMT_MSA.1/AKeys; FMT_MSA.1/GenKeys*):

- Authorisation Data (to be used for authorisation and re-authorisation of a key)
- Uprotected Flag (which indicates whether the his/her stored key is protected only with an infrastructural key, or additionally with his/her Authorisation Data.)

- Operational Flag (which indicates whether the key is in operational state.)

SF_Management_SAM

There are the following SAM management functions (*FMT_SMF.1/SAM*):

- Signer management

  (*FDP_ACC.1/Signer Creation, FDP_ACF.1/Signer Creation; FMT_MSA.1/Signer 1); FMT_MSA.3/Signer; FDP_ACC.1/Signer Maintenance; FDP_ACF.1/Signer Maintenance; FMT_MSA.1/ Signer 5),6); FMT_MSA.2*)

- Privileged User management

  (*FDP_ACC.1/Privileged User Creation; FDP_ACF.1/Privileged User Creation; FMT_MSA.3/Privileged User; FMT_MSA.1/Privileged User; FMT_MSA.2*)

- Configuration management

  (*FDP_ACC.1/SAM Maintenance; FDP_ACF.1/SAM Maintenance, FMT_MTD.1/SAM*)

- Backup and restore functions

  (FDP_ACC.1/SAM Backup, FDP_ACF.1/SAM Backup)

7.1.3 Key Security (SF_Crypto_CM, SF_Crypto_SAM)

SF_Crypto_CM

This security function is related to the whole lifecycle of the keys:

- Key import

  (FDP_IFF.1.2/KeyBasics 3,4,5; FD FTP_TRP.1/Admin; FAU_GEN.1.1/CM i), FCS_CKM.1/key_gen_Non-multiparty)

- Key generation (The CM generates different types of keys for its supported cryptographic operations.)

  (FCS_CKM.1/key gen_Non-multiparty; FCS_CKM.1/key gen_Multiparty; FCS_CKM.2

  FCS_RNG.1; FMT_MSA.3.1/Keys; FAU_GEN.1.1/CM e),g),t))

  / Key generation is based on random bit generation (FCS_RNG.1). The internal state of the RNG is seeded by an PTRNG. The different TOE models use different PTRNGs. (See 1.2 (TOE reference) and 1.4 (TOE description) for details.) /

- Key distribution (The CM perform ML-KEM (Kyber) Key-Encapsulation Mechanism.)

- Key restore from backup

  (FDP_ACF.1.2/CM_Backup; FAU_GEN.1.1/CM k))

- Binding of a set of attributes to the key

  (FMT_MSA.3/Keys; FDP_ACF.1.1/KeyUsage 2; FDP_ACF.1.2/KeyUsage 1; FMT_MSA.1/GenKeys; FMT_MSA.1/AKeys; FAU_GEN.1.1/CM j))

- Storage of the key (The CM protects the integrity of keys and their attributes. The CM protects the confidentiality of secret keys and their sensitive attributes - in both cases, when keys are held inside the TOE and when key are stored externally.

  (FDP_SDI.2; FDP_IFF.1.5/KeyBasics 1,6; FAU_GEN.1.1/CM l)

- Key export (The CM provides a function to export non-Assigned secret keys)

  (FDP_IFF.1.1/KeyBasics 3,4 FDP_IFF.1.2/KeyBasics 1,4,5; FDP_IFF.1.5/KeyBasics 2,3,4,6; FTP_TRP.1/Admin; FAU_GEN.1.1 i)

- Key usage (The CM supports different approved algorithms for different purposes identified in the Table 1.3.)

  (FDP_ACF.1.1/KeyUsage 1,3; FDP_ACF.1.2/KeyUsage 2,3; FIA_UAU.6/AKeyAuth; FIA_UAU.6/GenKeyAuth; FDP_RIP.1; FIA_AFL.1/CM_authorisation; FMT_MTD.1/Unblock; FDP_IFF.1.2/KeyBasics 6; FCS_COP.1/ dig sig_Non_multiparty; FCS_COP.1/ dig sig_Multiparty;FCS_COP.1/enc-dec _Non-multiparty; FCS_COP.1/dec _Multiparty; FCS_COP.1/hash; FCS_COP.1/TOTP_verification; FCS_COP.1/cmac operation; FAU_GEN.1.1/CM h), q))

- Key backup (The CM provides a function to backup secret keys.)

  (FDP_ACF.1.2/CM_Backup 1,3,4; FAU_GEN.1.1 k)

- Key destruction (All secret keys and all authorisation data are zeroised (with physically overwriting) at the end of their lifecycle or after they have been deallocated.)

  (FCS_CKM.4/CM; FDP_RIP.1.1; FAU_GEN.1.1/CM f)

SF_Crypto_SAM

The SAM does not perform cryptographic operations with Key User's key and does not delete Key User's key. The SAM invokes the CM with appropriate parameters whenever a cryptographic operation, a key generation or a key deletion is required.

FCS_CKM.1/invoke_CM:; FCS_COP.1/invoke_CM:*; FCS_CKM.4/SAM.*
At the same time SAM performs non-distributed cryptographic operations with infrastructural keys.

FCS_CKM.1/SAM_*; FCS_COP.1/SAM_* .

7.1.4 Access and information flow control (SF_Control_CM and SF_Control_SAM)

SF_Control_CM

The CM enforces the following Security Function Policies:

- Key Basics (Import of secret keys are not allowed. Export of secret key is allowed only for non-Assigned keys with "Export Flag="yes". Public keys will always be exported with integrity protection of their key value and attributes. Unblocking access to a key will not allow any subject other than those authorised to access the key at the time when it was blocked. No subject will be allowed to access the plaintext value of any secret key directly or to access intermediate values in any operation that uses a secret key.)

  (FDP_IFC.1/KeyBasics; FDP_IFF.1/KeyBasics)

- Key Usage (The "Uprotected Flag" and "Operational Flag" key attributes can be changed only by the Key User. The Authorisation Data can be changed only by the Key User. Only subjects with current authorisation for a specific secret key are allowed to carry out operations using the plaintext value of that key. Only cryptographic functions permitted by the secret key's Key Usage attribute shall be carried out using the secret key.)

  (FDP_ACC.1/KeyUsage; FDP_ACF.1/KeyUsage)

- Backup (Only Administrator are able to perform the backup or restore function (restore function is under dual control). All backups are signed and encrypted. Consequently, any backup preserves their integrity and confidentiality.) *(FDP_ACC.1/CM_Backup; FDP_ACF.1/CM_Backup)*

SF_Control_SAM

The SAM enforces the following additional SFPs:

- Privileged User Creation (Only a Privileged User is able to create a new Privileged User's account) *(FDP_ACC.1/Privileged User Creation; FDP_ACF.1/Privileged User Creation)*
- Signer Creation (Only a Privileged User can carry out create a new Signers account) *(FDP_ACC.1/Signer Creation; FDP_ACF.1/Signer Creation)*
- Signer Maintenance (Only a Privileged User or the owner Signer is able to delete a key identifier and a public key from a Signer'account)

  (FDP_ACC.1/Signer Maintenance; FDP_ACF.1/Signer Maintenance)
- Supply DTBS/R (Only an authorised Privileged User is able supply the R.DTBS/R on behalf of the Signer.)

  (FDP_ACC.1/Supply DTBS/R; FDP_ACF.1/Supply DTBS/R)
- Signer Key Pair Generation (Only a Signer can carry out the keyreq SAP command, requesting a new asymmetric key pair generation. Only a Privileged User can carry out the keygen CMAPI command generating a new asymmetric key pair and assigning it to a Signer's account.)

  (FDP_ACC.1/Signer Key Pair Generation; FDP_ACF.1/Signer Key Pair Generation)
- Signer Key Pair Deletion (Only a Signer and a Privileged User can carry out the delkey SAP command, requesting a key pair deletion.)

  (FDP_ACC.1/Signer Key Pair Deletion; FDP_ACF.1/Signer Key Pair Deletion)
- Signing (Only a Signer can carry out the chgkeypwd SAP command (which establishes or modifies the key Authorisation Data) and the "SAD" SAP command.)

  (FDP_ACC.1/Signing; FDP_ACF.1/Signing)
- SAM Maintenance (Only a Privileged User can carry out the SAM Maintenance related commands, transmitting information to the SAM to manage roles and configuration.)

  (FDP_ACC.1/SAM Maintenance; FDP_ACF.1/SAM Maintenance)
- Signer (The order of "Signer" related commands is regulated and controlled.)

  (FDP_IFC.1/Signer; FDP_IFF.1/Signer)
- Privileged User (The order of "Privileged User" related commands is regulated and controlled.) *(FDP_IFC.1/Privileged User; FDP_IFF.1/Privileged User)*

7.1.5 TSF data protection (SF_FPT_CM and SF_FPT_SAM)

SF_FPT_CM

The CM ensures the security of its TSF data, including the following:

- Self-tests, which demonstrate the correct operation of the TSF (*FPT_TST_EXT.1)*

- Secure failure, the capability to preserve a secure state when the different types of failures occur (*FPT_FLS.1*),
- Tamper protection (tamper detecting -*FPT_PHP.1*- and tamper response -*FPT_PHP.3*- capability). The different TOE models implement tamper response (FPT_PHP.3) differently. (Details can be found in [Trident-ARC].
- Reliable time stamp, providing accurate time keeping among else for authentication, synchronization, and audit purposes. The TOE contains a battery-backed internal system clock that is used to generate reliable time stamps for the CM. To provide high accuracy, the internal system clock's time is synchronized with time servers configured by the Administrator using the NTP [RFC 5905] service. (*FPT_STM.1*)

SF_FPT_SAM

The SAM is implemented as a local application within the same physical boundary as the CM.

Consequently, the CM provides for the SAM the following security services:

- a tamper-resistant environment,
- demonstration of the correct operation of the TSF (with different self-tests),
- preservation of a secure state in case of different types of failures.
- reliable time stamp

Related SFR: ---

7.1.6 Audit (SF_Audit_CM and SF_Audit_SAM)

SF_Audit_CM

The CM audits all security related events. (*FAU_GEN.1/CM*)

Every audit record includes a reliable time stamp (date and time of the event), subject identity (if applicable), identifier of the related CM and a human readable descriptive string about the related event.

For audit events resulting from actions of identified users, the CM associates each auditable event with the identity of the user that caused the event. (*FAU_GEN.2/CM*)

The TOE provides a reliable time source to the CM. (*FPT_STM.1/CM*)

The CM automatically transfers the blocks of audit records to an external audit server.

If the transfer of an audit block has failed, the CM temporarily accumulates audit blocks locally in an audit directory. Only the Administrator is able to export and delete the local audit file. (*FMT_MTD.1/AuditLog; FMT_SMF.1/CM 3*)

All audit blocks have a serial number and are signed with an infrastructural key, so the CM detects unauthorised modification (including deletion) to the stored audit records in the audit trail.

When local audit storage exhaustion is detected, the CM requires the local audit file to be successfully exported and deleted by the Administrator before allowing any other security related actions. (*FAU_STG.2*)

SF_Audit_SAM

The SAM audits all security related events. (*FAU_GEN.1/SAM*)

Every audit record includes a reliable time stamp (date and time of the event), subject identity (if applicable), identifier of the related SAM and a human readable descriptive string about the related event. The audit records do not include any data which allow to retrieve sensitive data. For audit events resulting from actions of identified users, the SAM associates each auditable event with the identity of the user that caused the event. (*FAU_GEN.2/SAM*)

The SAM receives a reliable time source from the CM. (*FPT_STM.1/SAM*)

The SAM invokes the CM to protect its audit records (from unauthorised modification, deletion and audit storage exhaustion).

7.1.7 Communication protection (SF_Comm_CM and SF_Comm_SAM)

SF_Comm_CM

The CM implements and enforces:

- a secure channel based on TLS protocol, for communication with ECAs (*FTP_TRP.1/External, FPT_ITT.1*)
- a secure channel based on TLS protocol, for communication with Administrator, through SSA (*FTP_TRP.1/Local, FPT_ITT.1*)
- a secure channel based on SSH protocol, for communication with Administrators, using the console command interface in the provided limited shell (*FTP_TRP.1/Admin, FPT_ITT.1*),
- a direct channel for communication with Administrators, using the console command interface with a physical keyboard (*FTP_TRP.1/Admin*),
- a secure channel based on TLS protocol, for internal communication among MPCAs (*FTP_TRP.1/External, FPT_ITT.1*),
- a secure channel based on SSH protocol, for communication with Administrators, using the console command interface in the provided limited shell (*FTP_TRP.1/Trusted_Update*),

SF_Comm_SAM

The SAM implements and enforces:

- a secure channel based on TLS protocol, for communication with Privileged Users, through the SSA *(FTP_TRP.1/SSA, FPT_ITT.1),*
- a secure channel based on SSH protocol, for communication with Privileged Users, using the console command interface in the provided limited shell (*FTP_ITC*),
- a secure channel based on the proprietary SAP protocol *(FTP_TRP.1/SIC, FPT_RPL.1; FDP_UCT.1; FDP_UIT.1),*
- a direct channel for communication with Privileged Users, using the console command interface with a physical keyboard (*FTP_ITC*).

7.1.8 Distributed structure (SF_Distributed_TOE)

In case of distributed configuration, the Trident consists of n (n=2, 3 or 4) separate TOE parts (MPCAs) to operate as a logical whole in order to fulfill the requirements of this Security Target.

This security function based on the distributed structure of the Trident ensures the following:

- Distributed cryptography

  (FCS_CKM.1/key gen_Multiparty; FCS_CKM.1/Invoke_CM:key gen_Multiparty; FCS_COP.1/dig sig_Multiparty; FCS_COP.1/Invoke_CM:dig sig_Multiparty;

  FCS_COP.1/dec_Multiparty)
- Secret sharing

  (FCS_CKM.1/key gen_Multiparty; FCS_COP.1/dig sig_Multiparty; FCS_COP.1/dec_Multiparty)
- Consistency protection (*FPT_SSP.2, FPT_TRC.1, FPT_ITT.1*)
- Degraded fault tolerance (*FRU_FLT.1*)

7.1.9 High Availability structure (SF_HA_TOE)

When the TOE is set up in distributed configuration with non-distributed key generation, it provides a High Availability functionality, where each MPCA maintains identical state to each other.

This security function ensures the following:

- Limited fault tolerance (*FRU_FLT.2*)

7.1.10 Trusted Update (SF_Trusted Update)

The Trident provides an SSH communication path between itself and remote supplier (developer/manufacturer) for trusted software/firmware update.

This security function ensures the following:

- Trusted Update *(FPT_TUD_EXT.1),*
- Trusted path (*FTP_TRP.1/Trusted Update),*
- Management of security functions behaviour *(FMT_MOF.1/ManualUpdate).*

7.2 TOE summary specification rationale

This section shows that the TSF and assurance measures are appropriate to fulfill the TOE security requirements.

Each security functional requirement is implemented by at least one security function (with few exceptions, which are explained).

The mapping of SFRs and SFs is given in the 7.1 Table.

SFR	SF
CM functionality	
FAU_GEN.1/CM	SF_Audit_CM
FAU_GEN.2/CM	SF_Audit_CM
FAU_STG.2	SF_Audit_CM

SFR	SF
FCS_CKM.1/key gen_Multiparty FCS_CKM.1/key gen_Non-multiparty	SF_Crypto_CM, SF_Distributed_TOE SF_Crypto_CM
FCS_CKM.2	SF_Crypto_CM
FCS_CKM.4/CM	SF_Crypto_CM
FCS_COP.1/dig sig_Multiparty FCS_COP.1/dig sig_Non_multiparty FCS_COP.1/hash FCS_COP.1/enc-dec _Non-multiparty FCS_COP.1/dec _Multiparty FCS_COP.1/TOTP_verification FCS_COP.1/cmac operation	SF_Crypto_CM, SF_Distributed_TOE SF_Crypto_CM SF_Crypto_CM SF_Crypto_CM SF_Crypto_CMSF_ SF_Distributed_TOE SF_Crypto_CM SF_Crypto_CM
FCS_RNG.1	SF_Crypto_CM
FDP_ACC.1/KeyUsage FDP_ACC.1/CM_Backup	SF_Control_CM SF_Management_CM, SF_Control_CM
FDP_ACF.1/KeyUsage FDP_ACF.1/CM_Backup	SF_Crypto_CM, SF_Control_CM, SF_Management_CM, SF_Crypto_CM, SF_Control_CM
FDP_IFC.1/KeyBasics	SF_Control_CM
FDP_IFF.1/KeyBasics	SF_Crypto_CM, SF_Control_CM, SF
FDP_SDI.2	SF_Crypto_CM
FDP_RIP.1	SF_Crypto_CM
FIA_AFL.1/CM_authentication FIA_AFL.1/CM_authorisation	SF_IA_CM SF_IA_CM, SF_Crypto_CM
FIA_UID.1/CM	SF_IA_CM
FIA_UAU.1/CM	SF_IA_CM
FIA_UAU.6/AKeyAuth	SF_IA_CM, SF_Crypto_CM
FIA_UAU.6/GenKeyAuth	SF_IA_CM, SF_Crypto_CM
FMT_MSA.1/GenKeys FMT_MSA.1/AKeys	SF_Management_CM, SF_Crypto_CM SF_Management_CM, SF_Crypto_CM

SFR	SF
FMT_MSA.3/Keys	SF_Management_CM, SF_Crypto_CM
FMT_MTD.1/Unblock FMT_MTD.1/AuditLog	SF_Management_CM, SF_Crypto_CM SF_Management_CM, SF_Audit_CM
FMT_SMF.1/CM	SF_Management_CM, SF_Audit_CM
FMT_SMR.1/CM	SF_IA_CM
FPT_STM.1/CM	SF_Audit_CM
FPT_FLS.1	SF_FPT_CM
FPT_PHP.1	SF_FPT_CM
FPT_PHP.3	SF_FPT_CM
FPT_TST_EXT.1	SF_FPT_CM
FTP_TRP.1/Local FTP_TRP.1/Admin FTP_TRP.1/External	SF_Comm_CM SF_Comm_CM, SF_Crypto_CM SF_Comm_CM
SAM functionality	
FAU_GEN.1/SAM	SF_Audit_SAM
FAU_GEN.2/SAM	SF_Audit_SAM
FCS_CKM.1/invoke_CM:key gen_Multiparty FCS_CKM.1/invoke_CM:key gen_Non-mMultiparty	SF_Crypto_SAM, SF_Distributed_TOE SF_Crypto_SAM
FCS_CKM.4/SAM	SF_Crypto_SAM
FCS_COP.1/invoke_CM:dig sig_Multiparty FCS_COP.1/invoke_CM:dig sig_Non-Multiparty FCS_COP.1/SAM_hash FCS_COP.1/SAM_key_derivation FCS_COP.1/SAM_TOTP_verification	SF_Crypto_SAM, SF_Distributed_TOE SF_Crypto_SAM SF_Crypto_SAM SF_Crypto_SAM SF_Crypto_SAM
FDP_ACC.1/Privileged User Creation FDP_ACC.1/Signer Creation FDP_ACC.1/Signer Key Pair Generation FDP_ACC.1/Signer Maintenance FDP_ACC.1/Supply DTBS/R FDP_ACC.1/Signing	SF_Management_SAM, SF_Control_SAM SF_Management_SAM, SF_Control_SAM SF_Control_SAM SF_Management_SAM, SF_Control_SAM SF_Control_SAM SF_Control_SAM

SFR	SF
FDP_ACC.1/SAM Maintenance FDP_ACC.1/SAM Backup	SF_Management_SAM, SF_Control_SAM SF_Management_SAM
FDP_ACF.1/Privileged User Creation FDP_ACF.1/Signer Creation FDP_ACF.1/Signer Key Pair Generation FDP_ACF.1/Signer Maintenance FDP_ACF.1/Supply DTBS/R FDP_ACF.1/Signing FDP_ACF.1/SAM Maintenance FDP_ACF.1/SAM Backup	SF_Management_SAM, SF_Control_SAM SF_Management_SAM, SF_Control_SAM SF_Control_SAM SF_Management_SAM, SF_Control_SAM SF_Control_SAM SF_Control_SAM SF_Management_SAM, SF_Control_SAM SF_Management_SAM
FDP_IFC.1/Signer FDP_IFC.1/Privileged User	SF_Control_SAM SF_Control_SAM
FDP_IFF.1/Signer FDP_IFF.1/Privileged User	SF_Control_SAM SF_Control_SAM
FDP_ETC.2/Signer FDP_ETC.2/Privileged User	---[443] ---[444]
FDP_ITC.2/Signer FDP_ITC.2/Privileged User	---[445] ---[446]
FDP_UCT.1	SF_Comm_SAM
FDP_UIT.1	SF_Comm_SAM
FIA_AFL.1/SAM	SF_IA_SAM
FIA_UID.2/SAM	SF_IA_SAM
FIA_UAU.1/SAM	SF_IA_SAM
FIA_UAU.5/Signer	SF_IA_SAM
FIA_UAU.5/Privileged User	SF_IA_SAM
FIA_ATD.1	SF_IA_SAM
FIA_USB.1	SF_IA_SAM
FMT_MSA.1/Signer FMT_MSA.1/Privileged User	SF_Management_SAM, SF_Management_SAM
FMT_MSA.2	SF_Management_SAM

[443] Since the drQSCD does not export user data then FDP_ETC.2/Signer is trivially satisfied.

[444] Since the drQSCD does not export user data then FDP_ETC.2/Privileged User is trivially satisfied.

[445] Since the drQSCD does not import user data then FDP_ITC.2/Signer is trivially satisfied.

[446] Since the drQSCD does not import user data then FDP_ITC.2/Privileged User is trivially satisfied.

SFR	SF
FMT_MSA.3/Signer FMT_MSA.3/Privileged User	SF_Management_SAM SF_Management_SAM
FMT_MTD.1/SAM	SF_Management_SAM
FMT_SMF.1/SAM	SF_Management_SAM
FMT_SMR.2/SAM	SF_IA_SAM
FPT_STM.1/SAM	SF_Audit_SAM
FPT_RPL.1	SF_Comm_SAM
FPT_TDC.1	---[447]
FTP_TRP.1/SSA FTP_TRP.1/SIC	SF_Comm_SAM SF_Comm_SAM
FTP_ITC.1/CM	SF_Comm_SAM
functionality of the distributed structure	
FPT_TRC.1	SF_Distributed_TOE
FPT_SSP.2	SF_Distributed_TOE
FPT_ITT.1	SF_Comm_CM, SF_Comm_SAM, SF_Distributed_TOE
FRU_FLT.1	SF_Distributed_TOE
functionality of the High Availability structure	
FRU_FLT.2	SF_HA_TOE
functionality for the trusted update	
FPT_TUD_EXT.1	SF_Trusted_Update
FTP_TRP.1/Trusted Update	SF_Trusted_Update
FMT_MOF.1/ManualUpdate	SF_Trusted_Update

Table 7.1 Mapping of SFRs and SFs

[447] Since the drQSCD does not store data outside its physical boundary, then FPT_TDC.1 is trivially satisfied.

8 References and Acronyms

8.1 References

[AIS31] BSI AIS 20 / AIS 31: Functionality classes for random number generators, Version 2.0

[ANSI X9.52] American National Standard for Financial Services X9.52-1998: “Triple Data Encryption Algorithm Modes of Operation.” American Bankers Association, Washington, D.C., July 29, 1998.

[Assurance] COMMISSION IMPLEMENTING REGULATION (EU) 2015/1502 of 8 September 2015 on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market

[Balloon] Balloon Hashing: A Memory-Hard Function Providing Provable Protection Against Sequential Attacks, Dan Boneh1, Henry Corrigan-Gibbs1, and Stuart Schechter, ePrint, 2016.

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

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

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

[Trident-ARC] Security Architecture Description – Trident, the distributed remote Qualified Signature Creation Device, Version 2.6, 19 March 2025

[Trident-TDS] TOE Design Documentation – Trident, the distributed remote Qualified Signature Creation Device, Version 2.6, 19 March 2025

[Trident-ADMG] Trident Administrators’ Guide – CM and SAM, Version 2.9, 19 March 2025

[Trident-DEVG] Trident Developers’ Guide – CMAPI and SAP, Version 2.8, 5 December 2024

[cPP ND] collaborative Protection Profile for Network Devices - Version 2.1, 24-September-2018

[EN 419221-5] Protection Profiles for Trust Service Provider Cryptographic Modules - Part 5: Cryptographic Module for Trust Services, EN 419221-5:2018, May 2018

[EN 419241-1] Trustworthy Systems Supporting Server Signing - Part 1: General System Security Requirements, EN 419241-1:2018, July 2018

[EN 419241-2] Trustworthy Systems Supporting Server Signing - Part 2: Protection Profile for QSCD for Server Signing, EN 419241-2:2019, February 2019

[eIDAS] 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

[Imp_Regulation] COMMISSION IMPLEMENTING REGULATION (EU) 2015/1502 of 8 September 2015 on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market

[EN 319401] Electronic Signatures and Infrastructures (ESI);
General Policy Requirements for Trust Service Providers

[EN 319411-1] Electronic Signatures and Infrastructures (ESI);
Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements

[EN 319411-2] Electronic Signatures and Infrastructures (ESI);
Policy and security requirements for Trust Service Providers issuing certificates; Part 2: Requirements for trust service providers issuing EU qualified certificates

[ISO19790] ISO/IEC 19790:2012 Information technology - Security techniques - Security requirements for cryptographic modules 2015-10-01

[NIST IR 8413] NIST IR 8413 Third Round Status Report of the NIST Post-Quantum Cryptography Standardization Process, July 2022

[TS 119312] ETSI TS 119312 Electronic Signatures and Infrastructures (ESI);
Cryptographic Suites Version 1.4.2 Feb 2022

[SPHINCS+] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl: The SPHINCS+ Signature Framework, September 23, 2019, https://sphincs.org/data/sphincs+-paper.pdf

[Kyber] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint,Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé: CRYSTALS-Kyber, Algorithm Specifications And Supporting Documentation (version 3.02), August 4, 2021, https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf

[NTRU] NTRU - Algorithm Specifications And Supporting Documentation, September 30, 2020, https://www.cryptojedi.org/papers/ntrunistr3-20200930.pdf, https://github.com/jschanck/ntru, https://ntru.org

[PKCS#1] RSA Laboratories, PKCS #1: RSA Encryption Standard, Version v2.2

[PKCS#5] RSA Laboratories - PKCS #5: Password-based Cryptographic Standard, Version 2.1

[PKCS#7] RSA Laboratories - PKCS #7: Cryptographic Message Syntax Standard, Version 1.5

[PKCS#10] RSA Laboratories - PKCS #10: Certification Request Syntax Standard, Version 1.7

[PKCS#11] RSA Laboratories, PKCS #11: Cryptographic Token Interface Standard, Version v2.30

[FIPS 140-2] FIPS PUB 140-2: Security Requirements for Cryptographic Modules, May 25, 2001

[FIPS 140-3] FIPS PUB 140-3: Security Requirements for Cryptographic Modules, March 22, 2019

[FIPS 180-4] FIPS PUB 180-4: Secure Hash Standard (SHS), August 2015

[FIPS 186-5] FIPS PUB 186-5: Digital Signature Standard (DSS), February 2023

[FIPS 197] FIPS PUB 197: Advanced Encryption Standard (AES), November 26, 2001

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

[FIPS 202] FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August, 2015

[FIPS 203] FIPS PUB 203: Module-Lattice-based Key-Encapsulation Mechanism Standard, August 2024

[FIPS 204] FIPS PUB 204: Module-Lattice-Based Digital Signature Standard, August 2024

[FIPS 205] FIPS PUB 205: Stateless Hash-Based Digital Signature Standard, August 2024

[RFC 2104] RFC 2104 - HMAC: Keyed-Hashing for Message Authentication

[RFC 2797] Certificate Management Messages over CMS

[RFC 4226] RFC 4226 - HOTP: An HMAC-Based One-Time Password Algorithm

[RFC 4269] RFC 4269 - The SEED Encryption Algorithm

[RFC 4492] RFC 4492 - Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)

[RFC 4493] RFC 4493 - The AES-CMAC Algorithm

[RFC 5208] RFC 5208 - Public-Key Cryptography Standards (PKCS) #8:

Private-Key Information Syntax Specification Version 1.2

[RFC 5246] RFC 5246 - The Transport Layer Security (TLS) Protocol, Version 1.2

[RFC 5639] RFC 5639 - Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation

[RFC 5794] RFC 5794 - A Description of the ARIA Encryption Algorithm

[RFC 5869] RFC 5869 - HMAC-based Extract-and-Expand Key Derivation Function (HKDF)

[RFC 5905] RFC 5905 - Network Time Protocol Version 4: Protocol and Algorithms Specification

[RFC 6238] RFC 6238 - TOTP: Time-Based One-Time Password Algorithm

[RFC 7515] RFC 7515 - JSON Web Signature (JWS)

[RFC 7518] RFC 7518 - JSON Web Algorithms (JWA)

[RFC 7519] RFC 7519 - JSON Web Token (JWT)

[RFC 8032] RFC 8032 - Edwards-Curve Digital Signature Algorithm (EdDSA)

[Schnorr] C. P. Schnorr: Efficient identification and signatures for smart cards, CRYPTO 1989: Advances in Cryptology — CRYPTO' 89 Proceedings pp 239-252

[Silverman] R. D. Silverman: A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths, RSA Laboratories Bulletin No. 13, April 2000

[SEC 2] Standards for Efficient Cryptography - SEC 2: Recommended Elliptic Curve Domain Parameters (January 27, 2010, Version 2.0)

[SOGIS] SOG-IS, SOG-IS Crypto Evaluation Scheme, Agreed Cryptographic Mechanisms, version 1.3, March 2023

[SP800-38A] NIST Special Publication 800-38A: Recommendation for Block Edition Cipher Modes of Operation, December 2001

[SP800-38B] NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005

[SP800-38C] NIST Special Publication 800-38C: Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, May 2004

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

[SP800-38E] NIST Special Publication 800-38E: Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices, January 2010

[SP800-56A] NIST Special Publication 800-56A rev3: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised), 2018

[SP800-57] NIST Special Publication 800-57 Part 1 Revision 5: Recommendation for for Key Management: Part 1 – General, 2020

[SP800-90Ar1] NIST Special Publication 800-90A: Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015

[SP800-224] NIST Special Publication 800-224: Keyed-Hash Message Authentication Code (HMAC), draft, June 2024

[X9.142] American National Standard for Financial Services ANSI X9.142-2020 - Public Key Cryptography for the Financial Services Industry - The Elliptic

Curve Digital Signature Algorithm ECDSA, American National Standards Institute, September 10, 2020

8.2 Acronyms

AC	Access Control
ANSI	American National Standards Institute
API	Application Programming Interface
CA	Certificate Authority
CC	Common Criteria
CFB	Cipher Feedback Mode
CGA	Certificate Generation Application
CM	Cryptographic Module
CMbr	Cryptographic Module Bridge
CMC	Certificate Management protocol using CMS
CMS	Cryptographic Message Syntax
CSR	Certification Signing Request
DRNG	Deterministic RNG
DTBS	Data To Be Signed
DTBS/R	Data To Be Signed or its unique representation
EAL	Evaluation Assurance Level
ECA	External Client Application
ECC	Elliptic-curve Cryptography
ECDH	Elliptic-curve Diffie–Hellman
ECDSA	Elliptic-curve Digital Signature Algorithm
EN	European Standard
ETSI	European Telecommunications Standards Institute
FIPS	Federal Information Processing Standard
FORS	Forest of Random Subsets
GF	Galois Field
HA	High Availability
HMAC	Hashed-based Message Authentication Code
HOTP	HMAC-Based One-Time Password (Algorithm)
IEC	International Electrotechnical Commission
IFC	Information Flow Control
ISO	International Organization for Standardization
IT	Information Technology
JWA	Json Web Algorithms

JWS	Json Web Signature
JWT	Json Web Token
KEM	Key-Encapsulation Mechanism
KU	Key User
LCA	Local Client Application
MAC	Message Authentication Code
MPC	Multi-Party Computation
MPCA	Multi-Party Cryptographic Appliance
MPCM	Multi-Party Cryptographic Module
MPCMd	Multi-Party Cryptographic Module daemon
NTP	Network Time Protocol
OS	Operating System
OSP	Organizational Security Policy
PKCS	Public-Key Cryptography Standards
PP	Protection Profile
PTRNG	Physical true RNG
PRF	Pseudorandom Function
QSCD	Qualified Electronic Signature (or Electronic Seal) creation device
RAD	Reference Authentication Data
RFC	Request for Comments
RNG	Random Number Generator
RSA	Rivest, Shamir and Adleman cryptosystem
SAD	Signature Activation Data
SAM	Signature Activation Module
SAP	Signature Activation Protocol
SAR	Security Assurance Requirement
SCA	Signature Creation Application
SCAL	Sole Control Assurance Level
SCD	Signature Creation Data (private cryptographic key stored in the QSCD)
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirement
SIC	Signer’s Interaction Component
SO	Security Objective

SOGIS	Senior Officials Group Information Systems Security
SSA	Server Signing Application
ST	Security Target
SVD	Signature Verification Data (public cryptographic key)
TDM	Tamper Detecting Module
TLS	Transport Layer Security
TOE	Target of Evaluation
TOTP	Time-Based One-Time Password (Algorithm)
TSC	TSF Scope of Control
TSF	TOE Security Functionality
TSP	TOE Security Policy
TSP	Trust Service Provider
TW4S	Trustworthy System Supporting Server Signing
VAD	Verification Authentication Data
WOTS+	Winternitz One-Time Signature Plus