DocuSign

Security Target
for
DocuSign QSCD as Qualified Signature Creation Device

Evaluation according to Common Criteria EAL4+

Contents

Tables

Figures

1 Security Target Introduction

This Security Target describes the security objectives and security requirements for DocuSign QSCD version 1.0. The specifications are consistent with the *Common Criteria for Information Technology Security Evaluation, Version 3.1 ([3], [4] and [5]).*

1.1 Security Target Reference

DocuSign QSCD as a Signature Device Security Target, Version 4.0.26, DocuSign team, 22 December 2020.

Document Identification: *QSCD-cc-st-QSigCD-4.0.26.docx*

1.2 TOE Reference

Vendor Name:	**DocuSign**
Product Name:	**DocuSign QSCD**
Software version:	**1.0.0.0**
Hardware version:	**2.0.0.0**
Guidance Doc:	**DocuSign QSCD Appliance Administrator Guide Version 1.0.0.0, DocuSign QSCD Appliance Developer Guide Version 1.0.0.0, QSCD Appliance Preparative Procedures Administrator Guide Version 1.0.0.0**

1.3 TOE Overview

1.3.1 TOE TYPE

The DocuSign QSCD is a digital signature product intended to be used as a Qualified Signature Creation Device (QSCD) in a secure operational environment.
The DocuSign QSCD Appliance is a network attached Appliance consisting of computer hardware, hardware for tamper resistance, hardened operating system, internal database and the Appliance server software.
The TOE is the whole DocuSign QSCD Appliance.
The internal operating system is an hardened operating system. This means that the of-the-shelf operating system is going through configuration changes to have it adaptive and limited/minimal to include only functionality that is required by the TOE.
For example, only necessary services (such as the networking services) are running inside the Operating system.
Also, in the hardened operating system, the firewall of the operating system allows networking access only to the defined networking services.

In the following document the terms *the Appliance* or *DocuSign QSCD* are equivalent to the term *DocuSign QSCD Appliance* and thus all these terms represent the TOE.

1.3.2 TOE usage and major security features

1.3.2.1 General

The Appliance enables users of organizations or other users' communities to easily incorporate a digital signature into any type of content such as documents or data.
The TOE is based on the following Common Criteria Protection Profiles and built based on the following protection profiles:

- **HSM Protection Profile [1]** – This protection profile covers the base HSM requirement for a hardware based cryptographic module with the necessary strict infrastructure that enables the binding of a signer to his/her own signature keys.

- **Signature Activation Mode (SAM) Protection Profile [2]** – This protection profile covers functionality such as the management of users and their authentication mechanisms.
  The module enables the strict binding of a signer to his/her unique signature

key.

The following description is aligned with the security requirements of the above Protection Profiles but describes the unique functionality of the TOE.

1.3.2.2 TOE usage and major security features with relation to [2]

In [2], the following text covers the usage and major security features of the TOE. For every security feature, it will be noted what is the relevant scope that is covered by this TOE.
More information of how the TOE implements the security features are written in the following sections of this chapter

- *Operator management:*
  - *Privileged Users can create other Privileged Users.* Application Note: This is fully covered by the TOE using a special role named Users Administrator.
- *System management*
  - *Privileged Users can handle system configuration.* Application Note: This is fully covered by the TOE using a special role named Appliance Administrator.
- *Signer management covers:*
  - *Privileged Users can create Signers.* Application Note: This is fully covered by the TOE using a special role named SSA Admin.
  - *Privileged Users can assign on of the three authentication schemes (direct, indirect or mixed) to a Signer.* Application Note: Signers are authenticated only based on indirect authentication scheme.
  - *Privileged Users or Signers can generate signing keys and signature Verification Data (SVD) using a Cryptographic Module and assign the signing key identifier and SVD to a Signer.* Application Note: Only the SSA Admin generates signing keys for signers
  - *Privileged Users or Signers can disable a signing key identifier to be used by a Signer.* *Application Note: Only the SSA Admin can delete signing keys for* signers
- *Signature operation*
  - *Privileged Users or Signers can supply a DTBS/R(s) to be signed.* Application Note: This is fully covered by the TOE using the SSA Admin role. Signers do not have the option to supply a DTBS/R.
  - *The link between signer authentication, DTBS/R(s) and signing key identifier is handled by the Signature Activation Data (SAD). This SAD*

is securely exchanged with the TOE using the Signature Activation Protocol (SAP).
Application Note: The indirect signer authentication scheme is supported. As part of the scheme, the SAD is created by the IDP or by the SSA and securely links between the signer authentication, the transaction ID and the signer key. The transaction ID is randomly generated by the TOE and links to the DTBS/R.
As part of the SAP protocol, the SAD is uploaded to the TOE by the SSA as part of the digital signature operation.

Within the TOE the following actions are performed:

- *The SAD is verified in integrity.*
  Application note: performed by the TOE as part of the digital signature operation
- *The SAD is verified that it binds together the Signer authentication, a DTBS/R(s) and signing key identifier.*
  Application note: performed by the TOE as part of the digital signature operation
- *The Signer identified in the SAD is authenticated using one of the three authentication schemes.*
  Application note: performed by the TOE as part of the digital signature. Only the indirect authentication scheme is supported.
- *The DTBS/R(s) used for signature operations is bound to the SAD.*
  Application note: performed by the TOE as part of the digital signature operation. The DTBS/R is presented by the Transaction ID
- *The signing key identifier is assigned to the Signer.*
  Application note: performed by the TOE as part of the digital signature.
- *The TOE uses Authorisation Data to activate the signing key within the Cryptographic Module.*
  Application note: performed by the TOE as part of the digital signature operation. Authorisation is unique for the intended digital signature operation.
- *The TOE uses the Cryptographic Module to create signatures.*
  Application note: performed by the TOE as part of the digital signature operation.

- *The TOE generates audit records for all security related events and relies on the SSA to store and provide access control for the records.*
  Application note: performed by the TOE using an external Audit Log Server.

1.3.2.3TOE usage and major security features with relation to [1]

In [1], the following text covers the usage and major security features of the TOE. For every security feature, it will be noted what is the relevant scope that is covered by this TOE.
More information of how the TOE implements the security features are written in the following sections of chapter 1.3.2

1.3.2.3.1 General requirements

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

Application Note: The TOE is designed to be deployed inside the Secure environment of the Trust Service Provider (TSP)

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

Application Note: The TOE includes physical base tamper detection mechanisms.

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

Application Note: The design of the TOE (as part of also supporting [2], is providing the necessary mechanisms to avoid disclosing, compromise and unauthorised modification. Also, the TOE ensures that services are only used in an authorized way.

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

Application Note: Based on [2], only after the signer is authorized by the IDP or the SSA to use his/her signature key, the signature key is allowed for signing. Besides signatures keys all other keys are support keys.

1.3.2.3.2 Use Case 2: Support for Remote Server Signing

This use case is aimed at TSPs supporting requirements for remote signing, or sealing, as specified in Regulation 910/2014. In this case the TOE on its own is not intended to meet the requirements for QSCDs in the context of remote signing set out in Annex II of (EU) No 910/2014. It is expected that the TOE would be used in conjunction with the Protection Profile to be defined in EN 419241-2 [2], and any other related Protection Profiles, to meet the requirements for Sole Control Assurance Level 2 as defined in EN 419241-1 [23]. These security requirements may govern aspects such as the definition of specific user identification and authentication methods (e.g. multi-factor authentication) used within the signing system and may affect the type and form of the authorization data that is passed to the cryptographic module in order to authorize use of a key.

Application Note: This requirement is covered by the TOE as the TOE also covers [2] Protection Profile

The TOE performs local cryptographic operations, and associated key management, which can be used by an application using server signing, as defined in EN 419241-1 [9], 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 TOE. The TOE 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.

Application Note: This requirement is covered by the TOE as the TOE also covers [2] Protection Profile

The TOE deals with ensuring the security of keys and their use for signature or seal creation. Non-cryptographic functionality concerned with assuring sole control of these keys, for example authentication, is provided by other ensured functionality outside the scope of the TOE.

Application Note: This requirement is covered by the TOE as the TOE also covers [2] Protection Profile

1.3.2.4 Additional information related to TOE usage and major security features

The Appliance handles Signer accounts. Each Signer can include one or more signature keys and other information such as the transactions of the Signer. Signers and keys related data are stored externally to the Appliance in a protected manner.
No user, including the Appliance administrator or any other administrator, can use other user's key for digital signature operation.

Upon a Signer requesting to use his/her signature keys for digital signature operation, the Signer will need to activate his/her signature key using an indirect authentication scheme. The access to the Appliance is by the SSA though the network. The security of the networking protocol is based on TLS protocol. The TLS protocol is used for any request that is sent to the Appliance.

The signature application is based on a Web Application, the Web Application interacts with the Appliance through an SSA (Server Signing Application), where the Data to Be Signed Representation (DTBS/R) will be sent to the Appliance by the SSA prior to the digital signature operation. The SSA will get a transaction ID from the TOE that represents this DTBS/R.

The signature operation is performed using a Signature Activation Protocol (SAP), which requires that Signature Activation Data (SAD) be provided at the local environment. The SAD binds together three elements: signer authentication with the signing key and the data to be signed (DTBS/R(s)).

To ensure the signer has sole control of his signing keys, the signature operation needs to be authorised. This is carried out by a Signature Activation Module (SAM), which can handle one endpoint of SAP, verify SAD and activate the signing key within a Cryptographic Module. Both the Cryptographic Module and the SAM are to be located within a tamper protected environment. SAD verification means that the SAM checks the binding between the three SAD elements as well as checking that the signer is authenticated.

One of the three SAD elements is the signer authentication. The signer authentication is assumed to be conducted according to SCAL.2 for qualified signatures [23]. In this ST, it means that the signer authentication is carried indirectly by the SAM. In this case, an external authentication service as part of the TW4S and/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.

The Signer will need to approve the signature operation through a local component named SIC (Signer's Interaction Component).

The digital signature that was produced by the Appliance will be incorporated into a document such as a PDF file, XML data or any other document or data type.

Multiple users can sign simultaneously. Each user session is fully separated from other user sessions.

For every Signer the following sub-entities are managed:

- **Signature keys** – every Signer can have several signature keys that can be ephemeral or permanent key.

- **Transactions** – As part of the signature process, the SSA uploads the DTBS/R to the Appliance and is replied with a unique and randomly generated transaction ID. This Transaction ID is referenced as part of the SAP protocol.

Authentication Schemes

The following authentication schemes are used by the appliance:

- SAML token-based validation
  In this indirect scheme (delegated authentication as described in [2]), the IDP will be authenticating the user and the IDP or the SSA will produce the SAML token (depending on the used scheme of interaction with the IDP) as a proof of the user authentication. The SAML token includes the transaction ID.
  A SAML token validation will be performed inside the Appliance.
  Only after that the SAML token is validated, the TOE will continue with the digital signature operation.

- User-ID/Password
  This authentication scheme is used for authenticating administrators when connecting to the DocuSign QSCD

Follows a high-level scheme that shows how external entities interact with the Appliance.
The three schemes differ in the way the signer interacts with the IDP as part of the signature operation and the entity that creates and signs the SAML token (SAD). Besides this interaction, the other parts of the system and the TOE are the same.

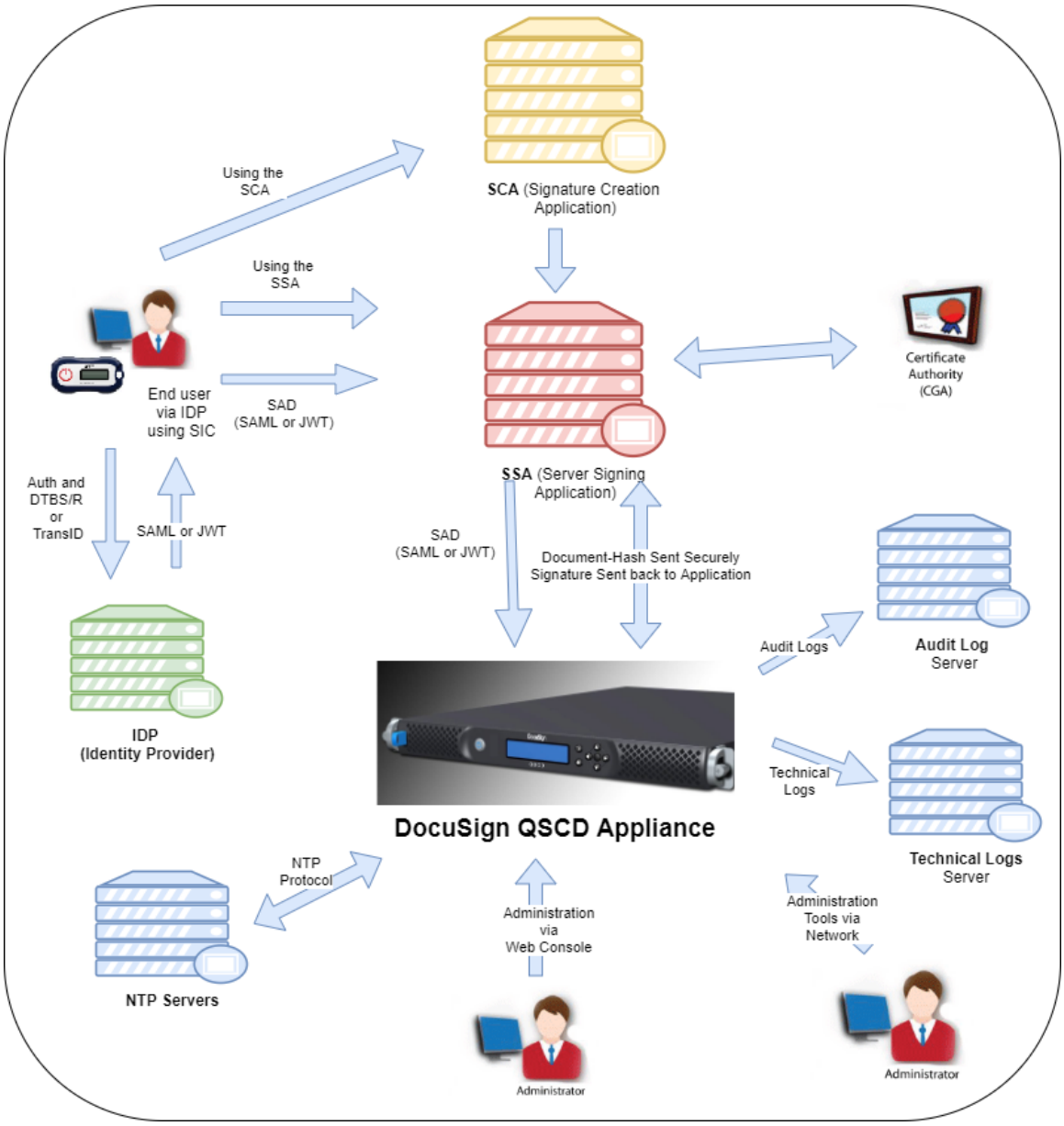


Figure 1.1 - DocuSign QSCD - High Level Design – Signature Creation Device – External IDP

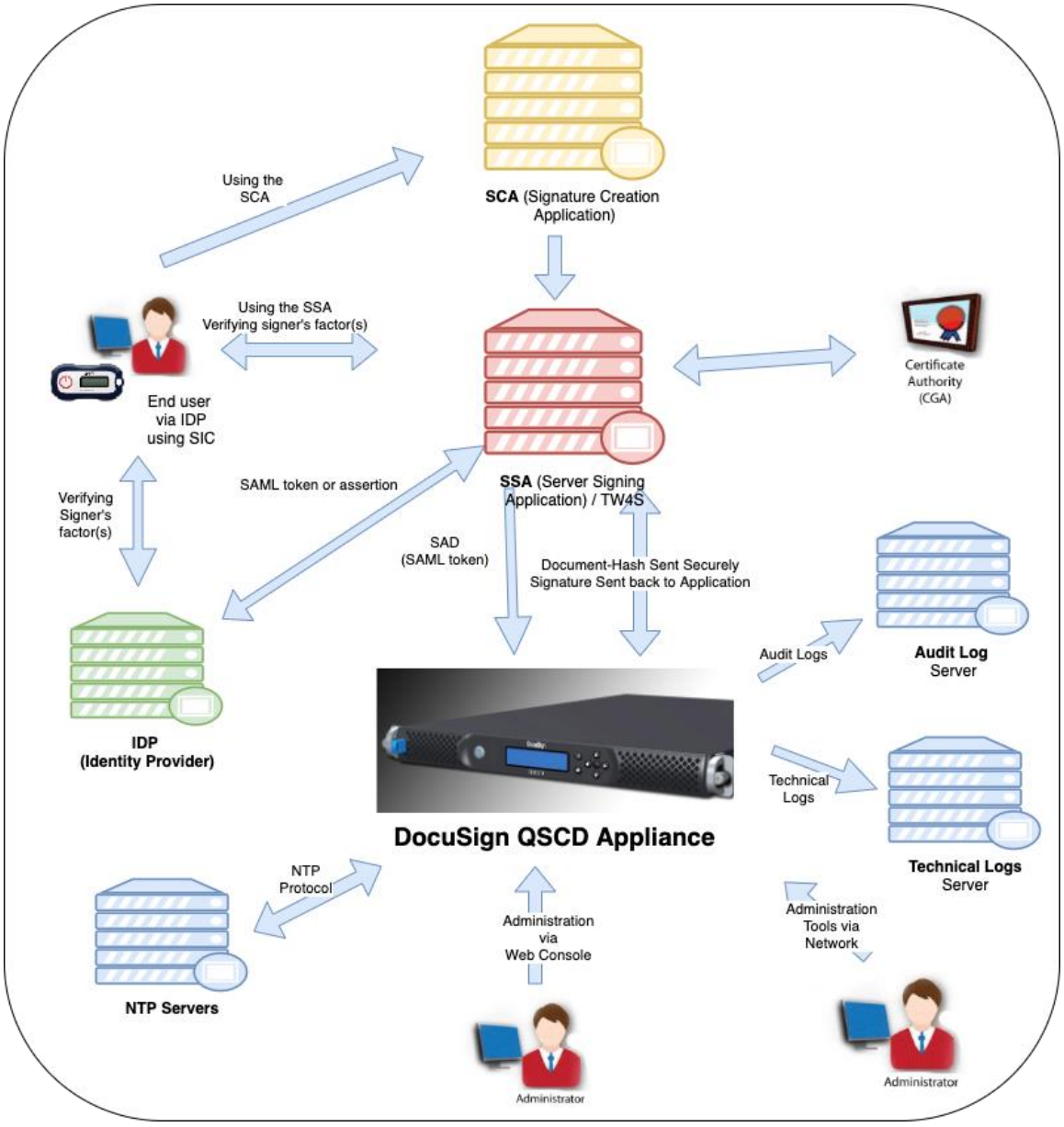


Figure 2.2 - DocuSign QSCD - High Level Design – Signature Creation Device – External IDP with SSA creating SAML token when IDP provides only an assertion (assertion is distinct from the SAD).

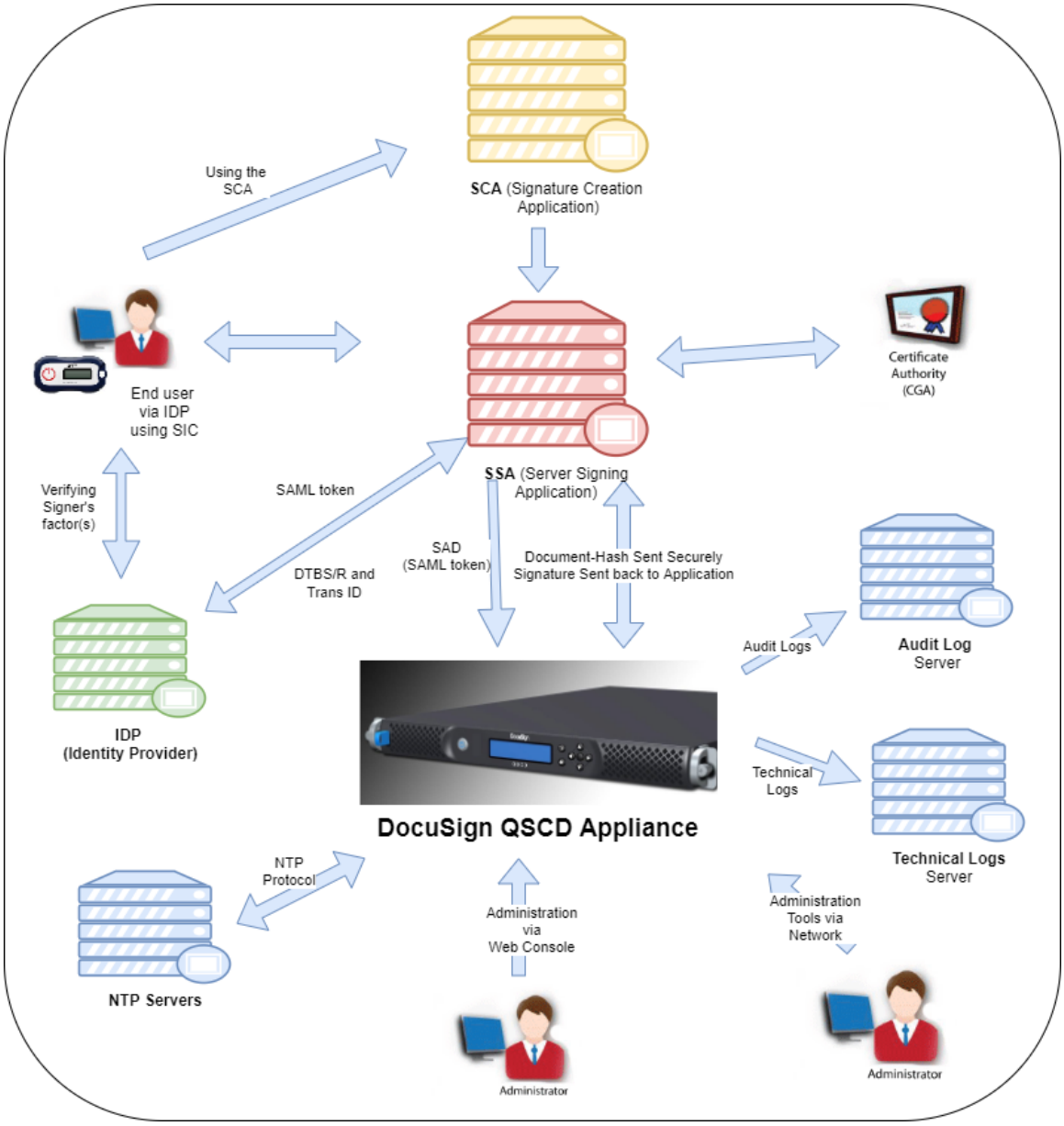


Figure 3 - DocuSign QSCD - High Level Design – Signature Creation Device– Internal IDP

1.3.3 Non-TOE hardware/software/firmware required by the TOE

The following non-TOE Hardware and Software are used in the operational environment:

- **SSA (Server Signing Application)**
  The SSA is a Web Application that is deployed in the operational environment

and enables the Signer to perform digital signatures through a Web Interface. The SSA will be used when the user signing experience is done through a Web Application.
In the case of using the Internal IDP model, the SSA interacts with the IDP for setting the DTBS/R and Transaction ID. The Internal IDP model will be used when the QTSP has its own IDP (refer to figure 2).
In the case of using external IDP model with the SSA (TW4S) the SSA will interact with IDP as verification of Signer's factor(s) maybe partially delegated. It means that the SAML Token is either generated by;
- the IDP only (refer to figure 1.1) when IDP authenticates and verifies all Signer's factors (in that case, the IDP creates and signs the SAML token and gives it to SSA that gives it to the TOE or SSA can create the SAML Token and give it to IDP for signing), or
- by the SSA (refer to figure 1.2) that collects the assertion from IDP that has been verified by one or several Signer's factor(s) and SSA verifies the other Signer factor. In that case the SSA creates and signs the SAML Token and gives it to the TOE.
The SSA will interact with Appliance in order to set the DTBS/R for the transaction and retrieving the Digital Signature at the end.
The user will interact with the Appliance through the SSA in order to activate the signature key for the purpose of digital signature operation.
The SSA will request for a generation of a new signature key (SCD) inside the Appliance and forward the returned certificate request to the CA.
The replied certificate will accessible by the SSA.
It is possible to request for a qualified certificate, this means that all digital signature operations using the qualified certificate will be defined as qualified digital signature operation. Also, it is possible to request for an advanced certificate, this means that all digital signature operations using the advanced certificate will be defined as an advanced signature operation.

- **SIC (Signer's Interaction Component)**
  The SIC is deployed inside the browser as part of the used Web Application. SIC is a component as described in [23].

- **SCA (Signature Creation Application)**
  The SCA is an application that is executed in the user's PC or in a Web Application.

  The SCA presents the data to be signed (DTBS) for review by the Signer, obtain prior to the signature process a decision by the Signer.
  The SCA will interact with the SSA for the purpose of digital signature creation, the digital signature will be replied back to the SCA, and the digital signature will be incorporated to the document by the SCA.

- **CA (Certificate Authority)**
  The CA generates certificates for signers based on the signature key that is generated in the Appliance.
  The SSA interfaces with the CA. It sends the certificate request to the CA and replies with a certificate.

- **IDP (IDentity Provider)**
  There exist an IDP either inside the operational environment (as described in Figure 2) or as a delegated IDP (as described in Figure 1.1 and Figure 1.2) .
  The IDP authenticates the signer and provides a proof of authentication in the format of a signed SAML token or format accepted by the SSA when the IDP doesn't verifies all Signer's factors. In this last case, the SAML token is created and signed by the SSA.
  The TOE will validate the SAML token and thus enable the signer to access his/her Signature Key.

- **Audit Logs Server**
  All audit information will be sent to an external audit log server.
  For performance reasons, several audit logs will be aggregated in the Appliance and sent out to the Audit Logs Server.

- **Technical Logs Server**
  Some technical logs information will be sent to an external technical logs server.

- **NTP Server**
  The TOE is synchronized with the NTP server for the purpose of having an accurate time.

- **Any administrator PC**
  Administrators can connect remotely to the Appliance using the DocuSign SA client deployed in their PC. There is a direct communication between the PC of the administrator to the Appliance using the TLS protocol.

1.4 TOE Description

The following section will describe in detail the DocuSign QSCD solution.

1.4.1 High level description of the DocuSign QSCD

The DocuSign QSCD is a network attached Appliance consisting of the following internal hardware components:

- Physically Secured Box
  The box is designed according to the security requirements of [25], and to the definition of level 3 as defined in [25]. In this level, the Appliance is designed to prevent access to any of the internal hardware components without causing a tamper to the Appliance.
  The whole hardware of the Appliance is included in the TOE.
  The only hardware element that is not part of the TOE is the dual power supply, which can be maintained without triggering the internal tamper device of the Appliance.

- Motherboard, memory, SSDs and CPU
  These components are off-the-shelf and state-of-the-art hardware components that enables the execution of the software modules of the Appliance.
  The internal non-volatile information of the Appliance is kept in one of the SSD of the Appliance.
  All these hardware components are part of the TOE.

- Tamper response and Tamper detection device
  This device is triggered whenever there is an attempt to access internal information of the Appliance. More information about the Tamper mechanism is explained in the sections below.
  The Tamper device is part of the TOE.

- Random Number Generation Device
  Internal Random number generator that provides that. Further details for the internal Random Number Generation device in section 1.4.3.2.4
  The RNG device is part of the TOE.

- Touch Screen
  The screen displays general information related to the Appliance.
  The device is part of the TOE.

- Dual Power Supply
  A redundant mechanism for providing power to the Appliance.
  This hardware component is not part of the TOE.

The following software provides the functionality of the Appliance:

- DocuSign QSCD software which is the major software component of the TOE and includes the main aspects of enabling end users to sign documents. All security related functionality is part of this component.

Figure 3 below provide a schematic description of the Appliance and the TOE.
All items marked with pink are included in the TOE.
Items mark in gray are excluded from the TOE.

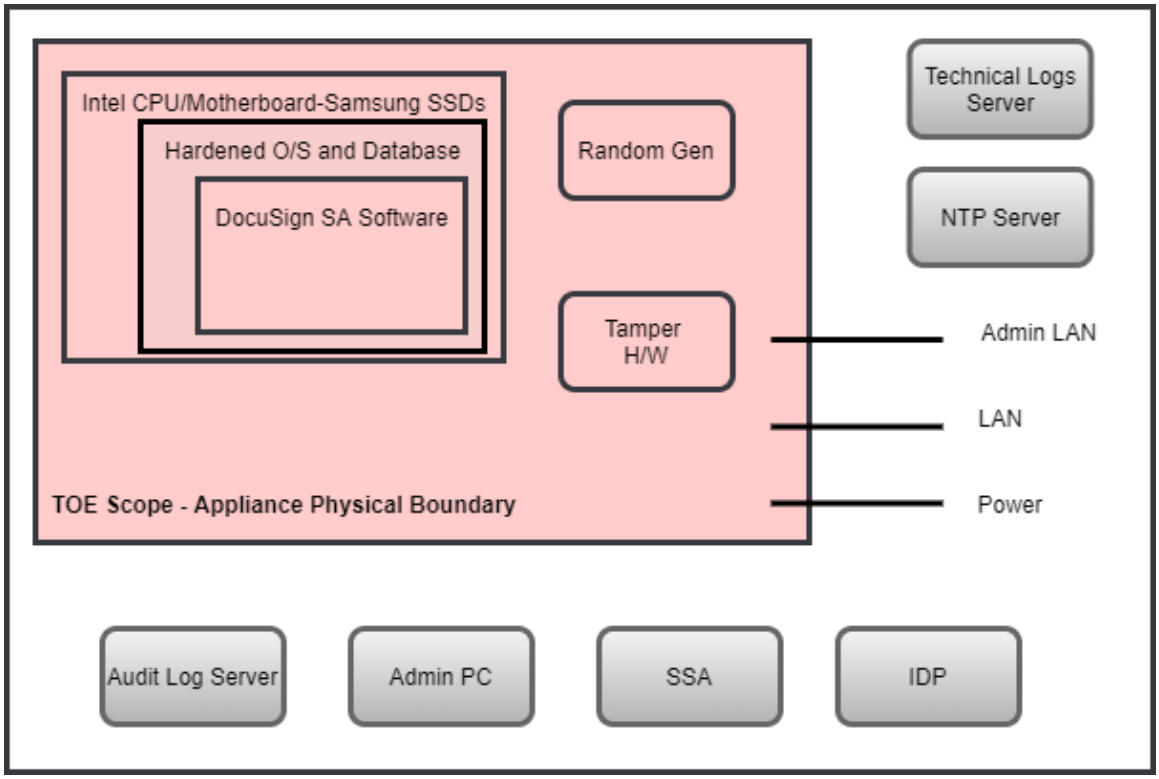


Figure 4 - DocuSign QSCD Internal Design

The Appliance is intended to be used as a digital signature product within an organizational environment and should be physically installed in a secure environment in the organization's data center and connected to the organizational network.
A single Appliance can securely manage many user accounts.

If a user wishes to digitally sign a document, the SIC will open a protected user session with the SSA.

Any communication with the Appliance, including the SSA communication with the Appliance, is based on TLS protocol Version 1.2 [15].
The TLS secure channel is based on TLS_RSA_WITH_AES_128_CBC_SHA256 or TLS_RSA_WITH_AES_256_CBC_SHA256 mechanisms, where the symmetric key establishment is based on a 2048 RSA key, the symmetric encryption algorithm is based on AES-128 in CBC mode or AES-256 in CBC mode. The data integrity algorithm is based on SHA256. The confidentiality and integrity elements are compliant with *[20] and [21]*.

Any creation of an audit log will be aggregated for a short while inside the Appliance and then be sent to an external audit server.

1.4.2 Cryptographic Module and Signature Activation Module (SAM)

As defined in [1] and [2], the TOE is a composition of a Cryptographic Module and a Signature Activation module deployed within the tamper protected environment of the Cryptographic Module.
The TOE is a multichip standalone appliance executing the functionality of both the cryptographic module and the SAM using the internal hardware components of the Appliance.
All the functionality of the Cryptographic Module's such as the cryptographic operations and key management are executed by processes running inside the internal Intel CPU of the TOE.
All the functionality of the SAM, such as making sure that only authorized signers can perform a digital signature with their designated signature keys are also executed by processes running inside the internal Intel CPU of the TOE.

There is a clear internal separation between the SAM and the CM (Cryptographic Module).
The SAM manages all users of the TOE and is capable of validate the users through presenting their SAD (Signature Activating Data). The SAM also keeps a strong binding of the signers to their signature keys.
Once the SAD is properly validated, the Cryptographic module uses his/her signature key to produce a digital signature.

1.4.3 TOE definition

Figure 5 - DocuSign QSCD Hardware – Front

Figure 6 - DocuSign QSCD Hardware – Back

The scope of the TOE is the whole DocuSign QSCD (see figures 3 and 4) including all Appliance's hardware and software components.

1.4.3.1 Physical Scope of the TOE Hardware

The Appliance is a steel, rack mountable box. The physical interfaces of the Appliance include the following elements:

- Network interface (Ethernet Interface using TCP/IP) – this component is part of the Appliance's motherboard
- Additional Network interface aimed for administration – this component is part of the Appliance's motherboard
- Power switch (a front switch)
- Two Power connectors (Dual Power Supply)
- Touch Screen for displaying minimal information
- A USB slot for a smartcard-based USB token

The internal hardware of the Appliance includes:

- Motherboard and CPU
- Two SSDs that maintains the Appliance's software and data

- A Tamper hardware device that automatically shut downs the Appliance when trying to open the Appliance. Also, critical information, such as the critical master keys is deleted when a temper event occurs.
  The tamper device is also responsible for providing true random seed that is used for generating signature keys among other random data.
- Dual Power supply and fans.

The TOE, which is based on the Physical box deployed with a certified software version is packaged in the developer site, tested and delivered to the end customer via courier delivery. The exact shipment address of the customer and its representative is defined as part of the sales process.
The customer's representative is required to check the completeness of the box upon receiving it from the developer.
The customer's representative is also required to validate that the two tamper seals in the back of the TOE are not damaged.

1.4.3.2 Logical Scope of the TOE

When powering on the Appliance, the Appliance software is activated. The software is using a hardened Operating System.

The Appliance's software includes several software modules that are aimed to enable end users to remotely access the Appliance and perform a digital signature operation and variant administrative roles to perform administrative operations.

One of the Administrative roles is the SSA Admin that performs many operations on behalf of the signer and also forwards the signer request for digital signature to the Appliance.

While the Appliance in operational state, the following sections describe the relevant services offered by the TOE.

1.4.3.2.1 Functional Signer related operations

Signers communicate securely with the Appliance through the SSA. The SSA communicate with the Appliance using TLS protocol over TCP/IP. The following operations are performed :

RSA based digital signature generation
Only after the TOE verifies the SAD, its binding to the user and to the already created transaction, the TOE performs a digital signature operation and replies with the digital signature.

Look at a following section of the description of the indirect authentication scheme of the signers.

1.4.3.2.2 Functional Administrative operations

An Appliance Administrator can perform administrative tasks through a secure DocuSign SA Client-DocuSign QSCD interface that is based on a secure network connection.

There are three types of administrators roles:

- **Appliance Administrator**
  installs the Appliance and manages Appliance related functionalities.
- **Users Administrators**
  manages administrative user accounts
- **SSA Admin**
  This role creates and deletes Signers Accounts and also perform many of the operations that are required for enabling the signer to eventually perform a digital signature operation.

Here follows some of the operations that can be performed by the Appliance Administrator or Users Administrator:

- The Users Administrator can perform User management operations (creating a new admin user account, deleting an existing admin user account or viewing admin user information)
- The Appliance Administrator can install the Appliance and change system parameters.
- The Appliance administrator can upload digitally signed software updates. The updated software will need to be also Common Criteria certified under this Security Target or an updated version of this Security Target
- The Appliance administrator can download technical logs
- The Appliance administrator can perform a backup operation.

For more information, refer to section 7.1.

1.4.3.2.3 Functional SSA Admin related operations

The SSA communicate securely with the Appliance using TLS protocol over TCP/IP. The communication is authorized using a special Administrative SSA Admin user.
The following operations are performed:

Signer Creation
A new Signer-Keys blob is created for the signer. The Signer-Keys blob is kept externally to the TOE in a protected manner.

RSA key generation
Generating a new RSA key. The generated signing key is performed internally inside the Appliance. The Appliance contains a hardware random generator which is part of the tamper device. Using a pseudo random generation (HMAC-DRBG – NIST SP 800-90Arev1) [16], the required random for the key generation is provided.
The RSA key generation algorithm is compliant with [10], [6] and [9]. The RSA key can be of one of the following size: 2048 bits, 3072 or 4096 bits.

RSA Key Deletion
An existing key can be deleted for a signer.

Supply DTBS/R
The SSA initiates the transaction by supplying the DTBS/R to the TOE. A randomly generated transaction ID is generated and delivered by the SSA to the Signer. The DTBS/R will be used as an input for the digital signature operation when the transaction ID will be supplied by the signer.

1.4.3.2.4 DocuSign QSCD's random number generation

The Appliance includes a built-in random number generation that is used in a large variety of operations such as signature key generation and other sensitive information as well as a set of critical keys, which are described in the next section.
The random generation is aligned with [6] and is based on using a both True Random number generation mechanism (trueran) and a pseudo-random (pseuran) number generation mechanism.
The true random is based on a chip that is integrated into the tamper device. The technology of the random chip is based on true quantum randomness from the shot noise of a light source captured by a CMOS image sensor.
The pseudo-random generation uses the above true random seed and calculates the random number using the deterministic algorithm described in [16].

1.4.3.2.5 Appliance's Master Keys

The Appliance uses the following critical AES-256 keys (256bit length) or shared secrets that are generated during the Appliance installation and are in both volatile memory of the Appliance and inside the internal tamper device:

- **Appliance KEK – Master key used for Key encryption (MK-KEK)**
  This 256bit critical key encrypts the signature keys of the signers.
  The SRV KEK encrypts also other information such pre-generated RSA keys.
- **Appliance Data Integrity secret – Master secret used for HMAC calculation/verification of database records (MK-MAC)**
  This 256bit critical secret protects the integrity of all the user information, key information, other user objects and other sensitive information in the database of the Appliance.
- **Appliance backup Encryption – Master key used for Encryption of the Appliance's backup (MK-BKP-ENC)**
  This 256bit critical key encrypts the backup of the Primary Appliance.
  The backup includes only configuration information and administrative information and does not include keys.
  This key is also used for encrypting a Signer-Keys blob when stored outside the TOE)
- **Appliance backup Integrity – Master secret used for HMAC calculation/verification of the Appliance's backup (MK-BKP-MAC)**
  This 256bit critical secret protects the integrity of the backup of the Appliance.
  This key is also used for HMACing a Signer-Keys blob when stored outside the TOE)

All generated critical keys use the Appliance random generation mechanism, as defined in the above section.

All critical keys and secrets are kept in dedicated SmartCard based USB tokens for installation and restoration purpose.
Each token is protected by a password known only to the Appliance Administrator.
To achieve dual control, the critical keys are split to two tokens using the Shamir's secret sharing algorithm [11]. Each token should be given to a different Appliance Administrator. A copy of the keys should be prepared as well.
The backup USB tokens are prepared during a special master key generation operation and its secured information is copied to an additional set of dedicated USB tokens.
The backup USB tokens must be kept in dedicated safes in the responsibility of a dedicated administrative personal.

The backup tokens are used in the following operations:

- **Installation of the Appliance**
  The Appliance Administrators will be required to provide the tokens as part of the installation process.
- **Reset Tamper**
  In the case of a tamper event, the Appliance's Administrators can perform a reset tamper operation.
  The Appliance Administrators should perform the Reset Tamper operation only if they are absolutely certain that the Appliance was opened in a

controlled manner.
In the case that the tamper event occurred as part of a security compromise, it is forbidden to perform the Reset Tamper operation due to the risk that bringing the Appliance to a production state may compromise inner information such as the Signer's keys.
In the case that the Appliance's Administrators approve the Reset Tamper operation, the backup USB tokens are required since all above critical information is wiped out from the tamper device and the only way to reconstruct the information is using the backup USB tokens. This operation is initiated for a special administrative client.

1.4.3.2.6 Auditing

As defined in [1] and [2], audit information to operations performed by the TOE are directed to an external Audit Server. Information is sent to the audit server in a protected manner.

1.4.4 Delivery method of TOE components

The Appliance is delivered from DocuSign Manufacturing via courier in a protected manner. Also empty USB tokens are delivered in the Appliance packaging.
As part of the delivered Appliance also a USB device is delivered. This USB device contains the User Guides and client software related components.
All these items are digital signed by DocuSign Code Signing certificate.
The content of this USB device may also be sent to the customer via email or cloud sharing from DocuSign Support to the customer.

2 Conformance Claim

2.1 General Conformance Claim

The TOE claims to be Common Criteria Part 2 extended and Common Criteria Part 3 conformant and written according to the Common Criteria version 3.1 revision 5 [3], [4] and [5].

The assurance requirement of this Protection Profile is **EAL4 augmented**. Augmentation results from the selection of:

- AVA_VAN.5 - Advanced methodical vulnerability analysis

2.2 PP Claim

The TOE claims strict conformance to the following PPs:

1. Protection profiles for TSP Cryptographic modules - Part 5 Cryptographic Module for Trust Services [1]
2. Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing [2]

3 Security Problem Definition

The following chapter defines the security problems that need to be addressed as part of the TOE. The chapter will enumerate the Threats, OSPs and assumptions that relate to the Security problem definition.
As the TOE is based on both [1] and [2], the term *Cryptographic Module* represent the part of the TOE that is inline with [1].

3.1 Assets

The TOE has the following assets, which are to be protected in integrity and confidentiality as described below. The TOE must ensure that whenever an asset is persisted outside the TOE, the TOE has performed the necessary cryptographic operations to enforce confidentiality and detect if an asset has been modified. Access control to TOE assets outside the TOE are to be enforced by the environment.

1. **R.SecretKey** (from [1]): secret keys used in symmetric cryptographic functions and private keys used in asymmetric cryptographic functions, managed and used by the TOE in support of the cryptographic services that it offers. This includes user keys, owned and used by specific users, and support keys used in the implementation and operation of the TOE. The asset also includes copies of such keys made for external storage and/or backup purposes. The confidentiality and integrity of these keys must be protected.

2. **R.SigningKeyID** (from [2]) : The signing key is the private key of an asymmetric key pair used to create a digital signature under the signer's sole control. The signing key can only be used by the Cryptographic Module. The TOE uses the asset R.SigningKeyID, which identifies a signing key in the Cryptographic Module. The binding of the R.SigningKeyID with R.Signer shall be protected in integrity.

   Application Notes:

   - From [2]: *The integrity and confidentiality of the signing key and the link between the R.SigningKeyID and the signing key is the responsibility of the Cryptographic Module. The TOE shall ensure that only the signer can use the signing key under his sole control.* - This note is satisfied by the TOE.
   - The Signing key as defined in [2] is a subclass of SecretKey as defined in [1].

3. **R.AuthorisationData** (from [2]) : is data used by the TOE to activate a signing key in the Cryptographic Module. The signing key is identified by R.SigningKeyID. It shall be protected in integrity and confidentiality.

   Application Notes:

   - From [2]: *The R.AuthorisationData is used by the Cryptographic Module to activate a signing key. The data may be an asset of the TOE or derived by the TOE from the SAD.*

In both cases, the TOE must verify the SAD before the R.AuthorisationData is used to activate the signing key in the Cryptographic Module
– This note is satisfied by the TOE.

- From [2]: *If the TOE derives the R.AuthorisationData from SAD then this data may not be held by the TOE*
  – This note is satisfied by the TOE. The authorisation data is not held by the TOE and derived from the SAD after a proper SAD validation.
- The R.AuthorisationData is transient and created based on a successfully SAD validation by the TOE. The authorization data is removed just after the digital signature operation.

4. **R.SVD** (from [2]) **or R.PubKey** (from [1])**:**
   signature verification data is the public part, associated with the signing key, to perform digital signature verification. The R.SVD shall be protected in integrity. The TOE uses a Cryptographic Module for signing key pair generation. As part of the signing key pair generation, Cryptographic Module provides the TOE with R.SigningKeyID and R.SVD. The TOE provides the R.SVD to the SSA for further handling for the key pair to be certified.

   Application Notes:

   - In [1], the following text stands for R.PubKey:
     public keys managed and used by the TOE in support of the cryptographic services that it offers (including user keys and support keys). This asset includes copies of keys made for external storage and/or backup purposes. The integrity of these keys must be protected.

   - The TOE does not keep R.SVD of a relevant signature key. A Signed R.SVD is replied as part of a signature key generation for the purpose of certificate generation by the CA.

5. **R.ClientData** (from [1]): data supplied by a client for use in a cryptographic function. Depending on the context, this data may require confidentiality and/or integrity protection.
6. **R.DTBS/R** (from [2]): set of data which is transmitted to the TOE for digital signature creation on behalf of the signer. The DTBS/R(s) is transmitted to the TOE. The R.DTBS/R shall be protected in integrity. The transmission of the DTBS/R(s) to the TOE shall require the sending party - Signer or Privileged User - to be authenticated.

   Application Notes:

   - In the context of [1], this may also be considered as a subclass of the above asset (R.ClientData).

   - From [2]: *The confidentiality of the R.DTBS/R is not required by Regulation (EU) No 910/2014 [18].*

7. **R.RAD** (from [1]): reference data held by the TOE that is used to authenticate an administrator (hence to control access to privileged administrator functions such as TOE backup, export of audit data) or to authorise a user for access to

secret and private keys (R.SecretKey). This asset includes copies of authentication/authorisation data made for external storage and/or backup purposes. The integrity of the RAD 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).

Application Notes:

- SAML authentication is based on a list of approved trusted RSA certificates or trusted RSA public keys.

8. **R.SAD** (from [2]):: signature activation data is a set of data involved in the signature activation protocol, which activates the signature creation data to create a digital signature under the signer's sole control. The R.SAD must combine:
   - The signer's strong authentication as specified in [23]
   - If a particular key is not implied (e.g a default or one-time key) a unique reference to R.SigningKeyID.
   - A given R.DTBS/R.

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

   Application Notes:
   - From [2]: *If the SAD does not require encrypted data then the confidentiality requirement is considered fulfilled. The ST writer shall describe which part of the SAD shall be protected in confidentiality* - The SAD does not require confidentiality.
   - From [2]: *The R.SAD may include some or all authentication factors or evidence from other systems that some or all authentication factors have been verified* - the SAML token that is created by the IDP or SSA and validated by the TOE.
   - From [2]: *The unique reference to R.SigningKeyID in the R.SAD could be certificate, key identifiers or derived information obtained from the signer's authentication.*
     Some solutions may use one-time signing keys, which are generated, certified and used within a limited signing session. The derived information from the signer's authentication may be used to provide session separation if a signer has multiple simultaneous signing sessions with the TOE, or to derive a R.SigningKeyID if the key is a one-time key. At the end of the session, the signing key is reliably deactivated.
     For solutions that only handle one signing key for each signer, the reference to the R.SigningKeyID may also be implied and omitted from the SAD.
     The ST writer shall describe what R.SigningKeyID is for a specific TOE – The TOE follows the guidance. A transaction ID is included in the SAD and can refer to a single R.SignerKeyID. It is also possible to have a R.SignerKeyID included in the SAD.

9. **R.Signature** (from [2]): is the result of the signature operation and is a digital signature value. R.Signature is created on the R.DTBS/R using R.SigningKeyID by the Cryptographic Module under the signer's control as part of the SAP. The R.Signature shall be protected in integrity. The R.Signature can be verified outside TOE using R.SVD.

10. **R.Audit** (from [2]): is audit records containing logs of events requiring to be audited. The logs are produced by the TOE and stored externally. The R.Audit shall be protected in integrity.

11. **R.Signer** (from [2]): is a TOE subject containing the set of data that uniquely identifies the signer within the TOE. The R.Signer shall be protected in integrity and confidentiality.

    Application Notes:

    - From [2]*: It is only within the TOE the R.Signer needs to be unique. It is not the responsibility of the TOE to establish a connection between the R.Signer and the signer's identity. The signer is said to own the R.Signer object which uniquely identifies him within the TOE* – This Note is satisfied by the TOE.

    - From [2]: *The R.Signer can include references to zero, one or several R.SigningKeyIDs and R.SVDs* –
      This Note is satisfied by the TOE

    - From [2]: *If the R.Signer does not require encrypted data then the confidentiality requirement is considered fulfilled. The ST writer shall describe which part of the R.Signer shall be protected in confidentiality.*
      - This note is satisfied by the TOE. There is no encrypted data required for R.Signer.

12. **R.Reference_Signer_Authentication_Data** (from [2]): is the set of data used by TOE to authenticate the signer. It contains all the data (e.g. OTP device serial number, phone numbers, protocol settings etc.) and keys (e.g. device keys, verification keys etc.) used by the TOE to authenticate the signer. This may include a SVD or certificate to verify an assertion provided as a result of delegated authentication. The R.Reference_Signer_Authentication_Data shall be protected in integrity and confidentiality

    Application Notes:

    - From [2]: *The R.ReferenceSignerAuthenticationData is used by the TOE to authenticate the signer, and the R.AuthorisationData is used by the TOE to activate a signing key in the Cryptographic Module* –
      This Note is satisfied by the TOE

    - From [2]: *If the R.ReferenceSignerAuthenticationData does not require encrypted data then the confidentiality requirement is considered fulfilled. The ST writer shall describe which part of the R.ReferenceSignerAuthenticationData shall be protected in confidentiality* - This note is satisfied by the TOE. There is no encrypted data required for R.ReferenceSignerAuthenticationData.

13. **R.TSFDATA** (from [2]): is the set of TOE configuration data used to operate the TOE. It shall be protected in integrity.

    Application Notes:
    - From [2]: *The TOE configuration data could include cryptographic algorithm, key length, flows for SAP etc* - As part of R.TSFDATA, the following information is

included: System Parameters, trusted RSA certificates and public keys.

14. **R.PrivilegedUser** (from [2]): is a TOE subject containing the set of data that uniquely identifies a Privileged User within the TOE. It shall be protected in integrity.
15. **R.ReferencePrivilegedUserAuthenticationData** (from [2]): is the set of data used by the TOE to authenticate the Privileged User. It shall be protected in integrity and confidentiality.

    Application Notes**:**

    - From [2]: *If the R.ReferencePrivilegedUserAuthenticationData does not require encrypted data then the confidentiality requirement is considered fulfilled. The ST writer shall describe which part of the R.ReferencePrivilegedUserAuthenticationData shall be protected in confidentiality* - The static password of the administrative role have a confidentiality requirement. The salted-hash of the static password is kept.

16. **R.Random** (from [2])**:** is random secrets, e.g. keys, used by the TOE to operate and communicate with external parties. It shall be protected in integrity and confidentiality.

3.2 Subjects

This following list of subjects interact with the TOE:

- Signer, which is the natural or legal person who uses the TOE through the SAP where he provides the SAD and can sign DTBS/R(s) using his signing key in the Cryptographic Module.
- Privileged User, which performs the administrative functions of the TOE and is able to provide a DTBS/R(s) to the TOE as part of the signature operation.

Application Notes:

- From [2]: *The list of subjects described in [23] clause 6.2.1.2 SRG M.1.2 contains more roles as it covers the whole T4WS. The ST writer shall describe the specific roles it implements and how these relate to authorisation rules in the SFRs.* – Described in this ST.
- From [2]: *In the case that SSA is used, 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. If the SSA as a service can perform administrative functions, e.g. creating signer, this is in this PP considered as Privileged User*
  - Described in this ST.
- From [2]: *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. [23]*
  – described in the ST. RA functionality can be done also using the TOE Client .

- Follows a formal representation of the TOE subjects:

Subjects	Definition	Subjects in the TOE
S.Admin [1]	An administrator of the TOE. Administrators are responsible for performing the TOE initialisation, TOE configuration and other TOE administrative functions.	*A Subject with R.Appliance Admin role.*
S.User-Admin [2]	Privileged User, which performs the administrative functions of the TOE and is able to provide a DTBS/R(s) to the TOE as part of the signature operation.	*A Subject with either R.Users Admin, R.SSA Admin or R.Appliance Admin role.*
S.User [1] or S.Signer [2]	*From [1]* - An end user of the TOE who can be associated with secret keys and authentication/authorisation data held by the TOE. An end user communicates with the TOE by using a browser (S.Application). *From [2]* – Which is the natural or legal person who uses the TOE through the SAP where he provides the SAD and can sign DTBS/R(s) using his signing key in the Cryptographic Module .	*A Subject with R.Signer role*
S.Application [1]	a client application, or process acting on behalf of a client application and that communicates with the TOE over a local or external interface. Client applications will in some situations be acting directly on behalf of end users (see S.User). Application Note: The TOE is both the CM and the SAM. Therefore the S.Application is the SAM based on a local interface.	*The SAM part of the TOE*

3.3 Threats

The following threats are defined for the TOE. An attacker described in each of the threats is a subject that is not authorised for the relevant operation but may present himself as an unknown user or as one of the other defined subjects.

Follows a formal list of the inspected threats:

T.KeyDisclose *Unauthorised disclosure of secret/private key (from [1])*

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

T.KeyDerive *Derivation of secret/private key (from [1])*

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

T.KeyMod *Unauthorised modification of a key (from [1])*

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

T.KeyMisuse *Misuse of a key (from [1])*

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

Remark: The threat includes unauthorised use of a cryptographic function that makes use of a key.

T.KeyOveruse *Overuse of a key (from [1])*
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 (from [1])*

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

Application note: The client applications (as defined in [2]) and cryptographic module (as defined in [1]) are implemented as one TOE. This threat relates to the interaction between the client application and the cryptographic module. Thus, this threat is not applicable.

T.DataMod *Unauthorised modification of client application data (from [1])*

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

Application note: The client applications (as defined in [2]) and cryptographic module (as defined in [1]) are implemented as one TOE. This threat relates to the interaction between the client application and the cryptographic module. Thus, this threat is not applicable.

T.Malfunction *Malfunction of TOE hardware or software (from [1])*

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

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

T.EnrolmentSignerImpresonation (From [2])

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

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

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

T.EnrollmentSIgnerAUthenicatinDataDisclosed (From [2])

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

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

The asset R.ReferenceSignerAuthenticationData is threatened.
Such data disclosure may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.
The threats on enrolment are threats on the environment in case external authentication is supported by the TOE.

T.SVDForgery (From [2])

An attacker modifies the R.SVD during transmission to the RA or CA. This results in loss of R.SVD integrity in the binding of R.SVD to signing key and to R.Signer.
The asset R.SVD is threatened.
If the CA relies on the generation of the key pair controlled by the TOE as specified in [22] clause 6.3.3 d) then an attacker can forge signatures masquerading as the signer.

Application Note:

- From [2]: *There should be a secure transport of R.SVD from TOE to RA or CA. The SAM is expected to produce a CSR.*
  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 [23] clause 6.3.1 a), this threat can be countered without any specific measures within the TOE.

T.AdminImpersonation (From [2])

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

T.MaintainanceAuthenticationDisclose (From [2])

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

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

This section describes threats for signature operation including authentication.

T.AuthenticationSignerImpersonation (From [2])

An attacker impersonates signer using forged R.Reference_Signer_Authentication_Data and transmits it to the TOE during SAP and uses it to sign the same or modified DTBS/R(s).
The assets R.Reference_Signer_Authentication_Data, R.SAD and R.Signing_Key_Id are threatened.

T.SignerAutherntictionDataModified (From [2])

An attacker is able to modify R.ReferenceSignerAuthenticationData inside the TOE or during maintenance.
The asset R.ReferenceSignerAuthentificationData is threatened.
Such data modification may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of signer.

T.SAPBypass (From [2])

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.SAPReplay (From [2])
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.SADForgery (From [2])
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.SignatureRequestDisclosure (From [2])
An attacker obtains knowledge of R.DTBS/R or R.SAD during transfer to TOE.
The assets R.DTBS/R and R.SAD are threatened.
If the R.DTBS/R or R.SAD do not require encrypted data then this threat is mitigated.

T.DTBSRForgery (From [2])

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

T.SignatureForgery (From [2])
An attacker modifies R.Signature during or after creation or during transfer outside the TOE.
The asset R.Signature is threatened.

Application Notes:
- From [2]: *The modification of a signature can be detected by the SSA or any relying party by validation of the signature.*

T.PriviledgedUserInsertion (From [2])
An attacker is able to create R.PrivilegedUser including R.ReferencePrivilegedUserAuthenticationData and is able to log on to the TOE as a Privileged User.
The assets R.PrivilegedUser and R.ReferencePrivilegedUserAuthenticationData are threatened.

T.ReferencePriviledgedUserAuthenticationDataModification (From [2])
An attacker modifies R.ReferencePrivilegedUserAuthenticationData and is able to log on to the TOE as the Privileged User.
The asset R.ReferencePrivilegedUserAuthenticationData is threatened.

T.AuthorizationDataUpdate (From [2])
Attacker impersonates Privileged User and updates R.Authorisation_Data and may be able to activate a signing key.
The assets R.AuthorisationData and R.SigningKeyID are threatened.

Application Notes:

- From [2]: *In some applications, it may be sufficient for an attacker with access to R.AuthorisationData and R.SigningKeyID to activate the signing key within the Cryptographic Module. Since the R.SigningKeyID is only to be protected in integrity and not in confidentiality, access to R.AuthorisationData should only be allowed for authorised operators* – as the TOE includes internally the cryptographic module, there is no direct access to the signing key.

T. AuthorisationDataDisclose (From [2])
Attacker discloses R.AuthorisationData during update and is able to activate a signing key.
The assets R.AuthorisationData and R.SigningKeyID are threatened.

T.ContextAlteraton (From [2])
An attacker modifies system configuration R.TSFDATA to perform an unauthorised operation.
The assets R.SigningKeyID, R.SVD, R.SAD, R.ReferenceSignerAuthenticationData and R.TSFDATA are threatened.

T.AuditAlteration (From [2])
An attacker modifies system audit and is able hide trace of TOE modification or usage.
The assets R.SVD, R.SAD, R.Signer, R.ReferenceSignerAuthenticationData, R.DTBS/R, R.Signature, R.AUDIT and R.TSF_DATA are threatened.

T.RANDOM (From [2])
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.

3.4 Organizational Security Policies

P.Algorithms or OSP.Crypto *Use of approved cryptographic algorithms (from [1] and [2])*

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

Application Notes:

- From [1]: *The relevant authorities and endorsements are determined by the TOE. For digital signatures within the European Union this is as indicated in [18] and an exemplary list of algorithms and parameters is given in [21] or [20] (see also section 4.4.1.4 in [1]).*

P.KeyControl *Support for control of keys (from [1])*

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

Application Notes**:**

- From [1]. *This policy is intended to ensure that the TOE can be used for qualified electronic seals and qualified electronic signatures as in [18], but recognises that not all keys are used for such purposes.*
  Therefore, although the TOE 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 (from [1] and [2])*

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

P.Audit *Audit trail generation (from [1])*
The TOE is required to generate an audit trail of security-relevant events, recording the event details and the subject associated with the event.

Application Notes:

- From [1]: *The cryptographic module TOE is assumed to be part of a larger system that manages audit data. The TOE therefore logs audit records, and it is assumed that these are collected, maintained and reviewed in the larger system. Hence there is no separate auditor role within the cryptographic module TOE, but the role of System Auditor is assumed to exist in the larger system – cf. A.AuditSupport in section 3.5.*

3.5 Assumptions

A.ExternalData *Protection of data outside TOE control (from [1])*

Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities must provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment.
In particular, any backups of the TOE and its data are maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data does not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data requires at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

A.Env *Protected operating environment (from [1])*

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

~~**A.DataContext** *Appropriate use of TOE functions (from [1])*~~

~~Any client application using the cryptographic functions of the TOE will ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and confidentiality). For example, when creating a digital signature over a DTBS the client application will ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and will correctly and securely manage the signature received from the TOE; and when certifying a public key the client application will ensure that necessary checks are made to prove possession of the corresponding private key. The client application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) performs a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS.~~
~~Client applications are also responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events.~~
~~Similar requirements apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys.~~

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

Application note:
Application note: The client applications (as defined in [2]) and cryptographic module (as defined in [1]) are implemented as one TOE. This threat relates to the interaction between the client application and the cryptographic module. Thus, this assumption is not applicable and is trivially satisfied.

~~A.UAuth~~ ~~*Authentication of application users (from [1])*~~

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

Application note:
Application note: The client applications (as defined in [2]) and cryptographic module (as defined in [1]) are implemented as one TOE. This threat relates to the interaction between the client application and the cryptographic module. Thus, this assumption is not applicable and is trivially satisfied.

A.AuditSupport *Audit data review (from [1])*

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

Application Notes:
- From [1]: *As noted for P.Audit in section 3.4, the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.*

~~A.AppSupport~~ ~~*Application security support (from [1])*~~

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

Application note:
The client applications (as defined in [2]) and cryptographic module (as defined in [1]) are

implemented as one TOE. This threat relates to the interaction between the client application and the cryptographic module. Thus, this assumption is not applicable and is trivially satisfied.

A.PriviledgedUser (From [2])

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

A.SignerEnrollment (From [2])

The signer shall be enrolled and certificates managed in conformance with the regulations given in [18]. Guidance for how to implement an enrolment and certificate management system in conformance with [18] are given in e.g. [22] or for qualified certificate in e.g. [24].

A.SignerAuthenticationDataProtection (From [2])
It is assumed that the signer will not disclose his authentication factors.

A.SignerDevice (From [2])
It is assumed that the device and SIC used by signer to interact with the SSA and the TOE is under the signer's control for the signature operation, i.e. protected against malicious code.

A.CA (From [2])
It is assumed that the qualified TSP that issues qualified certificates is compliant with the relevant requirements for qualified TSP's as defined in [18].

A.AccessProtected (From [2])
It is assumed that the TOE operates in a protected environment that limits physical access to the TOE to authorised Privileged Users. The TOE software and hardware environment (including client applications) is installed and maintained by Privileged Users in a secure state that mitigates against the specific risks applicable to the deployment environment.
It is assumed that any audit generated by the TOE are only handled by authorised personal in a physical secured environment. The personal that carries these activities should act under established practices.
It is assumed that where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities must provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment.

Application Notes:

- From [2]: *The ST writer shall describe which data is managed outside the TOE* - In general, all related information is kept inside the TOE.
  - This is fully described on this ST.

A.AUTHData (From [2])
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.TSPAudited (From [2])
It is assumed that the TSP deploying the SSA and TOE is a qualified TSP according to article 3 (20) of Regulation (EU) No 910/2014 [18] and audited to be compliant with the requirements for TSP's given by [18].

A.SecReq (From [2])
It is assumed that the TSP establishes an operating environment according to the security requirements for SCAL2 defined in [23].

4 Security Objectives

This section identifies and defines the security objectives for the TOE and the operational 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

OT.PlainKeyConf *Protection of confidentiality of plaintext secret keys (from [1])*

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

OT.Algorithms *Use of approved cryptographic algorithms (from [1] and [2] defined as OT.Crypto)*

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

Application Notes:

- From [1]: *See note under P.Algorithms on relevant references for digital signatures within the European Union* – followed by this ST.

OT.KeyIntegrity *Protection of integrity of keys (from [1])*

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

OT.Auth *Authorisation for use of TOE functions and data (from [1])*

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

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

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

Application Notes:

- From [1]: *Local client applications within a suitable security environment (such as client applications that are connected to the TOE by a channel such as a PCIe bus within the same hardware appliance) do not require authentication to communicate with the TOE, as noted in section 1.3.1. However, use of a secret key always requires prior authorisation*
  - not relevant to this TOE since the TOE is a composition of the Crypto Module and the SAM module.

OT.KeyUseConstraint *Constraints on use of keys (from [1])*

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

OT.KeyUseScope *Defined scope for use of a key after authorisation (from [1])*

The TOE is required to define and apply clearly stated limits on when authorisation and reauthorisation are required in order for a secret key to be used1. For example the TOE 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 TOE may implement a policy that requires re-authorisation before every use of a secret key.

Application Notes:

- From [1]: *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*
  - no re-authorisation is used by this TOE.

~~**OT.DataConf** *Protection of confidentiality of sensitive client application data (from [1])*~~

~~The TOE provides secure channels to client applications that can be used to protect the confidentiality of sensitive data (such as authentication/authorisation data) during transmission between the client application and the TOE, or during transmission between separate parts of the TOE where that transmission passes through an insecure environment.~~

~~Application Notes~~***:***

- ~~From [1]:. *Protection of secret keys (as a specific type of sensitive data) is also subject to additional protection specified in other TOE objectives. Any requirements for secure storage and control of access to other types of client application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport. For example, if a client application uses the TOE to perform cryptographic functions on data that represent a passphrase value and the passphrase value is to be stored on the TOE, then the client application would need to use an appropriate encryption function before storing the data on the TOE*~~
  – Since T.DataDisclose is not applicable, the OT.DataConf is not applicable .

~~**OT.DataMod** *Protection of integrity of client application data (from [1])*~~

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

~~Application Notes~~:

- ~~From [1]: *Any requirements for integrity protection of client application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport.*~~
  - Since T.DataMod is not applicable, the OT.DataMod is not applicable.

OT.ImportExport *Secure import and export of keys (from [1])*

The TOE allows import and export of secret keys only by using a secure method that protects the confidentiality and integrity of the data during transmission – in particular, secret keys 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 TOE also allows individual secret keys under its control to be identified as non-exportable, in which case any attempt to export them will be rejected automatically. Public keys may be imported and exported in a manner that protects the integrity of the data during transmission. Assigned keys cannot be imported or exported.

Application Notes:

- The server master keys are exported directly from the TOE during installation to dedicated USB tokens.
- The user keys are encrypted by a KEK Master Key (MK-KEK).

OT.Backup *Secure backup of user data (from [1])*

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

Application Notes:

- The backup does not include any key material. Only configuration information and administrative users accounts information is backed up.

OT.RNG *Random number quality (from [1] and [2] defined as OT.Random)*

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

OT.TamperDetect *Tamper Detection (from [1])*

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

OT.TamperResistance *Tamper resistance (from [1])*

The TOE prevents or resists physical tampering with specified system devices and components.

OT.FailureDetect *Detection of TOE hardware or software failures (from [1])*

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

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

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

OT.Audit *Generation of audit trail (from [1] and [2] – OT.AuditProtection)*

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

OT.SignerProtection (From [2])

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

OT.RefernceSignerAuthenticationData (From [2])

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

OT.SignerKeyPairGeneration (From [2])

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

OT.SVD (From [2])

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

OT.PriviledgedUserManagement (From [2])

The TOE shall ensure that any modification to R.PrivilegedUser and R.ReferencePrivilegedUserAuthenticationData are performed under control of a Privileged User.

OT.PriviledgedUserAuthentication (From [2])

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

Application Notes:

- From [2]: *The exception to this objective is when the initial (set of) Privileged Users are created as part of system initialization*
  – supported by the TOE. A single users administrator is created as part of the TOE installation.

OT.PriviledgedUserProtection (From [2])

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

OT.SignerManagement (From [2])

The TOE shall ensure that any modification to R.Signer, R.ReferenceSignerAuthenticationData, R.SigningKeyID and R.SVD are performed under control of the Signer or Privileged User.

OT.SADVerification (From [2])

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

Application Notes:

- From [2]: *Where the TOE 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*
  – as the TOE is a composition of the cryptographic modules and the SAM, an internal authorisation is built based on the SAD validation
- From [2] - *Requirements for authentication are described in [23] SRA_SAP.1.1* – followed by this ST

OT.SAP (From [2])

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

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

Application Notes:

- From [2]. *Signer authentication is assumed to be conducted according to [23] SCAL.2 for qualified signatures. This means signer authentication can be carried out in one of the following ways:*
    - *Directly by the SAM. In this case the SAM verifies the signer's authentication factor(s).*
    - *Indirectly by the SAM. 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.*
    - *A combination of the two directly or indirectly schemes.*
        - The TOE follows only the indirect authentication scheme.
        - The DTBS/R is deleted from the transaction table after the signature is collected by the SSA and in this way the solution is protected against a replay attack.

OT.SignatureAuthenticationDataProtection (From [2])

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

OT.DTBSRIntegrity (From [2])

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

OT.SignatureIntegrity (From [2])

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

OT.SystemProtection (From [2])

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

4.2 Security Objectives for the Operational Environment

OE.ExternalData *Protection of data outside TOE control (from [1])*

Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities shall provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment. This includes protection of data that is exported from, or imported to, the TOE (such as audit data and encrypted keys).
In particular, any backups of the TOE and its data shall be maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data shall not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data shall require at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

OE.Env *Protected operating environment (from [1] and [2])*

The TSP deploying the SSA and TOE shall be a qualified TSP according to article 3 (20) of Regulation (EU) No 910/2014 [18] and audited to be compliant with the requirements for TSP's given by [18]. The audit of the qualified TSP shall cover the security objectives for the operational environment specified in this clause.
The TOE shall operate in a protected environment that limits physical access to the TOE to authorised Administrators. The TOE software and hardware environment (including client applications) shall be installed and maintained by Administrators in a secure state that mitigates against the specific risks applicable to the deployment environment, including (where applicable):

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

OE.DataContext — *Appropriate use of TOE functions (from [1])*

Any client application using the cryptographic functions of the TOE shall ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and confidentiality). For example, when creating a digital signature over a DTBS the client application shall ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and shall correctly and securely manage the signature received from the TOE; and when certifying a public key the client application shall ensure that necessary checks are made to prove possession of the corresponding private key. The client application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) shall perform a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS.
Client applications shall be responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events.
Similar requirements shall apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys.
Appropriate procedures shall be defined for the initial creation of data and continuing operation of the TOE according to the specific risks applicable to the deployment environment and the ways in which the TOE is used.

Application Note:
Since the assumption A.DataContext is not applicable to the TOE, the corresponding Security Objective for the environment OE.DataContext is removed.

OE.Uauth — *Authentication of application users (from [1])*

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

Application Notes:

- There is no direct use of the Cryptographic module (according to [1]) by users, only through the defined in [2].

Since the assumption A.Uauth is not applicable to the TOE, the corresponding Security Objective for the environment OE.Uauth is not applicable.

OE.AuditSupport **Audit data review** *(from [1])*

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

Application Notes:

- From [1]: *As noted for P.Audit in section 3.4, the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.*

~~**OE.AppSupport** *Application security support (from [1])*~~

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

Application note:
Since the assumption A.AppSupport is not applicable to the TOE, the corresponding Security Objective for the environment OE.AppSupport is not applicable.

OE.SVDAuthenticity *(from [2])*

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

OE.CARequestCertificate *(from [2])*

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 [18].
The operational environment shall use a process for requesting a certificate, including SVD and signer information, and CA signature in a way, which demonstrates the signer is in control of the signing key associated with the SVD presented for certification. The integrity of the request shall be protected.

OE.CertificateVerification *(from [2])*

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

OE.SignerAuthenticationData *(from [2])*

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

OE.DelegatedAuthentication *(from [2])*

If the TOE has support for and is configured to use delegated authentication then the TSP deploying the SSA and TOE shall ensure that all requirements in [23] 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 [18], 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 [18]

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 [23] 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 Notes:

- The TOE only supports the indirect authentication mode using an IDP. The TOE has support for three modes of work with an IDP. These 3three modes are described in Figure 1.1, Figure 1.2 and Figure 2.

OE.Device (From [2])

The device, computer/tablet/smart phone containing the SIC and which is used by the signer to interact with the TOE shall be protected against malicious code. It shall participate using SIC as local part of the SAP and may calculate SAD as described in [23]. It may be used to view the document to be signed.

OE.CryptomoduleCertified (from [2])

If the TOE is implemented as a local application within the same physical boundary as the cryptographic module defined in [1] then the TOE relies on the

cryptographic module for providing a tamper-protected environment and for cryptographic functionality and random number generation.
If the TOE is implemented within a separate physical boundary then the TOE relies on the cryptographic module for cryptographic functionality and random number generation. The physical boundary shall physically protect the TOE conformant to FPT_PHP.1 and FPT_PHP.3 in [1].

Application Notes:

- From [2]: *In the case that the ST is conformant to this PP and to [1] as written in the PP Claim section, the certification of the ST covers this requirement for the Operational Environment* – This is the case of this TOE

OE.TW4Sconformant (from [2])

The TOE shall be operated by a qualified TSP in an operating environment conformant with [23].

4.3 Security Objective Rationale

There are no security objections introduced in this Security Target beyond the defined in [1] and [2].

The security objective rationale is identical to the rationales in both PPs except the removed security objectives due to the not applicable threats and assumptions in [1].

Security Objective rationale from [1]:

	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										

T.KeyMisuse				X	X															
T.KeyOveruse						X														
~~**T.DataDisclose**~~							~~X~~										~~X~~	~~X~~		
~~**T.DataMod**~~								~~X~~									~~X~~	~~X~~		
T.Malfunction													X							
P.Algorithms		X																		
P.KeyControl	X	X		X	X	X			X	X										
P.RNG											X									
P.Audit														X						
A.ExternalData															X					
A.Env																X				
~~**A.DataContext**~~																	~~X~~			
~~**A.AppSupport**~~																		~~X~~		
~~**A.UAuth**~~																			~~X~~	
A.AuditSupport																				X

Since the client applications (as defined in [2]) and cryptographic module (as defined in [1]) are implemented as one TOE, the interaction between the client applications and the TOE become internal communication inside the TOE. Thus the T.DataDisclose and T.DataMod that are aiming at the communication a channel between the client applications and cryptographic module become inapplicable. For the same reason, the assumptions A.DataContext, A.AppSupport and A.Uauth aiming at the client applications become inapplicable.

Therefore, as a consequence, the following corresponding security objectives should be removed:

- OT.DataConf
- OT.DataMod
- OE.DataContext
- OE.Uauth
- OE.AppSupport

Security Objective rationale from [2]:

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

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												
OSP.CRYPTO												X

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

T.CONTEXT_ALTERATION								X	
T.AUDIT_ALTERATION									X
T.RANDOM							X		
OSP.RANDOM							X		

	OE.SVD_AUTHENTICITY	**OE.CA_REQUEST_CERTIFICATE**	**OE.SIGNER_AUTHENTICATION_DATA**	**OE.DEVICE**	**OE.ENV**	**OE.CRYPTOMODULE_CERTIFIED**	**OE.TW4S_CONFORMANT**
Enrolment							
T.ENROLMENT_SIGNER_IMPERSONATION							X
T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED			X	X			
T.SVD_FORGERY	X	X					
Signer Management							
T.ADMIN_IMPERSONATION							
T.MAINTENANCE_AUTHENTICATION_DISCLOSE							
Usage							
T.AUTHENTICATION_SIGNER_IMPERSONATION							
T.SIGNER_AUTHENTICATION_DATA_MODIFIED							
T.SAP_BYPASS				X			
T.SAP_REPLAY				X			
T.SAD_FORGERY			X	X			
T.SIGNATURE_REQUEST_DISCLOSURE							
T.DTBSR_FORGERY				X			
T.SIGNATURE_FORGERY							
System							
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							
Organisational Security Policies							
OSP.TSP_AUDITED							X
OSP.RANDOM							
OSP.CRYPTO						X	
Assumptions							
A.PRIVILEGED_USER							X
A.SIGNER_ENROLMENT					X		

A.SIGNER_AUTHENTICATION_DATA_PROTECTION			X				
A.SIGNATURE_REQUEST_DISCLOSURE				X			
A.SIGNER_DEVICE				X			
A.CA		X					
A.ACCESS_PROTECTED					X		
A.AUTH_DATA				X			
A.TSP_AUDITED					X		
A.SEC_REQ							X

Therefore, this section is covered in [1] and [2].

5 Extended Components Definitions

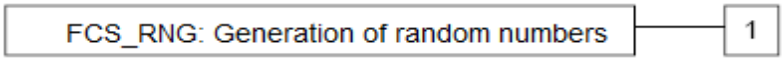
5.1 Generation of random numbers (FCS_RNG) (from [1] and [2])

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

Family behavior

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

Component leveling:



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 Notes:

- From [1] and [2]: *A physical random number generator (RNG) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (key strokes, mouse*

movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs where a hybrid physical RNG produces at least the amount of entropy the RNG output may contain and the internal state of a hybrid deterministic RNG output contains fresh entropy but less than the output of RNG may contain.

5.2 Basic TSF Self Testing (FPT_TST_EXT.1) (from [1])

The extended component defined here is a simplified version of FPT_TST.1 in [4]

Family behaviour

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

Component levelling:

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

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

- Indication that TSF self test was completed.

FPT_TST_EXT.1 *Basic TSF Self Testing*

Hierarchical to: No other components.

Dependencies: No dependencies.

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

6 Security Requirements

This chapter gives the security functional requirements and the security assurance requirements for the TOE and the environment.

Security functional requirements components given in section 6.1 *security functional requirements* are drawn from Common Criteria part 2 [4]. Operations for assignment, selection, iteration and refinement have been made.

The TOE security assurance requirements statement given in section 6.2 “TOE Security Assurance Requirement” is drawn from the security assurance components from Common Criteria part 3 [5].

The following textual conventions are used in this chapter as part of every SFR:

- Iteration
  Allows a component to be used more than once with varying operations. A slash (“/”) followed by an identifier placed at the end of the component indicates an iteration. In the case of a reference to a iteration or a group of the same iteration, the reference will be to the group of the iterations. For example, iterations FDP_ACF.1.1/Signer-Creation SFP, FDP_ACF.1.2/Signer-Creation SFP will be referred as FDP_ACF.1/Signer-Creation SFP.
- Assignment
  Allows the specification of an identified parameter and it is represented in *Italic* and underlined**.**
- Selection:
  Allows the specification of one or more elements from a list and it is represented in *italic* and underlined.
- Refinement:
  Allows the addition of details, that are represented in **bold** and underlined**.**

6.1 Security Functional Requirements

6.1.1 Security Audit (FAU)

6.1.1.1Security audit data generation (FAU_GEN)

6.1.1.1.1 Audit data generation (FAU_GEN.1) (from [1] and [2])

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the *not specified* level of audit; and
c) *Startup of the TOE;*
d) *Shutdown of the TOE*;
e) *Cryptographic key generation (FCS_CKM.1);* or *Signing Key Generation;*
f) *Cryptographic key destruction (FCS_CKM.4);* or *Signing Key Destruction;*
g) *Failure of the random number generator (FCS_RND.1);*
h) Privileged User Management;
i) Privileged User Authentication;
j) Signer Management;
k) Signer Authentication;
l) Signing Key Activation and Usage including the Hash of the DTBS/R(s) and R.Signature;
m) 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;
n) All attempts to import or export keys (FDP_IFF.1/KeyBasics);
o) All modifications to attributes of keys (FDP_ACF.1/KeyUsage, FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys);
p) Backup and restore (FDP_ACF.1/Backup): use of any backup function, use of any restore function, unsuccessful restore because of detection of modification of the backup data;
q) Integrity errors detected for keys (FDP_SDI.2);
r) Failures to establish secure channels (FTP_TRP.1/Local, FTP_TRP.1/External);
s) Self-test completion (FPT_TST_EXT.1);
t) Failures detected by the TOE (FPT_FLS.1);
u) All administrative actions (FMT_SMF.1, FMT_MSA.1 (all iterations), FMT_MSA.3/Keys);

v) Unblocking of access (FMT_MTD.1/Unblock);
w) Modifications to audit parameters (affecting the content of the audit log) (FAU_GEN.1)
x) Change of TOE Configuration;
y) Admin Change Password,
z) Reset Tamper;
aa) Tamper Detection;
bb) Upload Software Version;

Application Notes:

- From [2]: *Management of R.PrivilegedUser and R.Signer objects shall include all events, which creates, modifies or deletes the R.Signer or R.PrivilegedUser objects.*
  – This note is satisfied by the TOE.
- From [2]: *Signer authentication shall include failed verification of an assertion provided by a delegated party*
  - This note is satisfied by the TOE.
- From [2]. *TOE configuration shall include all events, which creates, modifies and deletes the configuration object*
  - This note is satisfied by the TOE
- From [2]: *Generation of a certification request is usage of the signing key and mandates an audit trail*
  - This note is satisfied by the TOE
- From [2]: *Some implementations may not, for privacy reasons, record the R.DTBS/R in the audit log. For such systems, the ST writer shall describe how the log can be used to demonstrate that particular DTBS/R(s) was signed*
  – The TOE records the DTBS.R in the audit log.
- The events *to import or export keys (FDP_IFF.1/KeyBasics);*
    - As the blob of binding of Signature keys to users is stored outside the TOE, the event of storing outside the blob or event of reading the blob into the TOE will be reported to the audit log
    - Master keys are imported to the TOE upon the following cases:
        - A TOE installation done in a secure environment
        - An administrative reset tamper event done in a secure environment
- The event of *Unblocking of access (FMT_MTD.1/Unblock);* is not relevant and thus will not be audited
- From [1]: *The Security Target is not required to identify separate levels of audit in FAU_GEN.1.1. However, the Operational Guidance is required to describe any configuration or other actions that apply to audit functions, and to make clear, in cases where logging of particular audit events is optional, how to ensure that any individual audit event is logged. Default logging actions of the TOE shall also be described in Operational Guidance.*
  The Administrative Actions logged need not be limited to those related to FMT SFRs: other administrative actions affecting the operation of SFRs should also be included (and listed as part of the assignment in FAU_GEN.1.1)
  – The TOE does not provide mechanisms to define levels in the audit log

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *Type of action performed (success or failure*), *identity of the role performed the operation and None*.

Application Notes:

- From [2]. *Audit trail shall not include any data which allow to retrieve sensitive data like R.SAD, R.ReferenceSignerAuthenticationData and R.AuthorisationData*
  – This note is satisfied by the TOE

6.1.1.1.2 User identity association (FAU_GEN.2) (from [1] and [2])

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

6.1.1.1.3 Guaranties of Audit Data Availability (FAU_STG.2) (from [1])

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

FAU_STG.2.2 The TSF shall be able to *prevent* unauthorised modifications to the stored audit records in the audit trail.

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

Application Notes:

- From [1]: *The Operational Guidance is required to describe any use that the TOE makes of an external audit server, the situation regarding records held locally on the TOE and those held externally on an audit server (e.g. the TOE might accumulate records locally before transferring them to an external audit server), and the way in which audit records are maintained when local audit storage is exhausted (including description of the actions taken by the TOE when audit storage exhaustion is detected). The Operational Guidance shall describe the protection applicable to all records created by the TOE (in order to provide prevention or detection of unauthorised modifications as in FAU_STG.2.2) and shall identify any obligations for the environment in maintaining audit trail protection. The expectation is that this will comprise cryptographic methods of prevention or detection of unauthorised modification (including deletion) of audit records.*

Control over export and deletion of the audit log records is limited to the Administrator role as specified in FMT_MTD.1/AuditLog.
– The TOE collects audit logs initiate either from the Crypto Module (According to [1]) or the SAM module (according to [2]). These logs are accumulated and sent to an external Audit Log Server.

6.1.2 Cryptographic support (FCS)

6.1.2.1Cryptographic key management (FCS_CKM)

6.1.2.1.1 Cryptographic key generation (FCS_CKM.1) (from [1] and [2])

FCS_CKM.1.1/SIGNATURE-KEY
The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *RSA* and specified cryptographic key sizes *2048, 3072 and 4096 Bit* that meet the following: *[10], [6], and [9]***.**

FCS_CKM.1.1/SYMMETRIC-KEY
The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *AES* and specified cryptographic key sizes *256 bit* that meet the following *[20],[21]***.**

Application Notes:

- From [1]: *The Security Target shall include all key generation operations that are intended to support TSP operations using one or more iterations of FCS_CKM.1.*
  The relevant authorities and endorsements for completion of the SFRs are determined by the context of the client applications that use the TOE. For digital signatures within the European Union this is as indicated in Regulation (EU) 910/2014 [8] and an exemplary list of algorithms and parameters is given in ETSI/TS 119 312 [21] or SOG-IS-Crypto [20] (see also 4.4.1.4 in [1]).
  Note that key generation needs to be linked to the setting of security attributes of a key (including the link to a subject who owns the key, via the setting of authorization data) as in FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys
  - this note is satisfied by the TOE.
- From [2]. *The TOE is expected to use a cryptographic module certified in conformance with [1], see also OE.CryptoModuleCertified for key generation. Although the TSF may not generate keys itself, this SFR expresses the requirement for the TSF to invoke the cryptographic module with the appropriate parameters whenever key generation is required. Guidance on cryptographic algorithms can be found in [21] and [20]*
  – this note is satisfied by the TOE, the TOE includes the cryptographic module
- From [2]: *The ST is expected to use cryptographic keys for different purposes, e.g. application, infrastructure, session etc. The ST writer should include an iteration of this SFR*

for every key type (e.g. RSA and AES) it generates itself
– this note is satisfied by the TOE

6.1.2.1.2 Cryptographic key destruction (FCS_CKM.4) (from [1] and [2])

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *zeroization* that meets the following: *FIPS 140-2, section 4.7.6*

Application Notes:

- From [1] and [2]: *The Security Target shall specify the method(s) of secure destruction of all secret keys and all support keys, and shall ensure that all are covered by a secure destruction method. If necessary then more than one iteration of FCS_CKM.4 may be included to describe different standards for secure deletion. The 'list of standards' in the final assignment may be met in the Security Target by simply providing a description of the action taken to zeroise the keys rather than referencing an external standard*
  – this note is satisfied by the TOE
- From [2]: *The TOE is expected to use a cryptographic module certified in conformance with [1] for key destruction.*
  Although the TSF may not destruct keys, this SFR expresses the requirement for the TSF to invoke the cryptographic module with the appropriate parameters whenever key destruction is required.
  - this note is satisfied by the TOE. The includes the cryptographic module
- Assigned Signature keys are destroyed from memory right after usage.
- Assigned Signature keys are kept encrypted in the QSCD and in an external storage. The encrypted keys that are kept inside the QSCD are either deleted using a key instruction or deleting a user instruction.
- Master keys are destroyed from memory upon shutdown to the TOE or setting the TOE to factory state.

6.1.2.2Cryptographic operation (FCS_COP)

6.1.2.2.1 Cryptographic operation (FCS_COP.1) (from [1] and [2])

FCS_COP.1.1/SIGNING
The TSF shall perform *digital signature-generation* in accordance with a specified cryptographic algorithm *RSA* and cryptographic key sizes *2048 bit, 3072 bit and 4096 bit* that meet the following: *[10] and [6]*.

FCS_COP.1.1/DATA-INTEG

The TSF shall perform H*MAC Calculation and Verification* in accordance with a specified cryptographic algorithm *HMAC-SHA256* and cryptographic key sizes *256 bit* that meet the following: *[7]*.

FCS_COP.1.1/KEY-ENCRYPTION

The TSF shall perform *Symmetric Key Encryption* in accordance with a specified cryptographic algorithm *AES256* and cryptographic key sizes *256 bit* that meet the following: *[20],[21]*.

FCS_COP.1.1/BKP-DATA-INTEG

The TSF shall perform H*MAC Calculation and Verification* in accordance with a specified cryptographic algorithm *HMAC-SHA256* and cryptographic key sizes *256 bit* that meet the following: *[7]*.

FCS_COP.1.1/BKP-ENCRYPTION

The TSF shall perform *Symmetric Encryption* in accordance with a specified cryptographic algorithm *AES256* and cryptographic key sizes *256 bit* that meet the following: *[20],[21]*.

FCS_COP.1.1/ADMIN-SESSION-DATA-INTEG

The TSF shall perform H*MAC Calculation and Verification* in accordance with a specified cryptographic algorithm *HMAC-SHA256* and cryptographic key sizes *256 bit* that meet the following: *[7]*.

FCS_COP.1.1/FIRM-UPD

The TSF shall perform *digital signature-validation* in accordance with *RSA* and cryptographic key sizes *2048 bit* that meet the following: *[10] and [6]*

FCS_COP.1.1/TLS-SESSION

The TSF shall perform *Asymmetric Cryptography, Symmetric Cryptography, HMAC Cryptography* in accordance with *RSA*, AES, HMAC and cryptographic key sizes *2048(RSA), 128/256(AES), HMAC-SHA256 bit* that meet the following: *[10], [7] and [6]*

Application Notes:

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

TOE supports software or firmware updates then the iterations shall include the cryptographic operations used to support the validation of digital signatures on the updates as described in the refinement to ADV_ARC.1 in 9.5.2 of [1].
The relevant authorities and endorsements for completion of each of these iterations are determined by the context of the client applications that use the TOE. For digital signatures and seals within the European Union this is as indicated in Regulation (EU) 910/2014 [8] and an exemplary list of algorithms and parameters is given in ETSI/TS 119 312 [21] or SOG-IS-Crypto [20]
– This note is satisfied by the TOE

- From [2].: *The TOE is expected to use a cryptographic module certified in conformance with [1] for cryptographic operations*
  – The TOE includes a cryptographic module

- From [2]: *The relevant authorities and endorsements for completion of the SFRs are determined by the context of the client applications that use the TOE. For digital signatures within the European Union, this is as indicated in Regulation (EU) No 910/2014 [18] and a list of approved signature and seal formats are given in [26]*
  – – This note is satisfied by the TOE

- Successful data integrity calculations checks, as well as key encryption/decryption operations will not be audited.

6.1.2.3 Generation of random numbers (FCS_RNG)

6.1.2.3.1 Generation of random numbers (FCS_RNG.1) (from [1] and [2])

FCS_RNG.1.1 The TSF shall provide a *hybrid deterministic* random number generator that implements:
physical: Random Sequence Generator based on true quantum randomness from the shot noise of a light source captured by a CMOS image sensor,
deterministic: HMAC-DRBG – NIST SP 800-90*A [16]*.

FCS_RNG.1.2 The TSF shall provide *octets of bits* that meet *Estimated entropy of 6.0*.

Application Notes:

- From [1]: *For more information on the selections and assignments see the SFR definition in 8.1 of [1].*
  The Security Target describes the uses made of the RNG and its relationship to other SFRs such as FCS_CKM.1, and to any random number generation function/service made available to users or clients applications
  - This note is satisfied by the TOE
- From [2]: *For more information on the selections and assignments, see the SFR definition in section* ***Error! Reference source not found.*** *in [2].*

From [2]. *The SFR FCS_RNG.1 only apply, if the TOE is not implemented as a local application within the same physical boundary as the cryptographic module – otherwise, the SFRs defined in [1] already provide requirements on generation of random numbers. This should be stated in the Security Target*
– Since the SAM part of the TOE is implemented as a local Application within the same physical boundary of the CM, then this SFR is based on [1]

6.1.3 User data protection (FDP)

6.1.3.1 Access Control Policy (FDP_ACC)

6.1.3.1.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1/Privileged-User-Creation (from [2])

The TSF shall enforce the *Privileged-User-Creation SFP* based on the following:

Subjects: Privileged User Administrator,

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

The TOE creates R.PrivilegedUser and R.Reference_Privileged_User_Authentication_Data with information transmitted by Privileged User.

Application Notes*:*

- From [2]: *The ST writer shall describe how the initial Privileged User is created and if there are additional requirements for quorum of Privileged User to create a new Privileged User* – The initial users Administrator and Appliance Administrator are created as part of the TOE installation.
- The Privileged User that creates the new account (ie the Subject) is a user with the role of Users Administrator
- The newly created Privileged User account (ie the Object) can be either an Appliance Administrator, a Users Administrator or an SSA Admin (Users Administrators that can create only signers)
- A single Users Administrator can create a Privileged user.
- Same authorizations are used for deleting or maintaining other privileged users.

FDP_ACC.1.1/Signer-Creation (from [2])

The TSF shall enforce the Signer-Creation SFP based on the following:

Subjects: Privileged User Administrator,

Objects: R.Signer and R.ReferenceSignerAuthenticationData,

Operations: CreateNewSigner

The TOE creates R.Signer and R.ReferenceAuthenticationData with information transmitted by the Privileged User

Application Notes*:*

- Since the Signer gets authenticated based on a SAML ticket validation, the Referenced-Authentication-Data is based on a list of trusted certificates or trusted public keys that are used for validating the SAML token. The list is part of the TOE configuration and is updated by the Appliance Administrator.
- The Privileged User is a user with the role of SSA Admin

FDP_ACC.1.1/Signer-Maintenance (from [2])

The TSF shall enforce the *Signer-Maintenance SFP* on
Subjects: Privileged Users and Signers

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

Operations: SignerMaintenance:

The Privileged User or Signer instructs the TOE to update R.ReferenceSignerAuthenticationData of R.Signer

Application Notes*:*

- Privileged user is an admin user with the role of SSA Admin
- The Signer Maintenance mechanisms mainly used for deletion a key or a signer or adding a new key to a signer via the key generation operation.

FDP_ACC.1.1/Signer-Key-Pair-Generation (from [2])

The TSF shall enforce the *Signer-Key-Pair-Generation SFP* on:
Subjects: Privileged User and Signer.
Objects: The security attributes R.SVD and R.SigningKeyID as part of R.Signer.
Operations: GenerateSignerKeyPair:
The Privileged User or Signer instructs the TOE to request the Cryptographic Module to generate a signing key pair R.SigningKeyID and R.SVD and assign them to the R.Signer

Application Notes:

- From [2]. *The ST writer shall describe how R.AuthorisationData is established*
  – After the SAD is validated the authorization data is established permitting a single and volatile access only to the keys that the user is mapped to (ie assigned keys)

- From [2]. *The ST writer shall describe if signing keys can be used by several cryptographic modules and how the keys are protected outside the module, including a description of how the association to R.Signer and R.AuthorisationData are maintained. See FDP_UCT.1*
  – The Signer's keys blob is maintained outside the TOE and can be used by several QSCDs for high availability purposes, providing that all the QSCDs are installed with the same Critical Keys set.
  The Signer Keys blob is protected in Integrity using the HMAC key secret, which is the *Appliance Backup Integrity Secret*. The signer keys are encrypted using the *Appliance Backup encryption key.* This way the authorization of the signer for using his/her keys is maintained.

- From [2]: *Signing keys may be generated by the Cryptographic Module in advance, as so called pre-generated keys, in order to improve performance. If the TOE uses pre-generated keys, the ST writer shall describe how these are protected before they are assigned to a Signer*
  – This is a configurable option of the TOE. All pre-generated keys are encrypted with the *MK-KEK* and MACed with *MK-MAC*

- From [2]. *The environment shall ensure if needed any transformation of R.SVD to a certification request and transport to CA*.- After the signature key is generated, a PKCS#10 certificate request is created based on a volatile public key that is calculated based on the private key.

- The SSA Admin authorises the Key Pair Generation for the Signer. The newly generated key will be bound to the signer account, and only the Signer will be able to use the newly generated key.
- The newly generated key will be stored externally to the TOE in a special blob where the keys are bounded to the signer in a protected manner.

FDP_ACC.1.1/Signer-Key-Pair-Deletion (from [2])

The TSF shall enforce the *Signer-Key-Pair-Generation SFP* on:

Subjects: Privileged User and Signer.

Objects: The security attributes R.SVD and R.SigningKeyID as part of R.Signer.

Operations: SignerKeyPairDeletion

The Privileged User or Signer instructs the TOE to request the Cryptographic Module to delete the signing key pair R.SigningKeyID and R.SVD from R.Signer

Application Notes:

- From [2]. *Deletion of R.SigningKeyID may also require that the signing key is deleted by the Cryptographic Module*
  – This note is satisfied by the TOE
  This SFR is limited to covering deletion of the R.SigningKeyID and R.SVD of R.Signer performed using one of the interfaces provided by the TOE and where authorisation to perform operations is managed by TOE.
  - As this TOE includes both the CM and SAM, the deletion of the signing key involves both meta information as well as the encrypted key value itself, which are all deleted as part of this action

- In the case of signer account deletion, his/her signing keys and *R.SigningKeyIDs* will be deleted as well.
- The SSA Admin is authorised to either deleted a specific key or delete the whole account, which will *delete R.SigningKeyID*.

FDP_ACC.1.1/Supply-DTBS/R (from [2])

The TSF shall enforce the *Supply-DTBS/R SFP* on:
Subjects: Privileged User.

Objects: The security attributes R.DTBS/R of R.Signer.

Operations: SupplyDTBSR

The Privileged User instructs the TOE to link the supplied DTBS/R(s) to the next signature operation for R.Signer

Application Notes:

- From [2]. *If the TOE does not provide facilities to supply the DTBS/R(s) then the relevant part of the SFR is trivially satisfied, and this should be stated in the ST.*
  - The TOE will reply with a random transaction ID after receiving the DTBS/R by the SSA Admin user when called by the SSA. The transaction ID will be passed as part of the SAD, thus enabling the TOE including the right DTBS/R as part of the signature operation

FDP_ACC.1.1/Signing (from [2])

The TSF shall enforce the *Signing SFP* on:

Subjects: Signer.

Objects: R.Authorisation_Data, security attributes R.Signing_Key_Id and R.DTBS/R of R.Signer and R.Signature.

Operations: *Signing*

The Signer instructs the TOE to perform a signature operation containing the following steps:
- The TOE establishes R.AuthorisationData for the

R.SigningKeyID.
- The TOE uses the R.AuthorisationData, and R.SigningKeyID to activate a signing key in the Cryptographic Module and signs the R.DTBS/R resulting in R.Signature.
- The TOE deactivates the signing key when the signature operation is completed

Application Notes:

- From [2]: *The ST writer shall describe how R.AuthorisationData is used to activate signing keys in the Cryptographic Module*
  – This is described below in an application note.

- From [2]: *The ST writer shall describe how the DTBS/R(s) is supplied to the TOE. It can be either in this function or using FDP_ACC.1/Supply DTBS/R*
  – This note is satisfied by the TOE. The DTBS/R is supplied through the SSA interface.

- From [2]: *Signing key deactivating means that the signer shall authorise any subsequent use of it*
  – This note is satisfied by the TOE.
- When the DTBS/R is supplied by the SSA, it will be kept for this user and a Transaction ID will be returned. The transaction ID is randomly generated by the TOE and will be provided to this operation by the SSA. The TOE will check that a proper DTBS/R is bound to the given transaction ID. The replied digital signature will be collected by the SSA.
- The SSA will supply the SAML Token signed either IDP that has verified the Signer's factors. Only after proper validation of SAML Token, the TOE authorises access to the signer's key according to the supplied R.SigningKeyID. Only for the specific usage, the signing key will perform the digital signature operation and reply with a digital signature.
  The transaction ID points to a specific key identified that belongs to the signer.

FDP_ACC.1.1/TOE-Maintenance

The TSF shall enforce the TOE-Maintenance *SFP* on:

Subject: Privileged User

Object: R.TSF_DATA

Operation: TOE_Maintenance

The Privileged User transmits information to the TOE to manage R.TSF_DATA

Application Notes:

- Maintenance operations are performed either using the TOE Administrative client deployment

FDP_ACC.1.1/Backup (from [1])

The TSF shall enforce the *Backup SFP* on

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

Application Notes:

- The backup will include only configuration information and privileged users information. This means that no key material will be kept in the backup.
  In this sense, this SFR is trivially satisfied.

FDP_ACC.1.1/KeyUsage (from [1])

The TSF shall enforce the *KeyUsage SFP* on

1. Subjects = *all*
2. Objects = keys
3. Operations = all

6.1.3.2 Access Control Functions (FDP_ACF)

6.1.3.2.1 Security attribute based access control (FDP_ACF.1)

The security attributes for the user, TOE components and related status are.

User, subject or object the attribute is associated with	Attribute	Status
User account	Role	Appliance Administrator (R.ApplianceAdmin), Users Administrator (R.UserAdmin), SSA Admin (R.SSA), Signer (R.Signer)
User account	Data integrity	yes, no
User account	Creation status	created, not created
User account	Lock status	locked, unlocked
Appliance Administrator, Users Administrator, SSA Admin	Static password RAD	value, empty
SCD	SCD status	init, operational, not operational

Table 1 - Security attributes for ACFs

Privileged-User-Creation (from [2])

FDP_ACF.1.1/Privileged-User-Creation

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

1. *Whether the subject is a Privileged User authorised to create a new Privileged User*.

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 CreateNewPrivilegedUser operation*

FDP_ACF.1.3/Privileged-User-Creation

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *None*.

FDP_ACF.1.4/Privileged-User-Creation

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *None*.

Application Notes:

- The Privileged user is a user with either a Users Admin role, Appliance Admin role or SSA Admin role
- When a static password is used to authenticate the newly created privileged user, the static password must be not empty and satisfies the password policy configuration. The password policy configuration requires minimal password length of 8 characters.
- Same authorizations are used for deleting or maintaining other privileged users.

Signer-Creation (from [2])

FDP_ACF.1.1/Signer-Creation

The TSF shall enforce the *Signer-Creation SFP* to objects based on the following**:**

1. *whether the subject is a Privileged User authorised to create a new Signer*.

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 CreateNewSigner operation*

Application Notes:

- The Privileged user is a user with SSA Admin role

FDP_ACF.1.3/Signer-Creation

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *None*.

FDP_ACF.1.4/Signer-Creation

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *None*.

Signer-Maintenance (from [2])

FDP_ACF.1.1/Signer-Maintenance

The TSF shall enforce the *Signer-Maintenance SFP* to objects based on the following**:**

1. *Whether the subject is a Privileged User or Signer authorised to maintain the Signer security attributes*.

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 SignerMaintenance operation*

FDP_ACF.1.3/Signer-Maintenance

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

1. *The Signer must be the owner of the R.Signer object to be maintained*

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*

Application Notes:

- From [2]. *The ST writer shall describe if R.ReferenceSignerAuthenticationData can be maintained by both Privileged User and Signer* – The ReferenceSignerAuthenticationData for signers is a trusted list of certificates and public keys that are used for the SAML validation. The trusted list is managed by the Appliance Administrator and handled as part of the TOE-Maintenance.
- The Signer-Maintenance handles adding a new key to the signer-keys blob or removing keys from the signer-keys blob as part of the key deletion operation.

Signer-Key-Pair-Generation (from [2])

FDP_ACF.1.1/Signer-Key-Pair-Generation

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

1. *whether the subject is a Privileged User or Signer authorised to generate a key pair*.

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 GenerateSignerKeyPair operation*

FDP_ACF.1.3/Signer-Key-Pair-Generation

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

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

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*

Application Notes:

- From [2]. *If pre-generated keys are used then FDP_ACF.1.4/Signer Key Pair Generation shall prevent assigning an already assigned key pair to the R.Signer object*
  – This note is satisfied by the TOE. The Appliance Administrator can configure the TOE to use pre-generated keys or having the signature key be generated upon the signature key generation request.
- From [2]. *Owning a R.Signer object is described in FIA_UAU.5/Signer.*

Signer-Key-Pair-Deletion (from [2])

FDP_ACF.1.1/Signer-Key-Pair-Deletion

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

1. *whether the subject is a Privileged User or Signer authorised to delete the Signer security attributes*.

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 SignerKeyPairDeletion operation*

FDP_ACF.1.3/Signer-Key-Pair-Deletion

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

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

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*

Supply-DTBS/R (from [2])

FDP_ACF.1.1/Supply-DTBS/R

The TSF shall enforce the *Supply-DTBS/R SFP* to objects based on the following**:**

1. *Whether the subject is a Privileged User authorised to supply a DTBS/R(s*)

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 who has been authorised to supply a DTBS/R(s) can carry out the Supply_DTBS/R operation*

FDP_ACF.1.3/Supply-DTBS/R

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *None*

FDP_ACF.1.4/Supply-DTBS/R

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *None*

Application Notes:

- From [2]: *If the TOE does not provide facilities to supply the DTBS/R(s) then the relevant part of the SFR is trivially satisfied, and this should be stated in the ST.*
  - This option is used by the TOE.

Signing

FDP_ACF.1.1/Signing

The TSF shall enforce the *Signing SFP* to objects based on the following**:**

1. *Whether the subject is a Signer authorised to create a signature*

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.SigningKeyID as bound in the SAD, and which is part of the R.Signer security attributes, can be used to create a signature*

FDP_ACF.1.3/Signing

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *None*

FDP_ACF.1.4/Signing

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *None*

Application Notes:

- From [2]. *In FDP_ACF.1.2/Signing the R.SigningKeyID can be implied if the signing uses a one-time keys or a signing key is known to be the default*
  – It is possible that the R.SigningKeyID will not be specified, but in the case that the signer has more than a single signature key and no specific signing key ID was specified, the signature operation will fail.

TOE-Maintenance

FDP_ACF.1.1/TOE-Maintenance

The TSF shall enforce the *TOE-Maintenance SFP* to objects based on the following:

1. *Whether the subject is a Privileged User authorised to maintain the TOE configuration data*.

FDP_ACF.1.2/TOE-Maintenance

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

1. *Only a Privileged User who has been authorised to maintain the TOE can carry out the TOE_Maintenance operation*

FDP_ACF.1.3/TOE-Maintenance

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *None*

FDP_ACF.1.4/TOE-Maintenance

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *None*

Backup SFP (From [1])

FDP_ACF.1.1/Backup SFP

The TSF shall enforce the *Backup SFP* to objects based on the following:

1. *whether the subject is an administrator*

FDP_ACF.1.2/Backup SFP

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

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

FDP_ACF.1.3/Backup SFP

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*.

FDP_ACF.1.4/Backup SFP

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

Application Notes:

- From [1]. *Preserving the binding of a set of attributes to its key (in FDP_ACF.1.2/Backup (4)) means that it is not possible for the attributes to be changed during a backup operation, or by modification of the backup data while it is away from the TSF.*
  Backups may contain keys whose export flag attribute marks them as 'non-exportable'. The ST author specifies the cryptographic operations used to protect confidentiality and integrity of any supported backups using one or more iterations of FCS_COP.1 - The backup operation does not include any signature keys and signers' information. Therefore, this requirement is trivially satisfied.
- From [1]. *If the TOE does not provide backup and restore operations then the Security Target shall include FDP_ACC.1/Backup and FDP_ACF.1/Backup but shall state in an Application Note for each of these SFRs that the relevant security requirements are trivially met because no backup facility is provided* - The backup operation does not include any signature keys and signers' information. Therefore, this requirement is trivially satisfied.
- The administrator in *FDP_ACF.1.1/Backup SFP* is the Appliance Administrator

KeyUsage SFP (From [1])

FDP_ACF.1.1/KeyUsage

The TSF shall enforce the *KeyUsage SFP* 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*

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*

FDP_ACF.1.3/KeyUsage

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*.

FDP_ACF.1.4/KeyUsage

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

Application Notes:

- From [1] and relates to FDP_ACF.1.1/KeyUsage: *Whether a subject is currently authorized for access to a secret key is determined by whether the subject has submitted the correct authorization data for the key, and whether this authorization is yet subject to one or more of the re-authorization conditions in FIA_UAU.6/KeyAuth*
  – This application note is satisfied by the TOE.
  - Assigned keys (ie users signature keys) can be used only by signers after SAD validation by the SAM. The signature keys can be used only for digital signature operation and no other purpose.
    The key attributes cannot be changed throughout the lifetime of the key.
  - Master keys are defined as support keys in [1]. These keys can only be used for their purpose.
    The key attributes cannot be changed throughout the lifetime of the key.
- From [1] and relates to FDP_ACF.1.1/KeyUsage. *Whether a subject is currently authorized to change the attributes of a secret key is determined by the iterations of FMT_MSA.1 in 9.4.7*

of [1]
- Attributes of keys cannot be changed through the lifetime of the key

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

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

- From [1]. *The requirements of FDP_ACF.1/KeyUsage apply regardless of how the key is stored by the TOE, including when the key is externally stored (cf. 4.4.1.3 of [1])*
  - This note is satisfied by the TOE.

6.1.3.3 Export from the TOE (FDP_ETC)

6.1.3.3.1 Export of user data without security attributes (FDP_ETC.1)

6.1.3.3.2 Export of user data with security attributes (FDP_ETC.2)

FDP_ETC.2.1/Signer (From [2])

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

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 TOE, are unambiguously 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 TOE: *The TOE is configured as None*

Application Notes:

- From [2]. *The ST writer shall describe which user data that can be exported from the TOE* – Describes as follows – Signer information as well as its signature keys can be stored externally to the TOE protected in confidentiality and in integrity with **MK-BKP-ENC** and **BK-BKP-MAC**

FDP_ETC.2.1/Privileged-User (From [2])

The TSF shall enforce the *Privileged User Creation SFP* when exporting user data, controlled under the SFP(s), outside of the TOE.

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 TOE, 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 TOE: *The TOE is configured as None*

Application Notes:

- From [2]. *The ST writer shall describe which user data that can be exported from the TOE.*
  - No Privileged User security attributes is exported as part of the listed operations. Therefore, this SFR is not relevant to the TOE

6.1.3.4 Information Flow Control Policy (FDP_IFC)

6.1.3.4.1 Subset information Flow Control (FDP_IFC.1)

KeyBasics (From [1])

FDP_IFC.1.1/KeyBasics

The TSF shall enforce the *KeyBasic SFP* on

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

Signer Flow

FDP_IFC.1.1/Signer-Flow

The TSF shall enforce the *Signer-Flow SFP* on *Privileged User and Signer accessing Signer security attributes for all operations*

Privileged User Flow

FDP_IFC.1.1/Privileged-User-Flow

The TSF shall enforce the *Privileged-User-Flow SFP* on *Privileged User accessing Privileged User security attributes for all operations*

6.1.3.5 Information Flow Control Functions (FDP_IFF)

6.1.3.5.1 Simple Security attributes (FDP_IFF.1)

KeyBasics (from [1])

FDP_IFF.1.1/KeyBasics

The TSF shall enforce the *KeyBasics SFP* based on the following types of subject and information security attributes:

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

Application Notes:

- All secret keys are Assigned and all secret keys are not exportable from the TOE, When pre-generated keys are used, they are not assigned, but will be assigned according to definitions of [2].

FDP_IFF.1.2/KeyBasics

The TSF shall permit the 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.*

FDP_IFF.1.3/KeyBasics

The TSF shall enforce the **following additional information flow control rules:** *none*

FDP_IFF.1.4/KeyBasics

The TSF shall explicitly authorise an information flow based on the following rules: *none*

FDP_IFF.1.5/KeyBasics

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

No subject shall be allowed to access the plaintext value of any secret key directly.

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*

Application Notes:

- From [1]. *A secure channel for export of keys in FDP_IFF.1.2/KeyBasics (1) or for import of keys in FDP_IFF.1.2/KeyBasics (3) is one that meets the requirements of FTP_TRP.1/Local or FTP_TRP.1/External.*
  The encrypted form required for keys imported or exported over a secure channel requires encryption of the key itself, in addition to any encryption provided by the secure channel. Unblocking a key as in FDP_IFF.1.2/KeyBasics (6) is intended only to restore the ability of subjects to authorize for access to a key by presenting the correct authorization data. As noted for FMT_MTD.1/Unblock, the subject who unblocks the key shall not be able also to use the key as a result
  - Keys can only be exported and imported via the following systemwide mechanisms:
    - A full backup and restore operation. The backup operation is covered in [1] in a dedicated manner. (also see reference to backup operation in the following note). The backup does not include any keys or signer related information, therefore this item is trivially satisfied. Users and Keys binding can be stored outside the TOE in protected manner in both confidentiality and integrity.

- *From [1]. The requirements of FDP_IFF.1/KeyBasics apply regardless of how the key is stored by the TOE, including when the key is externally stored (cf. 4.4.1.3).*
  Direct access to a key value in FDP_IFF.1.5/KeyBasics (1) is access that makes the value available for reading or modification – this includes operations that would subsequently allow reading or modification of the key (e.g. making a copy of the key with different attributes, or with a different object type that would then allow direct read access). Note that this PP assumes that key values are never modified after they have been generated.
  Export of a key as in FDP_IFF.1.5/KeyBasics (1), (2), (4) and (6) is not the same as backup (governed by FDP_ACF.1/Backup) or external storage of keys under continuing TOE control (governed by other parts of the Key Basics SFP in FDP_IFF.1/KeyBasics, and the Key Usage SFP in FDP_ACF.1/KeyUsage). Thus an Export Flag of 'non-exportable' does not prevent backup or external storage of the keys under continuing TOE control.
  The Security Target and/or Operational Guidance shall specify how any attributes not supplied with an imported key are set when the key is imported (or alternatively how such keys are rejected). Similarly the Security Target and/or Operational Guidance shall describe how the key's attributes are represented when exported, so that their meaning can be understood by the receiver.

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

  - See note above.

Signer Flow (from [2])

FDP_IFF.1.1/Signer-Flow

The TSF shall enforce the *Signer-Flow SFP* based on the following types of subject and information security attributes: *Privileged User and Signer accessing the Signer security attributes*

FDP_IFF.1.2/Signer-Flow

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

The TOE shall be initialized with FDP_ACC.1/TOE Maintenance.

To allow a Signer to sign, the Signer shall be created in the TOE by FDP_ACC.1/Signer Creation followed by FDP_ACC.1/Signer key Pair Generation.

After Signer is created the following operations can be done: FDP_ACC.1/Signer Key Pair Generation, FDP_ACC.1/Signer

Key Pair Deletion, FDP_ACC.1/Supply DTBS/R, FDP_ACC.1/Signer Maintenance and FDP_ACC.1/Signing.

FDP_IFF.1.3/Signer-Flow
The TSF shall enforce the**:** *None*

FDP_IFF.1.4/Signer-Flow
The TSF shall explicitly authorise an information flow based on the following rules: *None*

FDP_IFF.1.5/Signer-Flow
The TSF shall explicitly deny an information flow based on the following rules: *None*

Privileged User Flow

FDP_IFF.1.1/Privileged-User-Flow
The TSF shall enforce the *Privileged-User-Flow SFP* based on the following types of subject and information security attributes: *Privileged User accessing the Privileged User security attributes*

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

The TOE shall be initialized with FDP_ACC.1/TOE Maintenance.

FDP_IFF.1.3/Privileged-User-Flow
The TSF shall enforce the**:** *None*

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

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

6.1.3.6Import from outside of the TOE (FDP_ITC)

6.1.3.6.1 Import of user data with security attributes (FDP_ITC.2)

FDP_ITC.2.1/Signer

The TSF shall enforce the *Signer Creation SFP, Signer Key Pair Generation SFP, Signer Key Pair Deletion, Signer Maintenance SFP, Supply DTBS/R SFP and Signing SFP* when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/Signer

The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/Signer

The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received

FDP_ITC.2.4/Signer

The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/Signer

The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *None*

Application Notes:

- From [2]. *The ST writer shall describe which user data that can be imported to the TOE* -
- As Users and their keys are stored outside the TOE in a protected manner in both confidentiality and integrity, this information can be loaded to the TOE prior to the operations of Supply DTBS/R or Signing. This information is validated upon loading.
- The DTBS/R is imported to the TOE and kept in the transactions table of the signer or used as part of the digital signature operation.

FDP_ITC.2.1/Privileged-User

The TSF shall enforce the *Privileged User Creation SFP* when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/Privileged-User

The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/Privileged-User

The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received

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

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

Application Notes:

- From [2]. *The ST writer shall describe which user data that can be imported to the TOE*
  – The static password of the administrator is imported to the TOE and kept in a protected and disclosed manner.

6.1.3.7 Stored Data Integrity (FDP_SDI)

6.1.3.7.1 Stored data integrity monitoring and action (FDP_SDI.2)

FDP_SDI.2.1 (from [1])
The TSF shall monitor user data stored in containers controlled by the TSF for *integrity errors* on all **keys (including security attributes)**, based on the following attributes: *integrity protection data*.

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

1. *Prohibit the use of the altered data*
2. *Notify the error to the user*

Application Notes:

- From [1]. *No specific requirement is placed here on the nature of the integrity protection data, but the Security Target shall describe this protection measure, and shall identify the iteration of FCS_COP.1 that covers any cryptographic algorithm used.*
  This SFR may also be used in the implementation of the mechanism for protection against modification access to the value of a secret key in FDP_IFF.1.5/KeyBasics, and in the requirement for export of public keys with integrity protection in FDP_IFF.1.2/KeyBasics.
  The integrity protection data in FDP_SDI.2.1 is included in the list of attributes identified in FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys, and protects the value of the key and of its other security attributes, including when the key is externally stored by the TOE (cf. 4.4.1.3 of [1])
  - As described in the ST, any signature key and its binding to the signer has a MAC value that

is checked prior to any usage of the key.

6.1.3.8 Inter-TSF user data confidentiality transfer protection (FDP_UCT)

6.1.3.8.1 Basic data exchange confidentiality (FDP_UCT.1)

FDP_UCT.1.1 (from [2])

The TSF shall enforce the *Signer-Flow SFP and Privileged-User-Flow SFP* to *transmit* and *receive* user data in a manner protected from unauthorised disclosure.

6.1.3.9 Inter-TSF user data integrity transfer protection (FDP_UIT)

6.1.3.9.1 Data exchange integrity (FDP_UIT.1) (from [2])

FDP_UIT.1.1

The TSF shall enforce the *Signer Flow SFP and Privileged User Flow SFP* to *transmit and receive* user data in a manner protected from *modification and insertion* errors **for R.Signer and R.Privileged User and for R.SAD also** from *modification and replay* errors.

FDP_UIT.1.2

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

Application Notes:

- From [2]. *Insertion of objects would mean that authorised creation of Signer and Privileged User could be possible* – This note is satisfied by the TOE and relevant mainly to loading of Signer information from storage external to the TOE.

6.1.3.10 Residual information protection (FDP_RIP)

6.1.3.10.1 **Subset residual information protection (FDP_RIP.1)** (from [1])

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

- authorisation data
- secret keys

Application Notes:

- From [1]: *Authorization data is not to be stored persistently in the TOE; the refinements to ADV_ARC.1 in 9.5.2 of [1] require the approach to minimizing the time that this data is held before deallocation according to FDP_RIP.1*
  – This note is satisfied by the TOE

6.1.4 Identification and authentication (FIA)

6.1.4.1Authentication Failure (FIA_AFL)

6.1.4.1.1 **Authentication failure handling (FIA_AFL.1)** (from [1] and [2])

FIA_AFL.1.1 The TSF shall detect when ***a TOE Maintenance*** configurable positive integer within *3-8* unsuccessful authentication attempts occur related to *Privileged User and Signer authentication*

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall *suspend the Privileged User and when it is a Signer suspend the usage of R.SigningKeyID*

Application Notes:

- From [2]. *The ST writer may extend FIA_AFL.1 to introduce operations to unsuspend Privileged Users or Signers*
  – Direct authentication is applicable only for Administrative roles
- From [2]. *The SFR only applies when the TOE uses any direct authentication.*
  – Direct authentication is applicable only for Administrative roles

6.1.4.1.2 **User attribute definition (FIA_ATD)**

6.1.4.1.3 User attribute definition (FIA_ATD.1) (from [2])

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*.

6.1.4.2 User Authentication (FIA_UAU)

6.1.4.2.1 Timing of authentication (FIA_UAU.1) (from [1] and [2])

FIA_UAU.1.1 The TSF shall allow
(1) Self-test according to FPT_TST_EXT.1
(2) Identification of the user by means of TSF required by FIA_UID.1.
(3) Establishing a trusted path between remote user and the TOE by means of TSF required by FTP_TRP.1
on behalf of the user to be performed before the user is authenticated.

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

Application Notes:

- From [1]. *The Security Target shall separately identify any different types of identification and authentication, e.g. for Administrators, local users, application users, using separate iterations of the FIA_UID.1 and FIA_UAU.1 SFRs where the methods differ. The Security Target shall also separately identify the difference between authentication of users and authorization for use of keys as required for FIA_UAU.6/KeyAuth. Separate iterations of FIA SFRs may be necessary to capture these separate cases*
  – This ST follows [2] in this matter

  The 'list of additional TSF-mediated actions' in FIA_UAU.1.1 may be left empty (equivalent to an assignment of 'None') if applicable
  – This ST follows [2] in this matter

6.1.4.2.2 Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1/Signer
The TSF shall provide
1. *SAML Token validation*

to support Signer authentication.

FIA_UAU.5.2/Signer

The TSF shall authenticate any **Signer's** claimed identity according to

1. *Validating the SAML token based on deployed trusted RSA based certificates or trusted RSA based public keys.*

FIA_UAU.5.1/Privileged-User

The TSF shall provide *Static User Password* to support **Privileged User** authentication.

FIA_UAU.5.2/Privileged-User

The TSF shall authenticate any user's claimed identity according to *static password validation*.

Application Notes:

- From [2]. *This SFR only applies to signer authentication for maintaining signer (FDP_ACC.1/Signer Maintenance, FDP_ACC.1/Signer Key Pair Generation and FDP_ACC.1/Signer Key Pair Deletion) and for signing (FDP_ACC.1/Signing).*
  The ST writer shall list all the authentication factors type used to authenticate signer in accordance with [23]. In particular, the ST writer shall include rules for authentication as part of SAD verification, as in FDP_ACF.1.2/Signing when delegated parties are used to assert the Signer's identity.
  Successful authentication gives Signer access to the relevant R.Signer object as the owner – This is fully described in the ST and in the application notes below.

- A SAML token is produced by the IDP after proper validation of the signer. The SAML token is sent as a SAD, and as part of the digital signature operation.
  The TOE validates the SAML token based on a trusted list of certificates or public keys.

6.1.4.2.3 Re-authenticating (FIA_UAU.6) (from [1])

FIA_UAU.6.1 The TSF shall authorise and re-authorise the user for access to a secret key under the conditions

1. *Authorisation in order to be granted initial access to the key; and*
2. *Re-authorization is required after the current JWT token expired*

Application Notes:

- From [1]. *Note that any use of a key requires an initial authorization by presentation of the correct authorization data. Subsequent uses may require re-authorization on every use (in this case 'Authorization on every subsequent access to the key' is selected in FIA_UAU.6.1/KeyAuth (2)), or else the TOE may allow some uses of the key without further authorization until one of the specified re-authorization conditions occurs.*

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

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

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

  Each 'use' of a key is expected to relate to one cryptographic function carried out with the key. If there are circumstances where a different interpretation may be placed on the 'use' of a key then this shall be identified and explained in the Security Target and the Operational Guidance. The intention here is to make clear any situations that are relevant to a key owner who can be held responsible for use of the key (such as any case where a single authorization for use of a key could allow the creation of more than one signature using the authorized key). Note that in order to make qualified electronic signatures under Regulation (EU) 910/2014 [7] then the user/application shall be able to precisely control the signatures that can be made under each authorization.

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

6.1.4.3User identification (FIA_UID)

6.1.4.3.1 Timing of identification (FIA_UID.1) (from [1])

FIA_UID.1.1 The TSF shall allow

1. *Self test according to FPT_TST_EXT.1*
2. *None*

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

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

Application Notes:

- From [1]. *The 'list of additional TSF-mediated actions' may be left empty (equivalent to an assignment of 'None') if applicable*
  – The note is satisfied by the TOE.
- Any attempt to perform a TSF related request without being authenticated previously will reject the attempt without having a dedicated entry in the audit log.
  There are some specific general commands that are identified as anonymous commands that do not require a previous user authentication. This commands only reply with general information such as versioning information of the TOE and the overall TOE status.

6.1.4.3.2 User identification before any action (FIA_UID.2) (from [2])

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Notes:

- From [2]. *The 'list of additional TSF-mediated actions' may be left empty (equivalent to an assignment of 'None') if applicable.*
  – The note is satisfied by the TOE.

6.1.4.4User-subject binding (FIA_USB)

6.1.4.4.1 User-Subject binding (FIA_USB.1) (from [2])

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

1. *R.ReferenceSignerAuthenticationData*
2. *R.SigningKeyID*
3. *R.SVD*
4. *R.Signer*
5. *None*

to Signer.

1. R.ReferencePrivilegedUserAuthenticationData

2. None

to Privileged User.

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 authorised to create a new Signer.*
2. *Whether the subject is a Privileged User authorised to create a new Privileged User.*
3. *None*

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 authorised to modify an R.Signer object*
2. *Whether the subject is a Signer authorised to modify his own R.Signer object*
3. *None*

Application Notes:

- From [2]. *In FIA_USB.1.2 several attributes including R.SigningKeyID, R.SVD and R.DTBS/R may initially be empty.*
- From [2]. *The ST writer may include the R.AuthrorisationData as a security attribute of the Signer*
  - In this TOE, the Authorisation data is not a security attribute of the signer.
- From [2]. *The ST writer shall describe if R.DTBS/R is a Signer attribute. This is expected if a Privileged User and not the Signer submits it to the TOE*
  –The DTBS/R is a signer attribute since it is kept as part of the transactions table of the signer.

6.1.5 Security management (FMT)

6.1.5.1 Management of security attributes (FMT_MSA)

6.1.5.1.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1/Signer (From [2])

The TSF shall enforce the

1. *Signer Creation SFP* to restrict the ability to *create* the security attributes *listed in FIA_USB.1 for Signer* to *authorised Privileged User*
2. *Generate Signer Key Pair SFP* to restrict the ability to *generate* the security attributes *R.SVD and R.SigningKeyID* to *authorised Privileged User and Signer.*
3. *Signer Key Pair Deletion SFP* to restrict the ability to *destruct* the security attribute *R.SVD and R.SigningKeyID as part of R.Signer* to *authorised Signer*
4. *Supply DTBS/R SFP* to restrict the ability to *create* the security attribute *R.DTBS/R as part of R.Signer* to *authorised Privileged User*
5. *Signing SFP* to restrict the ability to *create* the security attribute *R.DTBS/R as part of R.Signer* to *authorised Signer.*
6. *Signing SFP* to restrict the ability to *query the security attributes as listed in FIA_USB.1* to *authorised Signer.*
7. *Signer Maintenance SFP* to restrict the ability to *change* the security attributes *R.ReferenceSignerAuthenticationData as part of R.Signer* to *authorised Privileged User and Signer*

FMT_MSA.1.1/Privileged-User (from [2])

The TSF shall enforce the

1. *Privileged User Creation SFP* to restrict the ability to create and *query* the security attributes *listed in FIA_USB.1 for Privileged User* to *authorised Privileged User*

FMT_MSA.1.1/GenKeys (from [1])

The TSF shall enforce the *Key Usage SFP to* restrict the ability to *modify* the security attributes *None* to *None*.

FMT_MSA.1.1/AKeys (from [1])

The TSF shall enforce the *Key Usage SFP to* restrict the ability to *modify* the security attributes *None* to None.

Application Notes:

- The application notes from [1] were not copied.
  - The note is satisfied by this TOE. There is no command that leads to changing of the security attribute of any key inside the TOE.

6.1.5.1.2 Secure security attributes (FMT_MSA.2) (from [2])

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

6.1.5.1.3 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1/Signer (from [2])

The TSF shall enforce the *Signer Creation SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Signer

The TSF shall allow the *Privileged User* to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3.1/Privileged-User (from [2])

The TSF shall enforce the *Privileged User Creation SFP* to provide *restrictive* 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* to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3.1/Keys (from [1])

The TSF shall enforce the *Key Usage SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Keys

The TSF shall allow the *None* to specify alternative initial values to override the default values when an object or information is created.

Application Notes:

- The application notes from [1] were not copied.
  - The note is satisfied by this TOE. There is no command that leads to changing of the security attribute of any key inside the TOE.

6.1.5.2 Specification of Management Function (FMT_SMF)

6.1.5.2.1 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1/From-HSM-PP (from [1])

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

1. *Unblock of access due to authentication or authorisation failures*
2. *Modifying attributes of keys*
3. *Export and deletion of the audit data, which can take place only under the control of the Administrator role*
4. *Backup and Restore Functions*
5. *No Key Import Functions*
6. *No Key Export Functions*

Application Notes:

- From [1]. *The unblocking of authentication or authorization failures in FMT_SMF.1.1 (1) is related to the authentication failures described in FIA_AFL.1. The attributes of keys in FMT_SMF.1.1 (2) correspond to the attributes in FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys. Export of audit data in FMT_SMF.1.1 (3) relates to the ability to export audit data from the TOE for preservation and storage elsewhere. The selections in FMT_SMF.1.1 (4), (5) and (6) identify whether or not the TOE provides the relevant functions*

(and shall therefore correspond to the relevant statements in the ST for FDP_IFF.1.2/KeyBasics, FDP_ACC.1/Backup and FDP_ACF.1/Backup.
– Since the TOE is used in indirect mode, no unblock is required in this case.

- *Modifying attributes of keys* and *deleting the Audit Log* operations are not supported by the TOE, therefore this part of the SFR is trivially satisfied.

- Only the backup and restore operation may be relevant although no signature keys or signer information is backed-up.

FMT_SMF.1.1/From-SAM-PP (from [2])

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

1. *Signer management*
2. *Privileged User management*
3. *Configuration management (Installation, setting system parameters and uploading trusted RSA certificates/public keys)*
4. *Upload a new software version.*

6.1.5.3 Management of TSF Data (FMT_MTD)

6.1.5.3.1 Management of TSF Data (FMT_MTD.1)

FMT_MTD.1.1/Unblock (from [1])

The TSF shall restrict the ability to *unblock* the *a blocked user account* to *None*

Application Notes:

- From [1]. *The list of TSF data assigned shall correspond to the relevant data blocked by authentication or authorization failures according to the associated iteration(s) of FIA_AFL.1. For the purposes of unblocking, the TSF data in the assignment includes any key that can be affected by blocking due to failure of authorization (as in FIA_UAU.6), as well as user accounts (as in FIA_UAU.1) blocked by authentication/authorization failures.*
  There is a distinction between administrators authorized to unblock a key and users authorized to use the key. When unblocking a secret key, the unblocking process shall not allow a subject to use the key other than a subject who is authorized by presentation of the current authorization data. For example, an administrator who is able to unblock the key cannot then ***use*** *the key as a result of the unblocking (so the unblocking process does not itself allow the key to be used, nor does it enable the authorization data to be changed without proving knowledge of the previous authorization data). This is a part of ensuring that sole control of secret keys can be achieved*.
- There is no administrative operation to unblock the user and thus if the user gets blocked he/she will not be able to sign.

FMT_MTD.1.1/AuditLog

The TSF shall restrict the ability to *control export and deletion of the audit log records* to the *Administrator role*.

Application Notes:

- From [1]. *The control of export and deletion of the audit log records helps to ensure their protection against accidental or malicious deletion (deletion should normally occur only after the records have been exported and preserved outside the TOE). Note that this does not require the Administrator to carry out these export or delete operations manually as long as the actions are controlled by the Administrator.*
- The TOE does not support any ability for export or deletion on the audit log by an administrator. However, the Audit log information is sent view secure channel to an Audit Log Server.

FMT_MTD.1.1/Modify-TSF-DATA (from [2]))

The TSF shall restrict the ability to *R.TSF_DATA* to the *Privileged User*.

Application Notes:

- From [2]. *The TSF data includes configuration of administrator roles.*
  – This note is satisfied by the TOE

6.1.5.3.2 Security management roles (FMT_SMR)

6.1.5.3.3 Security roles (FMT_SMR.1) (from [1]))

FMT_SMR.1.1 The TSF shall maintain the roles *Administrator*, *Key User*, *Users Administrator (R.UsersAdmin)*.

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

Application Notes:

- From [1]. *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 authorization requirements for the key – this role may sometimes act through a client application*
  – Any local client related definition is defined according to [2].
- The role Administrator stands for Appliance Administrator (R.ApplianceAdmin)
- The role *Key User* stands for Signer (R.Signer)

6.1.5.3.4 Restrictions on security roles (FMT_SMR.2) (from [2])

FMT_SMR.2.1 The TSF shall maintain the roles *Signer and Privileged User, None*

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

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

Application Notes:

- From [2]. *The ST writer shall describe which roles are defined in the TOE and which operations the role can perform*
  – This is defined in the ST.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 TSF physical protection (FPT_PHP)

6.1.6.1.1 Passive detection of physical attack (FPT_PHP.1) (from [1] and [2])

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

Application Notes:

- From [1]. *Passive detection of a physical attack is typically achieved by using physical seals and an appropriate physical design of the TOE that allows the TOE administrator to verify the physical integrity of the TOE as part of a routine inspection procedure.*
  Because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 for this TOE is equivalent to the physical security mechanisms for tamper detection and response required by 7.7.2 of [1] Physical security general requirements and 7.7.3 of [1] Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3. (Cf. refinement of AVA_VAN.5 in 9.5.2. of [1])
  – This note is satisfied by the TOE. *ISO/IEC 19790:2012 is equivalent to* [25]

6.1.6.1.2 Resistance to physical attack (FPT_PHP.3) (From [1] and [2])

FPT_PHP.3.1 The TSF shall resist *opening the Appliance* to the *cover* by responding automatically such that the SFRs are always enforced.

Application Notes:

- From [1]. *This SFR is linked to the requirements for passive detection of physical attacks in FPT_PHP.1, and should identify the relevant responses of the TOE involved in meeting the key zeroisation requirements of ISO/IEC 19790:2012 Security Level 3. As in the case of FPT_PHP.1, because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.3 for this TOE is equivalent to the level of assessment for this aspect of tamper detection and response required for 7.7.2 of [1] Physical security general requirements and 7.7.3 of [1] Physical security requirements by each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3. (Cf. refinement of AVA_VAN.5 in 9.5.2. of [1])*

– This note is satisfied by the TOE. *ISO/IEC 19790:2012 is equivalent to* [25]

6.1.6.2 Fail Secure (FPT_FLS)

6.1.6.2.1 Failure with preservation of secure state (FPT_FLS.1) (from [1])

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:
1. *Self-test according to FPT_TST_EXT.1 fails*
2. *Environmental conditions are outside normal operating range (including temperature and power)*
3. *Failures of critical TOE hardware components (including the RNG) occur*
4. *Corruption of TOE software occurs*
5. *None*

Application Notes:

- From [1]. *The Operational Guidance shall include a description of the specific failures that are detected (e.g. the thresholds for environmental conditions, and the nature of the monitoring of specific critical TOE hardware components), how these failures are notified, and the actions that should be taken in response to each.*

6.1.6.3 Replay detection (FPT_RPL)

6.1.6.3.1 Replay detection (FPT_RPL.1) (from [2])

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

FPT_RPL.1.2 The TSF shall perform *reject the signature operation* when replay is detect.

6.1.6.4 Time stamps (FPT_STM)

6.1.6.4.1 Reliable time stamps (FPT_STM.1) (from [1] and [2])

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

Application Notes:

- From [1]. *The TOE must provide timestamps suitable for supporting the time in an audit record for FAU_GEN.1. If the TOE provides additional timestamping services for client applications, or other record of the time of an operation for client applications, then these should be covered in one or more separate iterations of the SFR, with an Application*

Note added to define any specific requirement for reliability of the time information for that service. – There is no additional timestamping service

- From [2]. *The TOE may receive a reliable time source from its environment.*
  - It is possible to configure the TOE to get a reliable time source from an external entity in the operational environment

6.1.6.5 Inter-TSF TSF Data Consistency (FPT_TDC)

6.1.6.5.1 Inter-TSF basic TSF data consistency (FPT_TDC.1) (from [2])

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret
1. *R.Signer,*
2. *R.Reference_Signer_Authentication_Data,*
3. *R.SAD,*
4. *R.DTBS/R*
5. *R.SVD*
6. *R.Privileged_User*
7. *R.Reference_Privileged_User_Authentication_Data*
8. *R.TSF_DATA*

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

Application Notes:

- From [2]. *The SFR is used to handle the situation where the whole or part of the above data are stored outside the TOE*.

6.1.6.6Basic TSF self testing (FPT_TST_EXT)

6.1.6.6.1 Basic TSF self testing (FPT_TST_EXT.1) (from [1])

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests *during initial start-up (on power-on), periodically during normal operation, at the conditions Asymmetric key generation* to demonstrate the correct operation of the TSF:
1. *At initial start-up (on power-on):*
   a. *Software/firmware integrity test*
   b. *Cryptographic algorithm tests*

c. *Random number generator test*

2. *Periodically*
   a. *Random number generator test*
3. *Conditionally*
   a. *Pairwise Consistency at Asymmetric Key Generation*

Application Notes:

- From [1]. *Completion of the selection in FPT_TST_EXT.1.1 may be by 'None' (in which case the 'and' preceding the selection should be deleted and no selection text included). Completion of the list of additional tests in the final assignment may include tests performed at initial start-up (or power-on) and/or tests run under the conditions specified in the earlier selection and assignment. The term 'start-up (or power-on) means that the tests should be executed at least any time that the TOE is powered-on. The tests of the cryptographic functions shall include all cryptographic functions covered by FCS_COP.1. The Operational Guidance shall include a description of the errors that may arise from self-test and the actions that should be taken in response to each. The SFR is used to handle the situation where the whole or part of the above data are stored outside the TOE*
  – This note is satisfied by the TOE

6.1.7 Trusted path/channels (FTP)

6.1.7.1 Inter-TSF trusted channel (FTP_ITC)

6.1.7.1.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1/CM (From [2])

The TSF shall provide a communication path between itself and **a cryptographic module certified according to [EN 419 221-5]** that is logically distinct from other communication paths and provides assured authentication of its end points and protection of the communicated data from modification or disclosure.

FTP_ITC.1.2/CM

The TSF shall permit **the TSF and a cryptographic module certified according to [EN 419 221-5]** to initiate communication via the trusted channel.

FTP_ITC.1.3/CM

The TSF shall initiate communication via the trusted channel for *None.*

Application Notes:

- From [2]: *FTP_ITC.1/CM must be completed in a Security Target to reflect the way that the TOE communicates with the cryptographic module, and to justify its security. Where the TOE and the cryptographic module are located within the same hardware appliance (e.g. the TOE being a local application running on a server and communicating with a PCI card on the server's internal PCI bus) then the trusted channel may be mapped in the Security Target to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment).*
  - Both the SAM application and the CM are in the scope of the TOE. In addition, the TOE only supports local SAM application installed in the TOE. As mentioned in the application note from the [2], the trusted channel can be mapped in the Security Target to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment). Thus this SFR is trivially satisfied.

6.1.7.2Trusted path (FTP_TRP)

6.1.7.2.1 Trusted path (FTP_TRP.1)

FTP_TRP.1.1/Local (from [1])

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

FTP_TRP.1.2/Local

The TSF shall permit *the TSF* to initiate communication via the trusted path.

FTP_TRP.1.3/Local

The TSF shall require the use of the trusted path for *None.*

Application Notes:

- From [1]. *FTP_TRP.1/Local shall be completed in a Security Target to identify the local client applications and to reflect the way that the TOE communicates with them, and to justify the security of this communication path. Where the TOE and local client applications are located within the physical boundary of the same hardware appliance (e.g. local applications running on a server and communicating with a PCI card on the server's internal PCI bus) then the trusted path may be mapped in the Security Target to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment).*
  If the TOE does not provide an interface for local client applications, then this SFR is not applicable and is trivially satisfied. This should be stated in the Security Target.

The TOE may provide other additional channels that provide only authentication and integrity protection (not confidentiality), in which case other iterations of FTP_TRP.1 may be added in the ST, allowing the selection of only modification protection in FTP_TRP.1.1 for these additional iterations.
The Security Target shall identify in an application note the iterations of FCS_COP.1 that provide any cryptographic functions that contribute to the implementation of the trusted path, and the SFRs that provide the authentication of the end points.
- The Cryptographic module of the TOE provide an interface for local client applications according to the definitions of [2], therefore this SFR is not applicable and is trivially satisfied.

FTP_TRP.1.1/External (from [1])

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

FTP_TRP.1.2/External

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

FTP_TRP.1.3/External

The TSF shall require the use of the trusted path for *None*

Application Notes:

- From [1]. *FTP_TRP.1/External shall be completed in a Security Target to identify the external client applications and to reflect the way that the TOE communicates with them, and to justify the security of this communication path. The word "remote" in FTP_TRP.1.1/External and FTP_TRP.1.2/External refers to client applications that are described as "external" in the rest of this PP.*
  If the TOE does not provide an interface for external client applications, then this SFR is not applicable and is trivially satisfied. This should be stated in the Security Target.
  The TOE may provide other additional channels that provide only authentication and integrity protection (not confidentiality), in which case other iterations of FTP_TRP.1 may be added in the ST, allowing the selection of only modification protection in FTP_TRP.1.1 for these additional iterations.
  The Security Target shall identify in an application note the iterations of FCS_COP.1 that provide any cryptographic functions that contribute to the implementation of the trusted path, and the SFRs that provide the authentication of the end points.
  - The client applications (as defined in [2]) and cryptographic module (as defined in [1]) are implemented as one TOE. Therefore this SFR is not applicable and is trivially satisfied.

FTP_TRP.1.1/SIC (from [2])

The TSF shall provide a communication path between itself and ***Remote Signer through the SIC*** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification*

FTP_TRP.1.2/SIC

The TSF shall permit ***Remote Signer through SIC*** to initiate communication via the trusted channel.

FTP_TRP.1.3/SIC

The TSF shall initiate communication via the trusted channel for

1. *FDP_ACC.1/Signing*
2. *None*

Application Notes:

- From [2]. *Since it is not all data transmitted to the TOE that needs to be protected in confidentiality, FTP_TRP.1.1/SIC only requires protection from modification. The ST writer shall describe if the SAP can be used to transmit sensitive data and how these are protected in confidentiality.*
  The TOE is not expected to verify the SIC as a communication end point and it may rely on the signer authentication.
  - All input information for the signature operation is protected by the SAML token integrity

FTP_TRP.1.1/SSA (From [2])

The TSF shall provide a communication path between itself and ***Privileged User through SSA*** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification*

FTP_TRP.1.2/SSA

The TSF shall permit ***Privileged User through SSA*** to initiate communication via the trusted channel.

FTP_TRP.1.3/SSA

The TSF shall initiate communication via 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/TOE Maintenance*
8. *None*

Application Notes:

- From [2]. *Since it is not all data transmitted to the TOE that needs to be protected in confidentiality, FTP_TRP.1/SSA only requires protection from modification.*
  – Communication from SSA to the TOE is based on TLS1.2

- *FDP_ACC.1/Privileged User Creation* and *FDP_ACC.1/TOE Maintenance* is done from a dedicated client software that represent the SSA trusted channel.

6.2 Security Assurance Requirements

The assurance level for this TOE is EAL4+ AVA_VAN.5.

Assurance Class	Assurance components
ADV: Development	- ADV_ARC.1 Security architecture description - 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
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_IND.2 Independent testing – sample - ATE_FUN.1 Functional testing
AVA: Vulnerability assessment	- AVA_VAN.5 Advanced methodical vulnerability analysis

Table 2 - Assurance Requirements: EAL4+ AVA_VAN.5

6.2.1 Rationale for SARs

The assurance level for this Protection Profile is **EAL4 augmented with AVA_VAN.5**
EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialized processes and practices. It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions.
The TOE described in this Protection Profile is just such a product. Augmentation results from the selection of **AVA_VAN.5**. All the dependencies of AVA_VAN.5 are satisfied by other assurance components in the EAL4 assurance package.

6.2.2 AVA_VAN.5 - Advanced methodical vulnerability analysis

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

6.2.3 Refinements of Security Assurance Requirements

The following refinements are made to selected assurance requirements in Table 2.

The refinements made to the SARs are identical to the SAR refinements in [1] section 6.4.1.

Follows a SFR dependency satisfaction table

Requirement	Dependencies	PPs	Relevant Conf.
FAU_GEN.1	- FPT_STM.1	- [1] - [2]	
FAU_GEN.2	- FAU_GEN.1 - FIA_UID.1	- [1] - [2]	
FAU_STG.2	- FAU_GEN.1	- [2]	
FCS_CKM.1/SIGNATURE-KEY	- FCS_COP.1/SIGNING - FCS_CKM.4	- [1]	

Requirement	Dependencies	PPs	Relevant Conf.
FCS_CKM.1/SYMMETRIC-KEY	- FCS_COP.1/DATA-INTEG - FCS_COP.1/BKP-DATA-INTEG - FCS_COP.1/BKP-ENCRYPTION - FCS_COP.1/ADMIN_SESSION_DATA_INTEG - FCS_CKM.4	- [1]	
FCS_CKM.4	- FCS_CKM.1 - FDP_ITC.2/Signer	- [1]	
FCS_COP.1/SIGNING	- FCS_CKM.1/SIGNATURE-KEY - FCS_CKM.4	- [1]	
FCS_COP.1/DATA-INTEG	- FCS_CKM.1/SYMMETRIC-KEY - FCS_CKM.4	- [1]	
FCS_COP.1/KEY-ENCRYPTION	- FCS_CKM.1/SYMMETRIC-KEY - FCS_CKM.4	- [1]	
FCS_COP.1/BKP-DATA-INTEGRITY	- FCS_CKM.1/SYMMETRIC-KEY - FCS_CKM.4	- [1]	
FCS_COP.1/BKP-ENCRYPTION	- FCS_CKM.1/SYMMETRIC-KEY - FCS_CKM.4	- [1]	
FCS_COP.1/ADMIN-SESSION-DATA-INTEG	- FCS_CKM.1/SYMMETRIC-KEY - FCS_CKM.4	- [1]	
FCS_COP.1/FIRM-UPD	- FCS_CKM.1/SIGNATURE-KEY - FCS_CKM.4	- [1]	
FCS_COP.1/TLS-SESSION	- FCS_CKM.1/SIGNATURE-KEY - FCS_CKM.4	- [1]	
FCS_RNG.1		- [1]	
FDP_ACC.1/Privileged-User-Creation	- FDP_ACF.1/Privileged-User-Creation	- [2]	
FDP_ACC.1/Signer-Creation	- FDP_ACF.1/Signer-Creation	- [2]	
FDP_ACC.1/Signer-Maintenance	- FDP_ACF.1/Signer-Maintenance	- [2]	
FDP_ACC.1/Signer-Key-Pair-Generation	- FDP_ACF.1/Signer-Key-Pair-Generation	- [2]	
FDP_ACC.1/Signer-Key-Pair-Deletion	- FDP_ACF.1/Signer-Key-Pair-Deletion	- [2]	
FDP_ACC.1.1/Supply-DTBS/R	- FDP_ACF.1/Supply-DTBS/R	- [2]	
FDP_ACC.1.1/Signing	- FDP_ACF.1/Signing	- [2]	
FDP_ACC.1/TOE-Maintenance	- FDP_ACF.1/TOE-Maintenance	- [2]	
FDP_ACC.1/Backup	- FDP_ACF.1/Backup	- [1]	
FDP_ACC.1/KeyUsage	- FDP_ACF.1/KeyUsage	- [1]	
FDP_ACF.1/Privileged-User-Creation	- FDP_ACC.1/Privileged-User-Creation - FMT_MSA.3	- [2]	
FDP_ACF.1/Signer-Creation	- FDP_ACC.1/Signer-Creation - FMT_MSA.3	- [2]	
FDP_ACF.1/Signer-Maintenance	- FDP_ACC.1/Signer-Maintenance - FMT_MSA.3	- [2]	

Requirement	Dependencies	PPs	Relevant Conf.
FDP_ACF.1/Signer-Key-Pair-Generation	- FDP_ACC.1/Signer-Key-Pair-Generation - FMT_MSA.3	- [2]	
FDP_ACF.1/Signer-Key-Pair-Deletion	- FDP_ACC.1/Signer-Key-Pair-Deletion - FMT_MSA.3	- [2]	
FDP_ACF.1.1/Supply-DTBS/R	- FDP_ACC.1/ Supply-DTBS/R - FMT_MSA.3	- [2]	
FDP_ACF.1.1/Signing	- FDP_ACC.1/Signing - FMT_MSA.3	- [2]	
FDP_ACF.1/TOE-Maintenance	- FDP_ACC.1/TOE-Maintenance - FMT_MSA.3	- [2]	
FDP_ACF.1/Backup	- FDP_ACC.1/Backup - FMT_MSA.3	- [1]	
FDP_ACF.1/KeyUsage	- FDP_ACC.1/KeyUsage - FMT_MSA.3	- [1]	
FDP_ETC.2/Signer	- FDP_ACC.1/Signer-Creation	- [2]	
FDP_ETC.2/Privileged-User	- FDP_ACC.1/Privileged-User - Creation	- [2]	- Not Relevant
FDP_IFC.1/KeyBasics	- FDP_IFF.1/KeyBasics	- [2]	
FDP_IFC.1/Signer-Flow	- FDP_IFF.1/Signer-Flow	- [2]	
FDP_IFC.1/Privileged-User-Flow	- FDP_IFF.1/Privileged-User-Flow	- [2]	
FDP_IFF.1/KeyBasics	- FDP_IFC.1/KeyBasics - FMT_MSA.3/Keys	- [2]	
FDP_IFF.1/Signer-Flow	- FDP_IFC.1/Signer-Flow - FMT_MSA.3/Signer	- [2]	
FDP_IFF.1/Privileged-User-Flow	- FDP_IFC.1/Privileged-User-Flow - FMT_MSA.3/Privileged-User	- [2]	
FDP_ITC.2/Signer	- FDP_IFC.1/Signer-Flow - FMT_MSA.3/Signer	- [2]	
FDP_ITC.2/Privileged-User	- FDP_IFC.1/Privileged-User-Flow - FMT_MSA.3/Privileged-User	- [2]	
FDP_SDI.2	-	- [1]	
FDP_UCT.1	- FDP_IFC.1/Signer-Flow - FDP_IFC.1/Privileged-User-Flow - FTP_TRP.1/SIC - FTP_TRP.1/SSA	-	
FDP_UIT.1	- FDP_IFC.1/Signer-Flow - FDP_IFC.1/Privileged-User-Flow - FTP_TRP.1/SIC - FTP_TRP.1/SSA	-	-
FDP_RIP.1		- [1]	
FIA_AFL.1	- FIA_UAU.1	- [1] - [2]	
FIA_ATD.1	-	- [2]	
FIA_UAU.1	- FIA_UID.1	- [1]	

Requirement	Dependencies	PPs	Relevant Conf.
FIA_UAU.5/Signer	-	- [2]	
FIA_UAU.5/Privileged-User	-	- [2]	
FIA_UAU.6	-	- [1]	
FIA_UID.1	-	- [1]	
FIA_UID.2	-	- [2]	
FIA_USB.1	- FIA_ATD.1	- [2]	
FMT_MSA.1/Signer	- FDP_ACC.1/Signer-Creation - FDP_ACC.1/Signer-Key-Pair Generation - FDP_ACC.1/Signer-Key-Pair Deletion - FDP_ACC.1/Supply-DTBS/R - FDP_ACC.1/Signing - FDP_ACC.1/Signer-Maintenance - FMT_SMR.1 - FMT_SMF.1	- [2]	
FMT_MSA.1/Privileged-User	- FDP_ACC.1/Privileged-User-Creation - FMT_SMR.1 - FMT_SMF.1	- [2]	
FMT_MSA.1/GenKeys	- FDP_ACC.1/Key-Usage - FMT_SMR.1 - FMT_SMF.1	- [1]	
FMT_MSA.1/AKeys	- FDP_ACC.1/Key-Usage - FMT_SMR.1 - FMT_SMF.1	- [1]	
FMT_MSA.2	- FDP_ACC.1/Signer-Creation - FDP_ACC.1/Signer-Key-Pair Generation - FDP_ACC.1/Signer-Key-Pair Deletion - FDP_ACC.1/Supply-DTBS/R - FDP_ACC.1/Signing - FDP_ACC.1/Signer-Maintenance FDP_ACC.1/Privileged-User-Creation - FMT_MSA.1/Signer - FMT_MSA.1/Privileged-User - FMT_SMR.1	- [2]	
FMT_MSA.3/Signer	- FMT_MSA.1/Signer - FMT_SMR.1	- [2]	
FMT_MSA.3/Privileged-User	- FMT_MSA.1/Privileged-User - FMT_SMR.1	- [2]	
FMT_MSA.3/Keys	- FMT_MSA.1/GenKeys - FMT_MSA.1/AKeys - FMT_SMR.1	- [1]	
FMT_SMF.1-From-HSM-PP	-	- [1]	

Requirement	Dependencies	PPs	Relevant Conf.
FMT_SMF.1-From-SAM-PP	-	- [2]	
FMT_MTD.1/Unblock	- FMT_SMR.1 - FMT_SMF.1	- [1]	- Not relevant
FMT_MTD.1/AuditLog	- FMT_SMR.1 - FMT_SMF.1	- [1]	-
FMT_MTD.1/Modify-TSF-DATA	- FMT_SMR.1 - FMT_SMF.1	- [2]	
FMT_SMR.1	- FIA_UID.1	- [1]	
FMT_SMR.2	- FIA_UID.1	- [2]	
FPT_PHP.1		- [1]	
FPT_PHP.3	-	- [1] - [2]	
FPT_FLS.1	-	- [1]	
FPT_STM.1	-	- [1] - [2]	
FPT_RPL.1	-	- [2]	
FPT_TDC.1	-	- [2]	
FPT_TST_EXT	-	- [1]	
FTP_ITC.1/CM	- FDP_IFC.1/KeyBasics - FMT_MSA.3/Keys	- [2]	- Not Relevant
FTP_TRP.1/Local	-	- [1]	- Not Relevant
FTP_TRP.1/External	-	- [1]	- Not Relevant
FTP_TRP.1/SIC	-	- [2]	
FTP_TRP.1/SSA	-	- [2]	

Table 3 - SFR dependency satisfaction table

6.3 Security Requirements Rationale

There are no security functional requirements introduced in this Security Target beyond the defined in [1] and [2].

Security requirements rationale from [1]:

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

FMT_SMR.1				X										X
FMT_SMF.1				X										X
FMT_MTD.1/Unblock				X										
FMT_MTD.1/Audit Log														X
FMT_MSA.1/GenKeys					X									
FMT_MSA.1/AKeys					X									
FMT_MSA.3/Keys					X									
FAU_GEN.1														X
FAU_GEN.2														X
FAU_STG.2														X

Security requirements rationale from [2]:

	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_DATA	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD	OT.PRIVILEGED_USER_MANAGEMENT	OT.PRIVILEGED_USER_AUTHENTICATION	OT.PRIVILEGED_USER_PROTECTION	OT.SIGNER_MANAGEMENT	OT.SYSTEM_PROTECTION	OT.AUDIT_PROTECTION	OT.SAD_VERIFICATION	OT.SAP	OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION	OT.DTBSR_INTEGRITY	OT.SIGNATURE_INTEGRITY	OT.CRYPTO	OT.RANDOM
Security Audit																	
FAU_GEN.1										X							
FAU_GEN.2										X							
Cryptographic Support																	
FCS_CKM.1			X													X	
FCS_CKM.4			X														
FCS_COP.1			X												X	X	
FCS_RNG.1			X														X

User Data Protection																	
FDP_ACC.1/Privileged User Creation					X												
FDP_ACF.1/Privileged User Creation					X												
FDP_ACC.1/Signer Creation		X						X									
FDP_ACF.1/Signer Creation		X						X									
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/ TOE Maintenance									X								
FDP_ACF.1/TOE Maintenance									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															
FDP_UIT.1	X															
Identification and Authentication																
FIA_AFL.1						X					X					
FIA_ATD.1	X				X		X									
FIA_UAU.1						X					X					
FIA_UAU.5/Signer											X					
FIA_UAU.5/Privileged User						X										
FIA_UID.2					X		X	X								
FIA_USB.1	X		X		X		X									
Security Management																
FMT_MSA.1/Signer								X								
FMT_MSA.1/Privileged User					X			X								
FMT_MSA.2					X			X								
FMT_MSA.3/Signer								X								
FMT_MSA.3/Privileged User					X			X								
FMT_MTD.1									X							
FMT_SMF.1									X							
FMT_SMR.2									X							
Protection of the TSF																
FPT_PHP.1									X							
FPT_PHP.3									X							
FPT_RPL.1												X				
FPT_STM.1										X						
FPT_TDC.1	X				X											
Trusted Path/Channels																
FTP_TRP.1/SSA									X					X		

FTP_TRP.1/SIC												X	X	X			
FTP_ITC.1/CM			X												X		

Therefore, this section is covered in [1] and [2].

7 TOE Summary Specification

To fulfill the Security Functional Requirements, the TOE comprises the following Security Functions (TSF):

1. Access Control
2. Identification and Authentication
3. Cryptographic Operation
4. Secure communication and session management
5. Auditing
6. Tamper detection
7. High availability and disaster recovery
8. Self test

Each of the TOE security functions is described in the following sections in detail.

7.1 Access Control (TSF.ACC)

The access control rights being described below depend on the current user role “Appliance Administrator“ (R.ApplianceAdmin), “Users Administrator” (R.UserAdmin), SSA Admin (R.SSA) or “Signer” (R.Signer).
All access rights that defined below are enforced by the TOE.

1. The TOE makes sure that the creation of a user account is only allowed for an authenticated Users Administrators or SSA Admin for creation of a signer. Upon creation of a signer, the Signer-Keys blob structure is protected by an HMAC value calculated by the TOE.
   The TOE will make sure that only proper parameters are set when a new Administrator is created.
   The TOE will make sure that only proper parameters are set when a new Signer is created.
2. The TOE makes sure that the generation of the SCD/SVD pair is allowed only for SSA Admin on behalf of the signer and only when the signer account exists.
   During key generation, the TOE generates the certificate request, which will be sent to the CA.
   The TOE will also make sure that only the SSA Admin is allowed to modify the Signer-Keys blob.
   The TOE makes sure that the operation is allowed only for a signer that was created by the TOE.
   The TOE will validate the Signer-Keys blob using HMAC validation and will eventually calculate a new HMAC for the updated Signer-Keys blob.
3. The TOE makes sure that only the SSA Admin can revoke the SCD of a signer’s account.
   The TOE will also make sure that only the SSA Admin is allowed to modify the Signer-Keys blob.

The TOE will validate the Signer-Keys blob using HMAC validation and will eventually calculate a new HMAC for the updated Signer-Keys blob.
The TOE makes sure that the operation is allowed only for a signer that was created by the TOE and a key that was created for the specified Signer.

4. The TOE makes sure that only a SSA Admin can revoke the account of the Signer which will revoke the entire list of SCDs of the Signer.
5. The TOE makes sure that signature creation is allowed only for Signer for DTBS/R or transaction ID that (a) if security attribute “SCD operational” has been set to “yes”.
   The signature key will be used by the signer and only for the purpose of a digital signature operation.
   Any key that is used as part of the digital signature operation will be removed from the internal memory of the TOE and not be put in any nonvolatile memory inside the TOE.
   The TOE makes sure that the operation is allowed only for a signer that was created by the TOE and for a key that was generated by the TOE for the specific signer.
   The TOE will validate the Signer-Keys blob using HMAC validation.
6. If an account gets locked by the TOE after several authentication failures (relevant only for administrators), the TOE makes sure that only Users administrator can unlock the account. Besides unlocking the account, the TOE makes sure that the authenticated Users Administrators does not change any parameter of the account.
7. The TOE makes sure that the authenticated Appliance administrator can perform several administrative functions such as setting system parameters, uploading trusted RSA certificates or RSA public keys or downloading TOE backup.
8. The TOE makes sure that beside the above operations no other operations are permitted as well as setting any attribute.
   In particular, all signer keys are used only for digital signature operation and the is no operation that changes the attributes of the signature keys.
   The TOE makes sure that only the relevant signer can use his/her key for a digital signature operation.
   In the case of the Critical keys, the TOE makes sure that the keys are used only for their internal intended purpose.
9. Only the SSA Admin is allowed to start a transaction on behalf of the signer by uploading a DTBS/R, getting a transaction ID and having the signer use the transaction ID to bind to the originally loaded DTBS/R for signing.
   The TOE makes sure that the operation is allowed only for a signer that was created by the TOE and for a key that was generated by the TOE for the specific signer.
   The TOE will validate the Signer-Keys blob using HMAC validation.
10. Signers’ and its signature keys can be stored outside the TOE in a protect manner in both confidentiality (signature key is encrypted) and integrity. The information is loaded to the TOE prior to the supply-DTBSR and signature operation.

11. As the transaction is deleted when the digital signature is performed (either by the TOE itself or either by a command sent by the SSA), it is not possible to replay a digital signature operation.

7.2 Identification and Authentication (TSF.IA)

1. The TOE identifies users by means of a unique user identifier sent by the user during authentication. Each user can have the following roles: “Appliance Administrator“ (R.ApplianceAdmin), “Users Administrator” (R.UserAdmin) , “SSA Admin” (R.SSA), or “Signer” (R.Signer).
   A user can have a single role.
2. SAML Token authentication is based on a fixed list of trusted RSA certificates or trusted RSA public keys.
   Also, the SAML token includes the DTBS/R or transaction ID for getting the internally kept DTBS/R. The created digital signature will be based on the given DTBS/R.
   Any validation failure will raise the consecutive failure counter of the user.
   A validation will authorize using the signature key once. Re-authorization is required after the current JWT token expired.
3. Administrators are authenticated only using a static password.
   Any validation failure will raise the consecutive failure counter of the user.
4. The TOE provides protection of authentication information by locking the account after a predefined number of consecutive failed authentication attempts.
5. Administrator role is assigned to a user after successful authentication if and only if that role is allowed for the user in the TOE’s persistent storage.
6. As part of the accessing any sensitive entity such as the user account, an RSA key or a system parameter, the integrity of the entity is checked. This is done using the special Data Integrity server master key that is used for AES256-MAC verification operation. Upon failing to check the integrity of the entity, the relevant operation will fail. For example, when the user tries to login and MAC is invalid, the user will not be able to login and thus cannot continue with any operation such as digital signature.

7.3 Cryptographic Operation (TSF.Crypto)

1. The TSF generate 2048, 3072 or 4096 bit cryptographic RSA keys. Random numbers for key generation are provided by an internal RNG which is seeded by a true (physical) random source. This function is compliant with the specifications for random numbers and RSA key generation as specified in [6], [9] and [16].
2. Also, the TSF generate AES256 keys or Shared Secrets. The generated AES256 keys are compliant with specification [14]. The generated keys or generated secrets are located inside the tamper device. Key parts are kept in backup tokens in a dual control manner.

3. When a sensitive data item is deleted, the TSF zeroize the data. This applies to the following sensitive data items: users private RSA keys, RAD in persistent storage, symmetric keys and the users passwords data in volatile storage.
4. The Signature keys are encrypted only with the global master key – MK-KEK.
5. The TSF performs RSA digital signature-generation according to PKCS1 v1.5 (padding scheme EMSA-PKCS1-v1_5) [10] or PSS (padding scheme EMSA-PSS) with 2048, 3072 or4096 bit keys as specified in [6] and [9]. The DTBS/R is sent by the SCA to the TOE. In the case that a DTBS-Representation should be sent, a hash-value of the DTBS is send to the TOE. The hash value is calculated by the SCA.
   Signature keys are used only for the purpose of digital signature.
   The DTBS-representation is based on performing a hash upon the DTBS using one of the following algorithms: SHA-2 family (SHA-256, SHA-384, SHA-512), which are compliant with [6].
6. The SHA-256 of the static password and an administrator user's salt is kept on the Users Database and is used for the static password validation.
7. The following signature suites that are described in [6] are supported:
   - sha256-with-rsa

   With RSA key sizes of 2048, 3072 and 4096 bits.
   In Addition, also the following signature suites are supported by the DocuSign QSCD:
   - sha384-with-rsa
   - sha512-with-rsa

   With RSA key sizes of 2048 or 4096 bits.
   Signature keys are used only for the purpose of digital signature.
8. For every RSA key that is generated by the TSF, a following seed is used:
   - RSA 2048 – 100 bit seed
   - RSA 3072 – 100 bit seed
   - RSA 4096 – 100 bit seed

   The RSA key generation algorithm is based on [9] and is compliant with [6].
9. A public exponent of $2^{16}+1$ should be used to be compliant with [6].
10. The Backup file of the Appliance is encrypted with a AES256 Master Key. The Integrity of the Backup file is based on a calculating a MAC based on a AES256 Master Key.
11. Special HMAC based secrets are used to protect variant records in the database or Signer-Keys blob. For example, each user record, key record, transaction record among others is HMACed using this Master key so that any external modification to the record in the database can be traced.
12. Each software element in the TOE as well as any updated software is digitally signed with a 2048bit RSA key. This signature is validated at module startup or upon uploading an updated version.

7.4 Secure communication and session management(TSF.Comm)

1. The main communication between the clients and the TSF is always secure and no un-secured communication from external applications is allowed by the TOE. This communication is implemented using the TLS [8] [15] and [17] protocol. This secure communication guarantees the secrecy and data integrity of the messages to and from the TOE as well as the authentication of the TOE to the external application, which is based on the TLS protocol.
   There are two different client communication channels:
   - **TLS communication – regular client**
     The TLS server key and its matching certificate are loaded as part of the TOE manufacturing process. During manufacturing process, the TLS server key is generated outside the boundary of the TOE and uploaded to the TOE. During manufacturing, a matched TLS server certificate is uploaded to the TOE as well.
     The Appliance Administrator can configure the minimal TLS version that can be accepted by the Appliance from the following options: Minimum TLS v1.2.
     In this type of communication, right after the TLS session is established, the user is authenticated based on a User ID and a password. Depending on the user's information in the TOE's DB the user type (e.g. Users Administrator) the user's permissions are determined.

   The SSA as well as the administration PCs interacts with the TOE through the same TLS communication as the client.
2. Special mechanisms ensure that no sensitive parameter such as static password or SCD value can be available in a process memory to other user's session than the Signer.

7.5 Auditing

1. The TOE audits all security related events and send them to an external audit log server.
   Every entry in the log file includes date and time.
   The internal motherboard of the system includes an internal clock that is used queried to attach the current time to the relevant event.
2. The system time is synchronized with an external NTP server.
3. As there is a local cache mechanism of audit logs before the logs are transmitted to the Audit log server, there is a configurable maximum limitation to the size of the local cache of audit logs. If the amount of local audit log exceeds the maximum, the system will stop its operation.
   The Audit Logs that are sent from the TOE to the Audit Logs Server are signed.

7.6 Tamper detection & protection (TSF.Tamper)

1. The TOE implements the security function that resists physical tampering. The TOE hardware detects the physical tampering (opening of the TOE enclosure), actively erases sensitive data, and terminates main power. This ensures that the assets are not violated.
   During tamper state all functionality of the TOE is stopped and no service is provided (both Signer ones and administrative ones) even if the TOE is hardware restarted.
   When the TOE is hardware restarted it will maintain the tamper state such that the previous tamper condition can be reported.
2. Only after the tamper reason is deeply analyzed, the Appliance administrator can reset the tamper state by using a special reset tamper operation and providing the backup USB token.
3. The TSF shall ensure that the LAN interface cannot be used to gain access to RAD and SCD.

7.7 Self tests(TSF.Test)

The TSF provides a suite of the following self tests:

1) Start-up tests:
   a) Hardware POST (Power On Self Tests)
   b) Test for a previous tamper event
   c) Test integrity of executable code by verifying its digital signature

2) Tests run while TOE is operational and providing digital signature service:
   a) Encrypt-decrypt integrity test for each RSA key generated
   b) Test the output of the RNG in compliance with [16].
   c) Test integrity of the user account when read from persistent storage. This is done using the special Data Integrity server master key that is used for HMAC verification operation.
   d) Detecting environmental conditions such as temperature

If any of the start-up tests fail, the TOE will NOT enter operational state. If any of the continuous tests fail, the suspect data will not be used.

7.8 Appliance admin functions (TSF.Admin)

The TSF provides the following administrative functions:

1) **Configure system parameters**

The TOE makes sure that the Appliance administrator can configure variant of system parameters. These parameters refine the functionality of the TOE. The set of networking related parameters such as the IP address of the Appliance are not part of the system parameters.

2) **Upload Software**

The TOE makes sure that the Appliance administrator can upload software updates into the TOE.

3) **Appliance backup**

The TOE makes sure that the Appliance administrator can backup Appliance data to an encrypted file with data integrity measures.
The backup of the Appliance does not contain any Signer's or Signature keys related information.

4) **Appliance restoration**

The TOE makes sure that the Appliance administrator can restore an Appliance based on having the Appliance administrator provide a valid Backup File as input to the restoration operation.

5) **Uploading trusted RSA certificates or RSA public keys** for the purpose of SAML ticket validation.

7.9 Rationale for TSF

The following table gives the mapping of the TOE Security Functional Requirements as specified in cha7pter 6.1 and the TSF described above. The numbers in the table specify the component of the TSF which covers the requirement.

SFR \ TSF	ACC	IA	Crypto	Comm	Auditing	Tamper	Test	Admin
FAU_GEN.1					1			
FAU_GEN.2					1			
FAU_STG.2					3			
FCS_CKM.1/SIGNATURE-KEY			1,2,4,5,7,8,9,12				2a	
FCS_CKM.1/SYMMETRIC-KEY			2,4,11				2c	
FCS_CKM.4			3					
FCS_COP.1/SIGNING			1,2,4,5,7,8,9,12				2a	
FCS_COP.1/DATA-INTEG			2,4,11				2c	
FCS_COP.1/KEY-ENCRYPTION			2,4					
FCS_COP.1/BKP-DATA-INTEGRITY			2,11					
FCS_COP.1/BKP-ENCRYPTION			2,11					
FCS_COP.1/ADMIN-SESSION-DATA-INTEG			2,11					
FCS_COP.1/FIRM-UPD			5,7,8,9,12					
FCS_COP.1/TLS-SESSION			5,7,8,9	1				
FCS_RNG.1	1,2,9		1					
FDP_ACC.1/Privileged-User-Creation	1							
FDP_ACC.1/Signer-Creation	1							
FDP_ACC.1/Signer-Maintenance	2,3							
FDP_ACC.1/Signer-Key-Pair-Generation	2							
FDP_ACC.1/Signer-Key-Pair-Deletion	3,4							

SFR \ TSF	ACC	IA	Crypto	Comm	Auditing	Tamper	Test	Admin
FDP_ACC.1.1/Supply-DTBS/R	12							
FDP_ACC.1.1/Signing	5							
FDP_ACC.1/TOE-Maintenance	7							1
FDP_ACC.1/Backup	7							3
FDP_ACC.1/KeyUsage	8		5,7					
FDP_ACF.1/Privileged-User-Creation	1							
FDP_ACF.1/Signer-Creation	1							
FDP_ACF.1/Signer-Maintenance	2,3							
FDP_ACF.1/Signer-Key-Pair-Generation	2							
FDP_ACF.1/Signer-Key-Pair-Deletion	3,4							
FDP_ACF.1.1/Supply-DTBS/R	12							
FDP_ACF.1.1/Signing	5							
FDP_ACF.1/TOE-Maintenance	7							1
FDP_ACF.1/Backup	7							3
FDP_ACF.1/KeyUsage	8		5,7					
FDP_ETC.2/Signer	2,10							
FDP_ETC.2/Privileged-User – Not Relevant								
FDP_IFC.1/KeyBasics	8		5,7					
FDP_IFC.1/Signer-Flow	1,2,3,5,8,9							
FDP_IFC.1/Privileged-User-Flow	1,2,3,4,6,7,8,9							
FDP_IFF.1/KeyBasics	8		4,5,7					
FDP_IFF.1/Signer-Flow	1,2,3,5,8,9							
FDP_IFF.1/Privileged-User-Flow	1,2,3,4,6,7,8,9							

SFR \ TSF	ACC	IA	Crypto	Comm	Auditing	Tamper	Test	Admin
FDP_ITC.2/Signer	3,4,5,6,7,8,9,10							
FDP_SDI.2	10		11					
FDP_UCT.1	10		4					
FDP_UIT.1	10		11					
FDP_RIP.1	5,9					1		
FIA_AFL.1		2,3,4,5,6						
FIA_ATD.1	1,2,3,5,9		11					
FIA_UAU.1		1		1				
FIA_UAU.5/Signer		1,2,3,4,5		1				
FIA_UAU.5/Privileged-User		1,2,3,4,5		1				
FIA_UAU.6		2						
FIA_UID.1		1,2,3,4,5		1				
FIA_UID.2		1,2,3,4,5						
FIA_USB.1	1,2,3,5,9		11					
FMT_MSA.1/Signer	1,2,3,4,5							
FMT_MSA.1/Privileged-User	1							
FMT_MSA.1/GenKeys	8							
FMT_MSA.1/AKeys	8							
FMT_MSA.2	1,2,3,4,5							
FMT_MSA.3/Signer	1							
FMT_MSA.3/Privileged-User	1							
FMT_MSA.3/Keys	2							
FMT_SMF.1/From-HSM-PP								3,4
FMT_SMF.1/From-SAM-PP	1							1,2,5
FMT_MTD.1/Unblock – not relevant								
FMT_MTD.1/AuditLog					3			
FMT_MTD.1/Modify-TSF-DATA	10							2,5
FMT_SMR.1		1						
FMT_SMR.2		1						
FPT_PHP.1						1	1b	
FPT_PHP.3						1	1b	
FPT_FLS.1							1,2	
FPT_STM.1					1			
FPT_RPL.1	11				1			
FPT_TDC.1		6	11					
FPT_TST_EXT							1,2	

SFR \ TSF	ACC	IA	Crypto	Comm	Auditing	Tamper	Test	Admin
FTP_ITC.1/CM – Trivially satisfied								
FTP_TRP.1/Local – Not Relevant								
FTP_TRP.1/External – Not Relevant								
FTP_TRP.1/SIC				1,3		3		
FTP_TRP.1/SSA				1,3		3		

Table 4- SFR - TSF relationship

8 References

[1] Protection profiles for TSP Cryptographic modules - Part 5 Cryptographic Module for Trust Services. Known as EN 419221-5 :2018 version 1.0

[2] Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing. Known as EN 419241-2 v0.16

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

[4] Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components, April 2017 Version 3.1 Rev. 5. CCMB-2017-04-002

[5] Common Criteria for Information Technology Security Evaluation - Part 3: Security assurance components, April 2017 Version 3.1 Rev. 5. CCMB-2017-04-003

[6] Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms. ETSI TS102 176-1 V2.1.1 2011-07.

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

[8] RFC 2246 - The TLS Protocol, The Internet Society, January 1999

[9] FIPS PUB 186-4, Digital Signature Standard, Federal Information Processing Standards Publication, July 2013.

[10] RSA Laboratories, PKCS #1: RSA Encryption Standard, An RSA Laboratories Technical Note Version v2.1, Revised June 14, 2002

[11] Adi Shamir. “How to Share a Secret”, Communications of the ACM 22.11 (1979), pp. 612–613.

[12] Intentionally left blank

[13] RFC 2865 – RADIUS (Remote Authentication Dial In User Sevice), The Internet Society, June 2000.

[14] FIPS Pub 197 (2001): Advanced Encryption Standard (AES), Federal Information Processing Standards.

[15] RFC 5246 - The Transport Layered Security (TLS) Protocol version 1.2, The Internet Society, August 2008

[16] NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators), June 2015

[17] RFC 4346 - The Transport Layered Security (TLS) Protocol version 1.1, The Internet Society, April 2006

[18] 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

[19] COMMISSION IMPLEMENTING REGULATION (EU) 2015/1502, on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of eIDAS [18], September 2015

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

[21] ETSI TS 119 312 Electronic Signatures and Infrastructures (ESI); Cryptographic Suites, V1.2.1 (2017-05)

[22] ETSI EN 319 411-1 Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements, V1.2.1 (2018-02)

[23] Trustworthy Systems Supporting Server Signing Part 1: General system security requirements. Known as fprEN 419241-1, February 2018

[24] ETSI EN 319 411-2 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, V2.2.2 (2018-04)

[25] FIPS 140-2, Security Requirements for Cryptographic Modules, 2001

[26] COMMISSION IMPLEMENTING DECISION (EU) 2015/1506 of 8 September 2015 laying down specifications relating to formats of advanced electronic signatures and advanced seals to be recognised by public sector bodies pursuant to Articles 27(5) and 37(5) 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.

9 Appendix A – Acronyms

CA	Certificate Authority
CSP	Certificate Service Provider
DI	Directory Independent
DTBS	Data To Be Signed. All electronic data to be signed including a user message and signature attributes
DTBS-representation	Data To Be Signed Representation
DTBS/R	Data To Be Signed or its unique representation. Data received by a secure signature creation device as input in a single signature-creation operation.
EAL	Evaluation Assurance Level
IT	Information Technology
JWT	Jason Web Tokens
KEK	Key Encryption Key
MAC	Message Authentication Code
OTP	One Time Password
PP	Protection Profile
REST	Representational State Transfer
RNG	Random Number Generator
SAD	Signature Activation Data
SAM	Signature Activation Module
SAP	Signature Activation Protocol

SSA	Server Signing Application
SIC	Signer Interaction Component
SCA	Signature Creation application
SCD	Signature Creation Data. Private cryptographic key stored in the (Qualified) Signature Creation Device under exclusive control by the Signer to create a (Qualified) electronic signature
SVD	Signature Validation Data
SF	Security Function
SFP	Security Function Policy
QSCD	Qualified Signature Creation Device (**Qualified** electronic **Signature Creation Device** as defined in [18])
SAML	Security Assertion Markup Language
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	Trust Service Provider
QTSP	Qualified Trust Service Provider