Entrust E.U.

Security Target

Entrust Signature Activation Module

April 20, 2022

Contents

1 Security Target introduction

1.1 Security Target and TOE Reference

Title and version	**Security Target – Entrust Signature Activation Module, version 2.1.**
Issue data	April 2022.
Author	Entrust E.U.
CC version	Common Criteria version 3.1 Release 5.
Evaluated TOE	Entrust SAM, version 1.0.3.
TOE commercial name	Entrust Signature Activation Module, version 1.0.3.

1.2 System Overview

Entrust Signature Activation Module (SAM) is part of the architecture of a Trustworthy System Supporting Server Signing (TW4S). It integrates with a Server Signing Application (SSA) product to provide remote signing/sealing functionality to client applications via APIs or services operated by a TSP (Trust Service Provider) [1].

This document assumes that the SSA product is designed for and conforms to the European Standard [CEN EN 419 241-1], which includes scenarios for SCALs (Sole Control Assurance Level) 1 and 2; this standard is the reference of the [eIDAS] Regulation for server signature systems.

The following picture shows all the components of a TW4S according to [CEN EN 419 241-1] and to [CEN EN 419 241-2].

[1] From here, when reference is made to signature, it means "signature and seal".

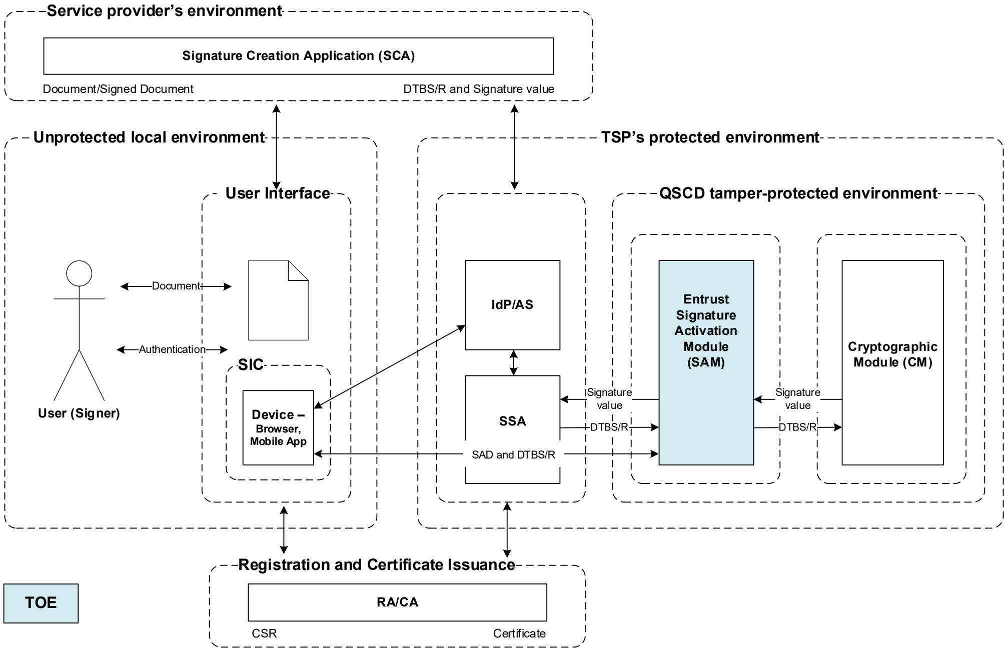

Service provider's environment
Signature Creation Application (SCA)
Document/Signed Document
DTBS/R and Signature value
Unprotected local environment
TSP's protected environment
User Interface
QSCD tamper-protected environment
Document
Authentication
User (Signer)
SIC
Device – Browser, Mobile App
IdP/AS
SSA
Entrust Signature Activation Module (SAM)
Cryptographic Module (CM)
Signature value
DTBS/R
SAD and DTBS/R
Registration and Certificate Issuance
RA/CA
CSR
Certificate
TOE

Figure 1. Architecture of a TW4S

The signing process is initiated by the Signature Creation Application (SCA). In order to do so, it first needs to receive the document to be signed. The document to be signed lifecycle can be summarized as follows:

- Typically the SCA requires the user to log in. One possibility is that it leverages the services provided by the TSP, in the diagram indicated as the Identity Provider and Authorization Server (IdP/AS), or that it does so by other means. The advantage of using the same IdP/AS is that the user can take advantage of single sign-on and get a better user experience. Note that in turn, the IdP/AS can be a single product or two separate products.
- The IdP/AS requires the user authentication for initiating the signature process. It can apply a 2-factor mechanism to reach a level of assurance substantial or high or postpone until the moment of the signature activation the need for the 2nd factor.
- The user provides the document to be signed. The SCA displays the document for the user to review before proceeding to sign it.
- The SCA computes the hash of the document, in the diagram indicated as the Data To Be Signed / Representation (DTBS/R).
- The SCA invokes the SSA signature operation, sending the DTBS/R.
- The IdP/AS explicitly consents/authorizes the user about the access of a concrete signing key to sign the concrete previous calculated DTBS/R. The user completes the 2-factor authentication mechanism to reach a level of assurance substantial or high. Then, the authenticated user can inspect the signing key identifier (keyId) and the hash of the document (DTBS/R). If the user agreed, then it explicitly consents/authorizes the signature by means of the creation of an authorization assertion, the SAD, only valid for

this particular transaction. An authorization token representing the SAD is given to the SSA.

- The SSA verifies the signature request containing the authorization token, requests the SAD to the AS/IdP component and invokes the SAM component.
- The SAM performs the authorized signature by checking the SAD and the DTBS/R. Then, it loads and activates the signing key in the Cryptographic Module (CM), and then encrypting the DTBS/R with this key thus generating the signature.
- The SCA attaches the computed signature into the initial document to be signed to generate the final document.

The Entrust Signature Activation Module software component is the TOE of this Security Target, which implements the Signature Activation Protocol (SAP) to obtain user Signature Activation Data (SAD). The TOE uses the SAD from the signer to activate the corresponding signing key for use in a Cryptographic Module (CM). The TOE uses a Cryptographic Module certified according the Protection Profile [CEN EN 419 221-5], as mandates the standard [CEN EN 419 241-2]. The TOE and the Cryptographic Module are a QSCD as specified in [eIDAS] Regulation.

See section Logical scope of the TOE for more details about the functionality of the TOE.

1.3 TOE Overview

This section briefly describes the usage of the TOE and its major security features, identifies the TOE type and identifies any major non-TOE hardware/software/firmware required by the TOE.

1.3.1 Usage and major security features of the TOE

Entrust Signature Activation Module is a software component that interacts with the Cryptographic Module (CM) in order to implement a Signature Activation Module (SAM) according to the European Standard [CEN EN 419 241-2]. The main objective of the Entrust Signature Activation Module component is to ensure the signer the sole control of their signing keys, which is carried out authorizing the signature operation. The SAM activates the signing key within a CM, handling a Signature Activation Protocol (SAP) which requires that Signature Activation Data (SAD) be provided at the local environment. The SAD binds together the signer authentication with the signing key and the data to be signed. The SAM component uses the SAD in order to guarantee with a high level of confidence -SCAL 2- that the signing keys are used under sole control of the signer.

SCAL 2 also requires the signing keys and the software of the SAM are protected by a tamper protected environment. Entrust Signature Activation Module component uses a dedicated tamper protected environment according to the requirements of [CEN EN 419 241-2] standard.

Entrust Signature Activation Module has been designed and conforms to the European Standard [CEN EN 419 241-2] which is aimed to meet (together with the CM) the requirements of a QSCD as specified in Regulation (EU) No 910/2014 [eIDAS].

Entrust Signature Activation Module product used in combination with a Cryptographic Module (CM) certified according the European Standard [CEN EN 419 221-5]is considered a TW4S for generating Qualified Electronic Signatures/Seals, and therefore it can be used in order to offer an eIDAS Trust Service for electronic remote signature/seal than can be used by a QTSP (Qualified Trust Service Provider).

The major security features of the TOE are the following:

- Identification and authentication of TOE users.
- Secure creation of TOE users.
- Signer Key Pair generation and deletion.
- Supply DTBS/R.
- Signing/Sealing.
- Secure Audit.
- Secure communication between the TOE and the SSA.

1.3.2 Required non-TOE hardware/software/firmware

The TOE relies upon the following IT additional hardware and software:

- Any component or TW4S compliant with [CEN EN 419 241-1] and [CEN EN 419 241-2], and supporting Entrust SAP v1.0, including a SSA component that handles communications between the TOE in the QSCD and the signer device.
- An Identity Provider able to provide signed JSON Web tokens (JWT) as a way to state the authentication and authorization of the users. For instance, TrustedX eIDAS AS/IdP components can achieve this purpose and it may be federated with an external IdP
- Hardware Security Module: Any Hardware Security Module homologated by Entrust E.U. and that is Common Criteria certified against [CEN EN 419 221-5] for instance the nCipher nShield Solo XC product family.
- A signer’s interaction component from which the user interacts with the signature creation application. Typically, it consists in a web browser, a mobile app with an embedded browser or a desktop application with an embedded browser.
- Operating System CentOS 7 or later, or RHEL 7 or later.

1.4 TOE Description

1.4.1 Physical scope of the TOE

In this section a list of software and guidance parts that constitute the TOE is provided.

The TOE is a software composed of several components that is supplied in the file Entrust SAM 1.0.3.zip. This file has the following resources necessary to provide the functionality indicated in the Logical scope of the TOE section:

- Folder “bin”. Binary files in format ELF 64-bit LSB executable, x86-64 GNU/Linux, with the software component Entrust Signature Activation Module, the server component which provides the API (sam file) and the tool which provides the administration procedures (admin file).

- Folder “doc”. Documentation files in which the installation, configuration and operation of the software component are described. The TOE documentation is distributed in two languages: Spanish (folder “es”) and English (folder “en”). Inside these folders is folder “ENTRUST_SAM”, where there are several HTML resources corresponding to documentation referenced as ENTRUST_SAM and version 1.0.3.

Additionally, the ZIP file Entrust SAM 1.0.3 includes document support_1.0.3.txt that indicates how to contact the Technical Support service for any questions or incidents about the TOE.

The values of the hash functions applied to the zip file SAM 1.0.3.zip are as follows:

- MD5: 0a6dd57d6039039e447defdfed46fba8
- SHA1: 2512EE9BD4D51585555ACE4FF4A11F89DBFF31F7
- SHA256: 4BE3403826BEB1A31154CD7E77424205C9497B32183DF0159FC941AD92DDCD54

The TOE is delivered to the final client through a website secure download mechanism.

1.4.1.1 TOE evaluated configuration

Because the TOE supports multiple configurations, this section provides information of the configuration that must be set in the elements included in the physical scope, and that has been evaluated in the Common Criteria certification process related to this Security Target.

In order for the TOE to offer all the security guarantees included in this Security Target, it is necessary to deploy it with the environment components specified in this section, and also the TOE must be configured through the process described in the "Installation" section of the official documentation of the product.

Specifically, the SAM configuration (both the private and public configuration) must force the TOE to meet the Common Criteria requirements included in this Security Target. This is achieved by means of the --common_criteria option of the private configuration, that forces the following:

- The minimum strength required in the authentication of Signer users (the value of the signer_authentication.minimum_loa parameter) can only be "substantial" or "high". If an attempt is made to import a static configuration that does not meet this condition, the import does not occur and an error message is reported.
- The SAM must have the intrusion detection capability of the appliance. i.e., the value of the tamper_resistance.enabled parameter cannot be false. If an attempt is made to import a static configuration that does not meet this condition, the import does not occur and an error message is reported.
- The -v option (it writes the log record generated by the command in the console) cannot be used in any admin command. If it is used, the command is not executed and an error message is reported.

1.4.2 Logical scope of the TOE

In this section the logical security features offered by the TOE are described.

The Entrust Signature Activation Module software component is the TOE of this Security Target, which implements the Signature Activation Protocol (SAP) to obtain user Signature Activation Data (SAD). The TOE uses the SAD from the signer to activate the corresponding signing key for use in a Cryptographic Module (CM). The TOE uses a Cryptographic Module certified according the Protection Profile [CEN EN 419 221-5], as mandates the standard [CEN EN 419 241-2]. The TOE and the Cryptographic Module are a QSCD as specified in [eIDAS] Regulation.

Entrust Signature Activation Module TOE implements the server-side endpoint of the Signature Activation Protocol (SAP). The endpoint is secured with TLS using mutual authentication, i.e. the SAM (server-side) provides strong authentication to the clients (Privileged Users) and it gets the client strongly authenticated. After the TLS handshake, an integral and confidential channel is provided for all communications between the TOE and the clients. In this implementation, Signers are strongly authenticated by using a totally or partially delegated scheme. In the totally delegated authentication scheme, a trusted IdP/AS component authenticates the signers and generates a signed authorization assertion attesting this authentication and the explicit consent of the signer to create a concrete signature; in this scheme, at least two authentication factors are required, and all of them are authenticated by the IdP/AS component. In the partially delegated scheme, at least one authentication factor is managed by the TOE (a user secret) and the rest of the factor/s are delegated to a trusted IdP/AS component.

Over the secure channel, the clients send signature activation protocol data units in order the SAM to process this data, make access control decisions, perform some actions and response with the proper data unit, either in case of the success or error.

From the security point of view, the Entrust Signature Activation Module component has a white list of authorized AS and SSA components. The former in order to allow it to generate and digitally sign the SAD (or authorization assertion), and the latter in order to securely connect to the SAM (using a TLS mutual authenticated connection). In this way, only the SSA components explicitly registered in the SAM can connect and make signature service requests. Also, the SAM will only accept SADs digitally signed by AS's it has previously registered as trusted.

After receiving the signature request, the Entrust Signature Activation Module verifies the SAD and the other data as follows:

- using the AS identifier inside the SAD, the Entrust Signature Activation Module recovers from the white list the trusted public key registered by this AS.
- the Entrust Signature Activation Module verifies the digital signature of the SAD using the previous public key.
- the Entrust Signature Activation Module verifies the time-stamp in the SAD to be fresh.
- the Entrust Signature Activation Module verifies that the total LoA received is substantial or high (at least 2-Factor authentication).

- the Entrust Signature Activation Module verifies that the signer identifier and DTBS/R in the SAD match the same data passed by the SSA component in the request.
- the Entrust Signature Activation Module verifies the integrity of the key pair and verifies that the R.Signing_key_id asset to be used has been authorized in the SAD and also belongs to the authenticated user through the R.Signer asset. The definition of R.Signing_key_id asset and R.Signer asset can be found in the Assets section.

After these checks, the Entrust Signature Activation Module does, as a minimum, the following tasks: (1) loads the signing identity data into the Cryptographic Module (CM) [2], (2) invokes the CM signing function upon the DTBS/R, (3) gets the created signature, (4) removes the signing identity data from the CM, (5) generates a digitally signed audit log attesting that this signature has been generated, and (6) sends back to the SSA component the created signature.

The SSA component receives the PKCS#1 signature from the SAM, and the signature is returned back to the SCA as the result of the signature service.

From a security point of view of the SAP protocol, the following can be concluded:

- Only Privileged Users can access the SAM. In the described system, the SSA components are Privileged Users that mutually authenticate with the SAM and securely exchange data by using a secure connection.
- Only the Signature Activation Module component and delegated IdP/AS components that are trusted by the SAM can authenticate users and authorize the DTBS/R and R.Signing_key_id for a given signature process. The trust is represented by means of a digitally signed authorization assertion (the SAD).
- The Signature Activation Protocol (SAP) is directly executed by the Signature Activation Module and the user through the SSA to obtain the SAD irrespective of the SCA that invokes the signature service. This is thanks to the OAuth2 user-centric three-legged framework.
- The TW4S system in general and the SSA component in particular, runs in a protected environment and operated by a trusted/recognized TSP.

The product allows the safe creation of users, through a process totally linked to the segregation of roles that the product maintains. Basically, these roles consist of the signers and some privileged roles who maintain and operate the product.

From the point of view of functionality, Entrust Signature Activation Module allows the generation and deletion of key pairs of the signers. The key pair generation operation for a Signer has the following two parts: generation of a key pair and assignment of the keys created to a specific signer (after certifying the public key). Regarding the generation, the explicit authorization of the signer is not necessary, allowing cases of use such as pre-generation of keys. A key pair assigned to a signer cannot be assigned to another user, thus respecting the uniqueness of the signers' keys. Regarding the removing of key pairs of a signer, both an specific privileged user (accepted in the system, and authenticated by the TLS protocol with client authentication) and the signer who own the key pair can request

[2] The private key component (SCD) of the signing identity can only be opened inside the CM since it has been protected by the CM when created/registered in the system.

the deletion of these signing key pair. When a key pair is deleted, the product invokes security mechanisms to ensure that the keys must no longer be used.

Entrust Signature Activation Module generates audit records for all the security events involved in it operation. The product issues and signs an event log for any given service request, and then it stores them in a permanent repository. Audit records can be verified and securely stored externally by the SSA.

2 Conformance Claims

The present Security Target conforms to the following assurance and functional requirements:

- General model of the “Part 1: Introduction and general model” of the Common Criteria Standard. April 2017. Version 3.1. Revision 5.
- Functional Requirements of the “Part 2: Security functional components” and Part 2 Extended of the Common Criteria Standard. April 2017, Version 3.1, Revision 5.
- Assurance Requirements of the “Part 3: Security assurance components”. April 2017, Version 3.1, Revision 5, for the **EAL4 Common Criteria certification level, augmented with ALC_FLR.2 and AVA_VAN.5**.

This security target claims a strict conformance with **Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing Protection Profile**, version 0.16, May 11, 2018, European Committee for Standardization (CEN) TC 224.

2.1 Conformance Rationale

The TOE type described in the Security Target and TOE Reference section of this document is consistent with the TOE type described in the [CEN EN 419 241-2] Protection Profile. As indicated in the [CEN EN 419 241-2] document, this TOE type basically consists of a Signature Activation Module (SAM) that assures the signer the sole control of their signing keys in a signature operation. To achieve this goal the SAM activates the signing key within a Cryptographic Module, handling a Signature Activation Protocol (SAP) which requires that Signature Activation Data (SAD) be provided at the local environment. The SAM component uses the SAD in order to guarantee with a high level of confidence -SCAL 2- ([CEN EN 419 241-1]) that the signing keys are used under sole control of the signer.

Because the security problem definition of this Security Target document is exactly the same as the security problem definition of the [CEN EN 419 241-2] Protection Profile, it is consistent with the statement of the security problem definition of this Protection Profile. All the assumptions, threats and security policies included in this Security Target are the same as the assumptions, threats and security policies included in the [CEN EN 419 241-2] PP.

Likewise, the security objectives of the [CEN EN 419 241-2] Protection Profile have been represented exactly in this Security Target document.

The security requirements included in the [CEN EN 419 241-2] Protection Profile have been reproduced strictly in this document.

3 Security Problem Definition

This section includes the assets, subjects, secure usage assumptions, threats and organizational security policies that are part of the security problem definition. At the end of this section, a relation between threats and assets is also included.

This information provides the basis for the Security Objectives specified in chapter Security Objectives, and for the Security Requirements for the TOE specified in chapter Security Requirements.

3.1 Assets

The TOE has the following assets, which are to be protected in integrity and confidentiality as described below. The TOE ensures 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 is enforced by the environment.

R.Signing_Key_Id

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

Application Note 1

For this TOE, this asset is a hash managed by the Cryptographic Module that identifies the signature key. The integrity between the signature key and this hash is guaranteed by the Cryptographic Module.

The integrity and confidentiality of the signing key and the link between the R.Signing_Key_Id and the signing key is the responsibility of the Cryptographic Module. The TOE ensures that only the signer can use the signing key under his sole control.

R.Authorisation_Data

This asset is data used by the TOE to activate a signing key in the Cryptographic Module. The signing key is identified by R.Signing_Key_Id. It is protected in integrity and confidentiality. Confidentiality is only applicable for confidential data (secret of the signer) and is guaranteed by encryption using a SAM infrastructure key.

Application Note 2

The asset R.Authorisation_Data consists basically of the signed assertion (SAD) and all the necessary data to ensure that the assertion comes from an authorized Authorization Server (Authorization Server public keys). The TOE verifies the SAD assertion before the R.Authorisation_Data is used to activate the signing key in the Cryptographic Module.

R.SVD

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

The TOE uses a Cryptographic Module for signing key pair generation. As part of the signing key pair generation, Cryptographic Module provides the TOE with R.Signing_Key_Id and R.SVD. The TOE provides the R.SVD to the SSA for further handling of the key pair when generating the corresponding X.509 certificate.

R.DTBS/R

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

Application Note 3

The confidentiality of the R.DTBS/R is not required by Regulation (EU) No 910/2014 [eIDAS].

R.SAD

Signature activation data is a set of data involved in the signature activation protocol, which activates the signature creation data to create a digital signature under the signer's sole control. The R.SAD combines 1) the signer's strong authentication as specified in [CEN EN 419 241-1], 2) If a particular key is not implied (e.g a default or one-time key) a unique reference to R.Signing_Key_Id, and 3) a given R.DTBS/R.

The R.SAD is protected in integrity and confidentiality.

Application Note 4

The R.SAD may include some or all evidences from other systems that all authentication factors have been verified.

Application Note 5

The R.SAD can include a user secret (user credential) that activate the user´s signing key. This part of the SAD is protected in confidentiality by an infrastructure asymmetric key specific for the encryption of SAD's confidential data.

Application Note 6

The unique reference to R.Signing_Key_Id in the R.SAD is the key identifier of the authorized key that will sign the DTBS structure.

Application Note 7

This asset is the signature activation data (SAD) generated by the signer during signature activation protocol (SAP) execution in order the SAM can activate the signing key to produce a digital signature under the sole control of the signer.

In this implementation the R.SAD data includes the following:

- *Signer's authentication data. It is included a reference of the signer's identity and the level of assurance (LoA) achieved. The authentication occurs according [CEN EN 419 241-1] - SCAL2 and a LoA of substantial or high must be achieved in order to generate qualified digital signatures.*
- *A signing key identifier (R.Signing_Key_Id) to reference the signing key that will be used to produce the digital signature.*
- *A sequence (one or several) of R.DTBS/R.*
- *Other protocol-related data, like issuing timestamp, issuer's identifier (specific Authorization Server – AS-), and the public key´s Authorization Server.*
- *Signature of all these previous data.*

The signer authentication can be carried out in one of the following schemes:

- *Indirectly by the TOE. In this case, an external authentication service as part of the TW4S or a delegated party that verifies the signer's authentication factors and issues an assertion that the signer has been authenticated. The TOE verifies the assertion.*
- *A combination of the indirect scheme and a direct scheme in which the TOE verifies a signer's authentication factor. In this case a part of the signer authentication is done directly by the TOE and another part is done indirectly by the TOE.*

When the SAM verifies a signer's authentication factor (mixed scheme), it verifies a secret that the Signer's only knows, and that has been included in the R.SAD asset.

The R.SAD is protected in integrity by a digital signature and, when it contains a signer´s secret, it is encrypted (using the public key of the TOE) for confidentiality.

R.Signature

Is the result of the signature operation and is a digital signature value. R.Signature is created on the R.DTBS/R using R.Signing_Key_Id by the Cryptographic Module under the signer's control as part of the SAP. The R.Signature is protected in integrity. The R.Signature is generated within the QSCD tamper-resistant environment, and it can be verified outside the TOE using the R.SVD.

R.Audit

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

Application Note 8

This asset is a sequence of event logs produced by the TOE for any given action or event caused in it. Audit records can be verified and securely stored externally by the SSA.

R.Signer

Is a TOE subject containing the set of data that uniquely identifies the signer within the TOE. The R.Signer is protected in integrity (electronic signature by the TOE), and it does not require encrypted data (it does not contain confidential data).

Application Note 9

The R.Signer asset is formed by one or several sign_identity_ids (R.Signing_Key_Ids) that uniquely identify the signer within the TOE. The R.Signer asset includes the R.Signing_key_id enabled for data signing by this SAM. If a R.Signing_key_id is not enabled by the R.Signer asset for a particular user, this user will not be able to activate this key.

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.

R.Reference_Signer_Authentication_Data

Is the set of data used by TOE to authenticate the signer. It contains all the data and keys used by the TOE to authenticate the signer. This may include a SVD or X.509 certificate to verify an assertion provided as a result of delegated authentication.

The R.Reference_Signer_Authentication_Data is protected in integrity and confidentiality.

Application Note 10

The R.Reference_Signer_Authentication_Data is used by the TOE to authenticate the signer, and the R.Authorisation_Data is used by the TOE to activate a signing key in the Cryptographic Module.

Application Note 11

In this implementation, Signers are strongly authenticated by using a delegated scheme (in which a trusted IdP/AS component authenticates the signers and generates a signed authorization assertion) or by using a mixed scheme in which the TOE verifies a secret that the Signer only knows. When the Signer's secret is used as an authentication factor, then it must be protected in confidentiality.

In this implementation, the R.Reference_Signer_Authentication_Data asset are the public keys of the Authorization Server component, and optionally the secret of the user (as an authentication factor, if this is used).

R.TSF_DATA

Is the set of TOE configuration data used to operate the TOE. It is protected in integrity.

Application Note 12

In this implementation, this asset are the static and dynamic configuration files of the SAM. The dynamic configuration is digitally signed by privileged users to guarantee integrity, and the access to the static configuration is strictly protected because only the First Privileged User can access it.

R.Privileged_User

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

Application Note 13

In this implementation the R.Privileged_User asset is a string representation of the privilege user and all their keys that are used to authenticate.

The TOE differentiates the following types of Privileged User: 1) First Privileged User (FPU), 2) Configuration Privileged User (CPU), 3) Operation Privileged User (OPU) and 4) Signing Service Privileged User (SSPU). More information about Privileged Users can be found in section Security Management*.*

R.Reference_Privileged_User_Authentication_Data

Is the set of data used by the TOE to authenticate the Privileged User. It is protected in integrity and confidentiality.

Application Note 14

The T.Reference_Privileged_User_Authentication_Data are the keys of the Privileged Users and a reference to the keys associated with the OCSs of the Privileged Users FPUs (in the creation of this asset, the role is assigned to this key).

All this asset is protected in integrity and for non public keys also in confidentiality.

R.Random

This asset represents the random secrets used by the TOE to operate, communicate and develop trust to external parties. It is protected in integrity and confidentiality.

Application Note 15

The infrastructure keys are part of this asset. The confidentially of these secrets is based on the HSM.

3.2 Subjects

This following list of subjects interacts with the TOE.

- Signer, which is the natural or legal person who uses the TOE through the SAP where they provide the SAD and can sign DTBS/R(s) using their 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 Note 16

The list of subjects described in [CEN EN 419 241-1] clause 6.2.1.2 SRG M.1.2 contains more roles as it covers the whole TW4S. More information about Privileged Users can be found in section Security Management*.*

Application Note 17

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 [ETSI EN 319 411-1].

3.3 Secure Usage Assumptions

A.PRIVILEGED_USER

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

A.SIGNER_ENROLMENT

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

A.SIGNER_AUTHENTICATION_DATA_PROTECTION

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

A.SIGNER_DEVICE

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

A.CA

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

A.ACCESS_PROTECTED

It is assumed that the TOE operates in a protected environment that limits physical access to the TOE to authorized 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 is only handled by authorized 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 Note 18

The protected data that are stored outside the TOE, can only be used within the QSCD tamper protected environment.

A.AUTH_DATA

It is assumed that the SAP is designed in such a way that the activation of the signing key is under sole control of the signer with a high level of confidence. If SAD is received by the

TOE, it must be assumed that the SAD was submitted under the full control of the signer by means that are in possession of the signer.

A.TSP_AUDITED

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

A.SEC_REQ

It is assumed that the TSP establishes an operating environment according to the security requirements for SCAL2 defined in [CEN EN 419 241-1].

3.4 Threats

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

3.4.1 Enrolment

The threats during enrolment are:

T.ENROLMENT_SIGNER_IMPERSONATION

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

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

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

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

T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED

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

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

The asset R.Reference_Signer_Authentication_Data is threatened.

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

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

T.SVD_FORGERY

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

The asset R.SVD is threatened.

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

Application Note 19

There is a secure transport of R.SVD from TOE to RA or CA. The SAM generates 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 [ETSI EN 319 411-1] clause 6.3.1 a), this threat can be countered without any specific measures within the TOE.

3.4.2 Signer Management

T.ADMIN_IMPERSONATION

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

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

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

T.MAINTENANCE_AUTHENTICATION_DISCLOSE

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

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

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

3.4.3 Usage

This section describes threats for signature operation including authentication.

T.AUTHENTICATION_SIGNER_IMPERSONATION

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

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

T.SIGNER_AUTHENTICATION_DATA_MODIFIED

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

The asset R.Reference_Signer_Authentification_Data is threatened.

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

T.SAP_BYPASS

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

The asset R.SAD is threatened.

T.SAP_REPLAY

An attacker replays one or more steps of SAP and is able to create a signature without the signer having authorized the operation.

The asset R.SAD is threatened.

T.SAD_FORGERY

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

The asset R.SAD is threatened.

T.SIGNATURE_REQUEST_DISCLOSURE

An attacker obtains knowledge of R.DTBS/R or R.SAD during transfer to TOE.

The assets R.DTBS/R and R.SAD are threatened.

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

T.DTBSR_FORGERY

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

The asset R.DTBS/R is threatened.

T.SIGNATURE_FORGERY

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

The asset R.Signature is threatened.

Application Note 20

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

3.4.4 System

T.PRIVILEGED_USER_INSERTION

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

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

T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION

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

The asset R.Reference_Privileged_User_Authentication_Data is threatened.

T.AUTHORISATION_DATA_UPDATE

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

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

Application Note 21

Access to R.Authorisation_Data is only allowed for authorized operators.

T. AUTHORISATION_DATA _DISCLOSE

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

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

T.CONTEXT_ALTERATION

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

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

T.AUDIT_ALTERATION

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

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

T.RANDOM

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

3.5 Organizational Security Policies

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

OSP.RANDOM

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

OSP.CRYPTO

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

Application Note 22

For cryptographic algorithms within the European Union this is as indicated in [eIDAS] and an exemplary list of algorithms and parameters is given in [ETSI TS 119 312] or [SOGIS].

3.6 Relation between threats and assets

This following table provides an overview of the relationships between asset, associated security properties and threats. For details consult the individual threats in the previous sections.

Asset	Security Dimensions	Threats
R.Signing_Key_Id	Integrity	T.ADMIN_IMPERSONATION T.MAINTENANCE_AUTHENTICATION_DISCLOSE T.AUTHENTICATION_SIGNER_IMPERSONATION T.CONTEXT_ALTERATION
R.Authorisation_Data	Integrity	T.AUTHORISATION_DATA_UPDATE

	Confidentiality	T.AUTHORISATION_DATA_UPDATE T.AUTHORISATION_DATA _DISCLOSE
R.SVD	Integrity	T.SVD_FORGERY T.ADMIN_IMPERSONATION T.CONTEXT_ALTERATION T.AUDIT_ALTERATION
R.DTBS/R	Integrity	T.SIGNATURE_REQUEST_DISCLOSE T.DTBSR_FORGERY
	Origin authentication	T.DTBSR_FORGERY
R.SAD	Integrity	T.AUTHENTICATION_SIGNER_IMPERSONATION T.CONTEXT_ALTERATION T.AUDIT_ALTERATION T.SAP_BYPASS T.SAP_REPLAY T.SAD_FORGERY
	Confidentiality	T.AUTHENTICATION_SIGNER_IMPERSONATION T.DTBSR_FORGERY T.CONTEXT_ALTERATION
R.Signature	Integrity	T.SIGNATURE_FORGERY
R.Audit	Integrity	T.AUDIT_ALTERATION
R.Signer	Integrity	T.ENROLMENT_SIGNER_IMPERSONATION
R.Reference_Signer_Authentication_Data	Integrity	T.ENROLMENT_SIGNER_IMPERSONATION T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED

		T. SIGNER_AUTEHNTICATION_DATA_MODIFIED T.ADMIN_IMPERSONATION T.MAINTENANCE_AUTHENTICATION_DISCLOSE T.AUTHENTICATION_SIGNER_IMPERSONATION T.CONTEXT_ALTERATION T.AUDIT_ALTERATION
	Confidentiality	T.ENROLMENT_SIGNER_IMPERSONATION T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED T.SIGNER_AUTEHNTICATION_DATA_MODIFIED T.ADMIN_IMPERSONATION T.MAINTENANCE_AUTHENTICATION_DISCLOSE T.AUTHENTICATION_SIGNER_IMPERSONATION T.CONTEXT_ALTERATION
R.Privileged_User	Integrity	T.PRIVILEGED_USER_INSERTION T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION
R.Reference_Privileged_User_Authentication_Data	Integrity	T.PRIVILEGED_USER_INSERTION T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION
	Confidentiality	T.PRIVILEGED_USER_INSERTION T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION
R.RANDOM	Integrity	T.RANDOM
	Confidentiality	T.RANDOM

R.TSF_DATA	Integrity	T.CONTEXT_ALTERATION T.AUDIT_ALTERATION

Table 1. Relation between threats and assets

4 Security Objectives

This chapter identifies and defines the security objectives for the TOE and its environment. Security objectives reflect the stated intent and counter the identified threats, as well as comply with the identified organizational security policies and assumptions.

4.1 Security objectives for the TOE

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

4.1.1 Enrolment

OT.SIGNER_PROTECTION

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

OT.REFERENCE_SIGNER_AUTHENTICATION_DATA

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

OT.SIGNER_KEY_PAIR_GENERATION

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

OT.SVD

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

4.1.2 User Management

OT.PRIVILEGED_USER_MANAGEMENT

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

OT.PRIVILEGED_USER_AUTHENTICATION

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

Application Note 23

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

OT.PRIVILEGED_USER_PROTECTION

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

OT.SIGNER_MANAGEMENT

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

4.1.3 Usage

OT.SAD_VERIFICATION

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

Application Note 24

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.

Application Note 25

Requirements for authentication are described in [CEN EN 419 241-1] SRA_SAP.1.1.

OT.SAP

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

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

Application Note 26

The signer authentication is assumed to be conducted according to [CEN EN 419 241-1] SCAL2 for qualified signatures. In this implementation, the signer authentication scheme can be a delegated scheme (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 TOE verifies the assertion) or a mixed scheme (external factor plus an signer's secret that is verified by the TOE).

OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION

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

OT.DTBSR_INTEGRITY

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

OT.SIGNATURE_INTEGRITY

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

OT.CRYPTO

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

4.1.4 System

OT.RANDOM

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

OT.SYSTEM_PROTECTION

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

OT.AUDIT_PROTECTION

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

4.2 Security Objectives for the Operational Environment

OE.SVD_AUTHENTICITY

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

OE.CA_REQUEST_CERTIFICATE

The operational environment shall ensure that the qualified TSP that issues qualified certificates is compliant with the relevant requirements for qualified TSP's as defined in [eIDAS].

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

OE.CERTIFICATE_VERFICATION

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

OE.SIGNER_AUTHENTICATION_DATA

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

OE.DELEGATED_AUTHENTICATION

If the TOE has support for and is configured to use delegated authentication then the TSP deploying the SSA and TOE shall ensure that all requirements in [CEN EN 419 241-1] SRA_SAP.1.1 are met.

In addition, the TSP shall ensure that:

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

If the signer is only authenticated using a delegated party, the TSP shall ensure that the secret key material used to authenticate the delegated party to the TOE shall reside in a certified cryptographic module consistent with the requirement as defined in [CEN EN 419 241-1] SRG_KM.1.1.

The audit of the qualified TSP according to CEN EN 419 241-1 shall provide evidence that any delegated party meets requirements from CEN 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.

OE.DEVICE

The device, computer/tablet/smart phone containing the SIC and which is used by the signer to interact with the TOE shall be protected against malicious code. It shall participate using SIC as local part of the SAP and may calculate SAD as described in [CEN EN 419 241-1]. It may be used to view the document to be signed.

OE.ENV

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

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

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

OE.CRYPTOMODULE_CERTIFIED

The TOE is implemented within a separate physical boundary as the cryptographic module defined in [EN 419 221-5], then the TOE relies on the cryptographic module for cryptographic functionality and random number generation. The physical boundary shall physically protect the TOE conformant to FPT_PHP.1 and FPT_PHP.3 in [EN 419 221-5].

OE.TW4S_CONFORMANT

The TOE shall be operated by a qualified TSP in an operating environment conformant with [CEN EN 419 241-1].

4.3 Rationale for the Security Objectives

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

4.3.1 Security Problem Definition and Security Objectives

The following tables map security objectives with the security problem definition.

	Enrolment	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_D	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD
Enrolment					
T.ENROLMENT_SIGNER_IMPERSONATION		X	X		

T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED		X	X		
T.SVD_FORGERY				X	X
Signer Management					
T.ADMIN_IMPERSONATION					
T.MAINTENANCE_AUTHENTICATION_DISCLOSE			X		
Usage					
T.AUTHENTICATION_SIGNER_IMPERSONATION					
T.SIGNER_AUTHENTICATION_DATA_MODIFIED			X		
T.SAP_BYPASS					
T.SAP_REPLAY					
T.SAD_FORGERY					
T.DTBSR_FORGERY					
T.SIGNATURE_FORGERY					
System					
T.AUTHORISATION_DATA_UPDATE					
T.AUTHORISATION_DATA_DISCLOSE					
T.CONTEXT_ALTERATION					
T.AUDIT_ALTERATION					
T.RANDOM					

Table 1. TOE Security Objectives and threats I

	User Management	OT.PRIVILEGED_USER_MANAGEMENT	OT.PRIVILEGED_USER_AUTHENTICATION	OT.PRIVILEGED_USER_PROTECTION	OT.SIGNER_MANAGEMENT
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.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					
T.AUTHORISATION_DATA_DISCLOSE					
T.CONTEXT_ALTERATION					
T.AUDIT_ALTERATION					
T.RANDOM					

Table 2. TOE Security Objectives and threats II

	System	OT.RANDOM	OT.SYSTEM_PROTECTION	OT.AUDIT_PROTECTION
Enrolment				
T.ENROLMENT_SIGNER_IMPERSONATION				
T.ENROLMENT_SIGNER_AUTHENTICATION_DATA_DISCLOSED				
T.SVD_FORGERY				
Signer Management				
T.ADMIN_IMPERSONATION				
T.MAINTENANCE_AUTHENTICATION_DISCLOSE				
Usage				
T.AUTHENTICATION_SIGNER_IMPERSONATION				
T.SIGNER_AUTHENTICATION_DATA_MODIFIED				
T.SAP_BYPASS				
T.SAP_REPLAY				
T.SAD_FORGERY				
T.DTBSR_FORGERY				
T.SIGNATURE_FORGERY				
System				
T.PRIVILEGED_USER_INSERTION				
T.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA_MODIFICATION				
T.AUTHORISATION_DATA_UPDATE			X	
T.AUTHORISATION_DATA_DISCLOSE			X	
T.CONTEXT_ALTERATION			X	
T.AUDIT_ALTERATION				X

T.RANDOM		X		

Table 3. TOE Security Objectives and threats III

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

Table 4. TOE Security Objectives and threats IV

	Enrolment	OT.SIGNER_PROTECTION	OT.REFERENCE_SIGNER_AUTHENTICATION_D	OT.SIGNER_KEY_PAIR_GENERATION	OT.SVD	OT.RANDOM	OT.CRYPTO
OSP.RANDOM						X	
OSP.CRYPTO							X

Table 5. TOE Security Objectives and Organizational Security Policies

	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.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							

Table 6. Threats and Security Objectives for the environment

	OE.SVD_AUTHENTICITY	OE.CA_REQUEST_CERTIFICATE	OE.SIGNER_AUTHENTICATION_DATA	OE.DEVICE	OE.ENV	OE.CRYPTOMODULE_CERTIFIED	OE.TW4S_CONFORMANT
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

Table 7. Security Objectives for the environment and Assumptions and Security Objectives for the environment

4.3.2 Threats and objectives

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

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

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

It is also covered by OE.TW4S_CONFORMANT as that requires signer enrolment to be handled in accordance with [IMPREG 2015/1502] for level at least substantial.

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

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

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

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

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

It is also covered by OT.SVD requiring the SVD to be protected while inside the TOE.

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

It is also covered by OE.SVD_AUTHENTICITY requiring the environment to protect the SVD during transmit from the TOE to the CA.

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

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

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

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

T.SIGNER_AUTHENTICATION_DATA_MODIFIED is covered by OT.SIGNATURE_AUTHENTICATION_DATA_PROTECTION requiring the SAD transported protected in the SAP.

It is also covered by OT.REFERENCE_SIGNER_AUTHENTICATION_DATA requiring that authentication data be securely handled.

It is also covered by OT.SAP requiring the integrity of the SAD is protected during transmit in the SAP.

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

It is also covered by OE.DEVICE requiring the SIC to participate the in SAP.

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

It is also covered by OE.DEVICE requiring the SIC to participate the in SAP.

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

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

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

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

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

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

It is also covered by OE.DEVICE requiring the SIC to participate the in SAP.

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

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

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

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

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

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

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

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

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

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

4.3.3 Organizational Security Policies and Objectives

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

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

4.3.4 Assumptions and objectives

A.PRIVILEGED_USER is covered by OE.TW4S_CONFORMANT which requires that the system where the TOE operates is compliant with [CEN EN 419 241-1] where clause SRG_M.1.8 requires that administrators are trained.

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

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

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

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

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

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

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

A.SEC_REQ is covered by OE.TW4S_CONFORMANT requiring the system where the TOE operates is compliant with [EN 419 241-1].

5 Security Requirements

5.1 Subjects, Objects and Operations

This section describes the subjects, object and operations supported by the TOE

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

Table 1. Subjects supported by the TOE

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

Table2. Objects supported by the TOE

Subject	Operation	Object	Description
R.Privileged_User	Create_New_Privileged_User	R.Privileged_User	A new privileged user can be created which covers the object representing the new privileged user as well as the

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

Table 3. Operations supported by the TOE

5.2 SFRs overview

This section gives an overview of how the SFRs are related to handle TOE usage scenarios and Signer object.

Signer object

- FIA_ATD.1 and FIA_USB.1 requires that the R.Signer object is maintained by the TOE.
- FDP_ITC.2/Signer describes requirements for importing the R.Signer object.
- FDP_ETC.2/Signer describes requirements for exporting the R.Signer object.
- FDP_UIT.1 requires the R.Signer object to be protected in integrity when imported and exported.

- FPT_TDC.1 requires the TOE to be able to interpret R.Signer object related data when shared with SSA.
- FMT_MSA.1, FMT_MSA.2 and FMT_MSA.3 describes rules for creation, maintaining and usage of the R.Signer object as well as requirements to its values.

Authentication

- FDP_UCT.1 ensure that access control and information flow data are transmitted in a confidential way.
- FIA_UID.2 and FIA_UAU.1 requires that each user is identified and authenticated before any action on behalf of the user can take place.
- FIA_UAU.5/Signer and FIA_UAU.5/Privileged User describe the list of authentication mechanism.

Create Signer

- FDP_ACC.1/Signer Creation using FDP_ACF.1/Signer Creation describes access control requirements for creating a R.Signer object.
- FIA_USB.1 defines authorisation rules for creating new R.Signer objects.

Signer Key Pair Generation

- FDP_ACC.1/Signer Key Pair Generation using FDP_ACF.1/Signer Key Pair Generation describes access control requirements for signing key pair generation.
- FCS_CKM.1 describes rules for how signing key pair is generated.

Signer Key Pair Deletion

- FDP_ACC.1/Signer Key Pair Deletion using FDP_ACF.1/Signer Key Pair Deletion describes access control requirements for signing key pair deletion.
- FCS_CKM.4 requires keys to be securely destructed.

Signer Maintenance

- FDP_ACC.1/Signer Maintenance using FDP_ACF.1/Signer Maintenance describes access control requirements for updating the R.Reference_Signer_Authenticaton_Data of a R.Signer object.

Supply DTBS/R

- FDP_ACC.1/Supply DTBS/R using FDP_ACF.1/Supply DTBS/R describes access control requirements for a Privileged User to supply a DTBS/R(s).

Signing

- FDP_IFF.1/Signer and FDP_IFC.1/Signer describing requirements on preconditions for a signature operation can be carried out.
- FDP_UIT.1 requires the R.SAD object to be protected from modification and replay.
- FDP_ACC.1/Signing using FDP_ACF.1/Signing describes access control requirements for signing.
- FCS_COP.1 requires the TOE to perform cryptographic operation conformant with a ST specified list of algorithms.

Privileged User object

- FIA_ATD.1 and FIA_USB.1 requires that the R.Privileged User object is maintained by the TOE.
- FDP_ITC.2/Privileged User describes requirements for importing the R.Privileged User object.
- FDP_ETC.2/ Privileged User describes requirements for exporting the R.Privileged User object
- FDP_UIT.1 requires the R.Privileged User object to be protected in integrity when imported and exported.
- FPT_TDC.1 requires the TOE to be able to interpret R.Privileged User object when shared with a trusted IT product the SSA.
- FMT_MSA.1, FMT_MSA.2, FMT_MSA.3 describes rules for creation, maintaining and usage of the R.Privileged User object as well as requirements to its values.

Privileged User Creation

- FDP_ACC.1/Privileged User Creation using FDP_ACF.1/Privileged User Creation describes access control requirements for creating a R.Privileged User object.
- FIA_USB.1 defines authorisation rules for creating new R.Privileged User objects.

TOE Maintenance

- FDP_ACC.1/TOE Maintenance using FDP_ACF.1/TOE Maintenance.
- FMT_SMF.1 and FMT_SMF.2 requires the TOE to be able to carry out management functions and maintain users and roles.

Audit

- FAU_GEN.1 and FAU_GEN.2 describes what shall be audited.

Communication

- FPT_ITC.2 requires that all communication to the TOE comes from the SSA.
- FTP_TRP.1/SSA and FTP_TRP.1/SIC requires that either the Privileged User or the Signer initiates the communication.

5.3 Security Functional Requirements

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

5.3.1 Security Audit (FAU)

5.3.1.1 FAU_GEN.1 Audit Generation

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 [selection: *not specified*] level of audit; and

c) Privileged User management;

d) Privileged User authentication;

e) Signer management;

f) Signer authentication;

g) Signing key generation;

h) Signing key destruction;

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

j) Change of TOE configuration;

k) [assignment: *Tampering detection, audit record integrity validation*].

Application Note 27

Management of R.Privileged User and R.Signer objects include all events, which creates, modifies or deletes the R.Signer or R.Privileged User objects.

Signer authentication includes failed verification of an assertion provided by a delegated party.

TOE configuration includes all events, which creates, modifies and deletes the configuration object.

Application Note 28

Generation of a certification request is usage of the signing key and mandates an audit trail.

Application Note 29

In this implementation, the TOE records the R.DTBS/R 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 ST, [assignment: *Type of action performed (success or failure), identity of the role which performs the operation, [*assignment*: none]*].

Application Note 30

Audit trail does not include any data which allow to retrieve sensitive data.

5.3.1.2 FAU_GEN.2 User identity association

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.

5.3.2 Cryptographic Support (FCS)

5.3.2.1 FCS_CKM.1 / RSA - Cryptographic key generation

FCS_CKM.1.1 / RSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *RSA PCKS#1 v.1.5*] and specified cryptographic key sizes [assignment: *2048 bits, 4096 bits*] that meet the following: [assignment: *PKCS#1 RSA Cryptography Standard, FIPS 186-4 Digital Signature Standard (DSS), IETF RFC 3447*].

Application Note 31

The TOE uses a cryptographic module certified in conformance with [EN 419 221-5], see also OE.CRYPTOMODULE_CERTIFIED 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.

Application Note 32

This ST uses cryptographic keys for different purposes. In this sense, an iteration has been included of this SFR for every key type it generates itself.

5.3.2.2 FCS_CKM.1 / EC based DSA - Cryptographic key generation

FCS_CKM.1.1 / EC based DSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *EC based DSA algorithm - Curve family NIST*] and specified cryptographic key sizes [assignment: *ECC prime256v1 bits, prime384v1 bits and prime521v1 bits*] that meet the following: [assignment: *NIST FIPS 186-4*].

Application Note 33

The TOE uses a cryptographic module certified in conformance with [EN 419 221-5], see also OE.CRYPTOMODULE_CERTIFIED 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.

Application Note 34

This ST uses cryptographic keys for different purposes. In this sense, an iteration has been included of this SFR for every key type it generates itself.

5.3.2.3 FCS_CKM.1 / AES - Cryptographic key generation

FCS_CKM.1.1 / AES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *AES algorithm*] and specified cryptographic key sizes [assignment: *see FCS_COP.1*] that meet the following: [assignment: *Direct generation using FCS_RNG.1*].

Application Note 37

The TOE uses a cryptographic module certified in conformance with [EN 419 221-5], see also OE.CRYPTOMODULE_CERTIFIED 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.

Application Note 38

This ST uses cryptographic keys for different purposes. In this sense, an iteration has been included of this SFR for every key type it generates itself.

5.3.2.4 FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *active overwriting of the portion of memory containing the key*] that meets the following: [assignment: *none*].

Application Note 39

The TOE uses a cryptographic module certified in conformance with [CEN EN 419 221-5] for destruction of all keys.

Although the TSF does 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.

5.3.2.5 FCS_COP.1 / Digital signature generation and verification - Cryptographic operation

FCS_COP.1.1 / Digital signature generation and verification The TSF shall perform [assignment: *digital signature generation and verification*] in accordance with a specified cryptographic algorithm [assignment: *RSA PCKS#1 v.1.5; EC based DSA algorithm - Curve family FIPS Publication 186-4*] and cryptographic key sizes [assignment: *For RSA: 2048, 4096; For ECDSA: NIST P-256, NIST P-384, NIST P-521*] that meet the following: [*For RSA: PKCS#1 RSA Cryptography Standard, FIPS 186-4 Digital Signature Standard (DSS), IETF RFC 3447; For ECDSA: FIPS Publication 186-4*].

Application Note 40

The TOE uses a cryptographic module certified in conformance with [EN 419 221-5] for cryptographic operations.

Application Note 41

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 [eIDAS] and a list of approved signature and seal formats are given in [Formats].

5.3.2.6 FCS_COP.1 / Encryption and decryption - Cryptographic operation

FCS_COP.1.1 / Encryption and decryption The TSF shall perform [assignment: *encryption and decryption*] in accordance with a specified cryptographic algorithm [assignment: *RSA PCKS#1 v.1.5, OAEP; AES CBC*] and cryptographic key sizes [assignment: *For RSA: 2048 bits, 4096 bits; for AES: 128 bits*] that meet the following: [assignment: *For RSA: PKCS#1 RSA Cryptography Standard, FIPS 186-4 Digital Signature Standard (DSS), IETF RFC 3447; for AES: FIPS 197*].

Application Note 42

The TOE uses a cryptographic module certified in conformance with [EN 419 221-5] for cryptographic operations.

Application Note 43

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 [eIDAS] and a list of approved signature and seal formats are given in [Formats].

5.3.2.7 FCS_COP.1 / Message digest - Cryptographic operation

FCS_COP.1.1 / Message digest The TSF shall perform [assignment: *message digest*] in accordance with a specified cryptographic algorithm [assignment: *SHA256, SHA384, SHA512*] and cryptographic key sizes [assignment: *not applicable*] that meet the following: [assignment: *FIPS 180-4*].

Application Note 44

The TOE uses a cryptographic module certified in conformance with [EN 419 221-5] for cryptographic operations.

Application Note 45

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 [eIDAS] and a list of approved signature and seal formats are given in [Formats].

5.3.2.8 FCS_COP.1 / Message authentication - Cryptographic operation

FCS_COP.1.1 / Message authentication The TSF shall perform [assignment: *hash-based message authentication code*] in accordance with a specified cryptographic algorithm [assignment: *HMAC*] and cryptographic key sizes [assignment: *256 bits, 384 bits, 512 bits*] that meet the following: [assignment: *FIPS 198-1*].

Application Note 46

The TOE uses a cryptographic module certified in conformance with [EN 419 221-5] for cryptographic operations.

Application Note 47

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 [eIDAS] and a list of approved signature and seal formats are given in [Formats].

5.3.2.9 FCS_RNG.1 Generation of random numbers

FCS_RNG.1.1 The TSF shall provide a [selection: *physical*] random number generator that implements: [assignment: *AIS 31 class PTG.2 according to [AIS31]:*

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

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

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

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

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

FCS_RNG.1.2 The TSF shall provide [selection: *octets of bits*] that meet [assignment: *AIS 31 class PTG.2 according to [AIS31]: (PTG.2.6) Test procedure A and none does not distinguish the internal random numbers from output sequences of an ideal RNG; (PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997*].

Application Note 48

For more information on the selections and assignments, see the SFR definition in section FCS_RNG.1 Generation of random numbers*.*

Application Note 49

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 [EN 419-221-5] already provide requirements on generation of random numbers.

5.3.3 User Data Protection (FDP)

5.3.3.1 FDP_ACC.1/Privileged User Creation - Subset access control

FDP_ACC.1.1/ Privileged User Creation The TSF shall enforce the [assignment: *Privileged User Creation SFP*] on [assignment: *subjects: Privileged Users; objects: new security attributes* *for the Privileged User to be created; operations: Create_New_Privileged_User (the TOE creates R.Privileged_User and R.Reference_Privileged_User_Authentication_Data with information transmitted by Privileged User)*].

Application Note 50

The first Privilege User that must be created in the system is the FPU (First Privilege User). This user is unique in the TOE and he is responsible for the start-up of the component (its creation), as well as the administration of the initial static configuration. This role is able to create the Operation Privileged User and the Configuration Privileged User roles. The CPU (Configuration Privileged User) role is responsible for signing the SAM dynamic configurations, and it is the user that can create the SSPU (Signing Service Privileged User) roles.

For more information about the creation of Privileged Users, see section User Data Protection.

5.3.3.2 FDP_ACF.1/Privileged User Creation - Security attribute based access control

FDP_ACF.1.1/ Privileged User Creation

The TSF shall enforce the [assignment: *Privileged User Creation SFP*] to objects based on the following: [assignment: *whether the subject is a Privileged User authorized 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: [assignment: *only a Privileged User who has been authorised for creation of new users can carry out the Create_New_Privileged_User operation*].

FDP_ACF.1.3/ Privileged User Creation

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

FDP_ACF.1.4/ Privileged User Creation

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

5.3.3.3 FDP_ACC.1/Signer Creation - Subset access control

FDP_ACC.1.1/ Signer Creation

The TSF shall enforce the [assignment: *Signer Creation SFP*] on [assignment*: subjects: Privilege Users; objects: R.Signer and R.Reference_Signer_Authentication_Data; operations: Create_New_Signer (the TOE creates R.Signer and R.Reference_Signer_Authentication_Data with information transmitted by Privileged User)*].

5.3.3.4 FDP_ACF.1/Signer Creation - Security attribute based access control

FDP_ACF.1.1/ Signer Creation The TSF shall enforce the [assignment: *Signer Creation SFP*] to objects based on the following: [assignment*: whether the subject is a Privileged User authorized 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: [assignment: *only a Privileged User who has been authorised for creation of new users can carry out the Create_New_Signer operation*].

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

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

5.3.3.5 FDP_ACC.1/Signer Maintenance - Subset access control

FDP_ACC.1.1/ Signer Maintenance The TSF shall enforce the [assignment: *Signer Maintenance SFP*] on [assignment: *subjects: Privileged Users and Signer; objects: The security attributes R.Reference_Signer_Authentication_Data of R.Signer; operations: Signer_Maintenance (the Privileged User or Signer instructs the TOE to update R.Reference_Signer_Authentication_Data of R.Signer)*].

5.3.3.6 FDP_ACF.1/Signer Maintenance - Security attribute based access control

FDP_ACF.1.1/ Signer Maintenance The TSF shall enforce the [assignment: *Signer Maintenance SFP*] to objects based on the following: [assignment: *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: [assignment: *only a Privileged User or Signer who has been authorised to maintain a Signer can carry out the Signer_Maintenance operation*].

FDP_ACF.1.3/ Signer Maintenance The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *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: [assignment: *if the Signer does not own the R.Signer object, it can't be maintained*].

Application Note 51

For more information about the signer maintenance operation, see section User Data Protection*.*

5.3.3.7 FDP_ACC.1/Signer Key Pair Generation - Subset access control

FDP_ACC.1.1/ Signer Key Pair Generation The TSF shall enforce the [assignment: *Signer Key Pair Generation SFP*] on [assignment: *subjects: Privileged User and Signer; objects: the security attributes R.SVD and R.Signing_Key_Id as part of R.Signer; operations: Generate_Signer_Key_Pair (the Privileged User or Signer instructs the TOE to request the Cryptographic Module to generate a signing key pair R.Signing_Key_Id and R.SVD and assign them to the R.Signer)*].

Application Note 52

For more information about the signer maintenance operation, see section User Data Protection*.*

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

FDP_ACF.1.1/ Signer Key Pair Generation The TSF shall enforce the [assignment: *Signer Key Pair Generation SFP*] to objects based on the following: [assignment: *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: [assignment: *only a Privileged User or Signer who has been authorised to generate the key pair can carry out the Generate_Signer_Key_Pair 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: [assignment: *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: [assignment: *if the Signer does not own the R.Signer object, key pair shall not be generated; if the key pair is already assigned to a R.Signer object, key pair shall not be assigned*].

Application Note 53

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.

Application Note 54

Owning a R.Signer object is described in FIA_UAU.5/Signer.

5.3.3.9 FDP_ACC.1/Signer Key Pair Deletion - Subset access control

FDP_ACC.1.1/ Signer Key Pair Deletion The TSF shall enforce the [assignment: *Signer Key Pair Deletion SFP*] on [assignment: *subjects: Privileged User and Signer; objects: the security attributes R.Signing_Key_Id and R.SVD of R.Signer; operations: Signer_Key_Pair_Deletion (the Privileged User or Signer instructs the TOE to delete the R.Signing_Key_Id and R.SVD from R.Signer)*].

Application Note 55

This SFR is limited to covering deletion of the R.Signing_Key_Id and R.SVD of R.Signer performed using one of the interfaces provided by the TOE and where authorisation to perform operations is managed by TOE.

5.3.3.10 FDP_ACF.1/Signer Key Pair Deletion - Security attribute based access control

FDP_ACF.1.1/ Signer Key Pair Deletion The TSF shall enforce the [assignment: *Signer Key Pair Deletion SFP*] to objects based on the following: [assignment: *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:

[assignment: *only a Privileged User or Signer who has been authorised to delete a key pair can carry out the Signer_Key_Pair_Deletion 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: [assignment: *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: [assignment: *If the Signer does not own the R.Signer object, the key pair can't be deleted*].

5.3.3.11 FDP_ACC.1/Supply DTBS/R - Subset access control

FDP_ACC.1.1/ Supply DTBS/R The TSF shall enforce the [assignment: *Supply DTBS/R SFP*] on [assignment: *subjects: Privileged User; objects: the security attributes R.DTBS/R of R.Signer; operations: Supply_DTBS/R (the Privileged User instructs the TOE to link the supplied DTBS/R(s) to the next signature operation for R.Signer)*].

Application Note 56

Since the TOE does not provide facilities to supply the DTBS/R, then the relevant part of the SFR is trivially satisfied.

5.3.3.12 FDP_ACF.1/Supply DTBS/R - Security attribute based access control

FDP_ACF.1.1/ Supply DTBS/R The TSF shall enforce the [assignment: *Supply DTBS/R SFP*] to objects based on the following: [assignment: *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: [assignment: *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: [assignment: *none*].

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

Application Note 57

Since the TOE does not provide facilities to supply the DTBS/R, then the relevant part of the SFR is trivially satisfied.

5.3.3.13 FDP_ACC.1/Signing - Subset access control

FDP_ACC.1.1/ Signing The TSF shall enforce the [assignment: *Signing SFP*] on [assignment: *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: (1) the TOE establishes R.Authorisation_Data for the R.Signing_Key_Id, (2) the TOE uses the R.Authorisation_Data, and R.Signing_Key_Id to activate a signing key in the Cryptographic Module and signs the R.DTBS/R resulting in R.Signature, and (2) the TOE deactivates the signing key when the signature operation is completed)*].

Application Note 58

Section User Data Protection *includes a description of how R.Authorisation_Data is used to activate signing keys in the Cryptographic Module.*

Application Note 59

Section User Data Protection *includes a description of how the DTBS/R(s) is supplied to the TOE.*

Application Note 60

Signing key deactivating means that the signer shall authorise any subsequent use of it.

5.3.3.14 FDP_ACF.1/Signing - Security attribute based access control

FDP_ACF.1.1/ Signing The TSF shall enforce the [assignment: *Signing SFP*] to objects based on the following: [assignment: *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: [assignment: *(1) the R.SAD is verified in integrity; (2) the R.SAD is verified that it binds together the Signer authentication, a set of R.DTBS/R and R.Signing_Key_Id; (3) the R.DTBS/R used for signature operations is bound to the R.SAD; (4) the Signer identified in the SAD is authenticated according to the rules specified in FIA_UAU.5/Signer; (5) only an R.Signing_Key_Id as bound in the SAD, and which is part of the R.Signer security attributes, can be used to create a signature*].

FDP_ACF.1.3/ Signing The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *the Signer must be the owner of the R.Signer object used to generate the signature*].

FDP_ACF.1.4/ Signing The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: *if the Signer does not own the R.Signer object, it can't be used to create a signature*].

Application Note 61

The TOE does not work with default keys or single-use keys. A user can have N keys, and always has to authorize the use of each one.

5.3.3.15 FDP_ACC.1/TOE Maintenance - Subset access control

FDP_ACC.1.1/ TOE Maintenance The TSF shall enforce the TOE Maintenance SFP [assignment: *TOE Maintenance SFP*] on [assignment: *subjects: Privileged User; objects: R.TSF_DATA; operations: TOE Maintenance (the Privileged User transmits information to the TOE to manage R.TSF_DATA*)].

5.3.3.16 FDP_ACF.1/TOE Maintenance - Security attribute based access control

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

[assignment: *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: [assignment: *None*].

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

Application Note 62

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

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

FDP_ETC.2.1/ Signer The TSF shall enforce the [assignment: *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 TSF.

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

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

FDP_ETC.2.4/ Signer The TSF shall enforce the following rules when user data is exported from the TSF: [assignment: *none*]

Application Note 63

Section User Data Protection *includes the user data that can be exported from the TOE.*

5.3.3.18 FDP_IFC.1/Signer - Subset information flow control

FDP_IFC.1.1/ Signer The TSF shall enforce the [assignment: *Signer Flow SFP*] on [assignment: *Privileged User and Signer accessing Signer security attributes for all operations*].

5.3.3.19 FDP_IFF.1/Signer - Simple security attributes

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

FDP_IFF.1.2/ Signer The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: *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 The TSF shall enforce the [assignment: *none*].

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

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

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

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

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

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

FDP_ETC.2.4/ Privileged User The TSF shall enforce the following rules when user data is exported from the TSF: [assignment: *none*]

Application Note 64

Section User Data Protection *includes the user data that can be exported from the TOE.*

5.3.3.21 FDP_IFC.1/Privileged User - Subset information flow control

FDP_IFC.1.1/ Privileged User The TSF shall enforce the [assignment: *Privileged User Flow SFP*] on [assignment: *Privileged User accessing Privileged User security attributes for all operations*].

5.3.3.22 FDP_IFF.1/Privileged User - Simple security attributes

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

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

FDP_IFF.1.3/ Privileged User The TSF shall enforce the [assignment: *none*].

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

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

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

FDP_ITC.2.1/ Signer The TSF shall enforce the [assignment: *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: [assignment: *none*].

Application Note 65

Section User Data Protection *includes the user data that can be imported to the TOE.*

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

FDP_ITC.2.1/ Privileged User The TSF shall enforce the [assignment: *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: [assignment: *none*].

Application Note 66

Section User Data Protection *includes the user data that can be imported to the TOE.*

5.3.3.25 FDP_UCT.1 - Basic data exchange confidentiality

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

5.3.3.26 FDP_UIT.1 Data exchange integrity

FDP_UIT.1.1 The TSF shall enforce the [assignment: *Signer Flow SFP and Privileged User Flow SFP*] to [selection: *transmit, receive*] user data in a manner protected from [selection: *modification, insertion, replay*] errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether [selection: *modification, deletion, insertion*] has occurred.

Application Note 67

Insertion of objects would mean that authorised creation of Signer and Privileged User could be possible.

5.3.4 Identification and Authentication (FIA)

5.3.4.1 FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: *For Signers: R.Reference_Signer_Authentication_Data, R.Signing_Key_Id, R.SVD, R.Signer; For Privileged Users: R.Reference_Privileged_User_Authentication_Data.*].

5.3.4.2 FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow [assignment: *Establishment of a trusted path between the TOE and the SSA (through the SSPU) accomplishing the TSFs providing FTP_TRP.1; User identification accomplishing the TSFs providing FIA_UID.2*] 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.

5.3.4.3 FIA_UAU.5/Signer - Multiple authentication mechanisms

FIA_UAU.5.1/Signer The TSF shall provide [assignment: *Authorization assertion signed by a trusted Idp/AS component that authenticates the signer, and User secret*] to support signer authentication.

FIA_UAU.5.2/Signer The TSF shall authenticate any user's claimed identity according to the [assignment: *The signer authentication is conducted according to [EN 419 241-1] SCAL2 for qualified signatures. In this TOE, the signer authentication scheme can be (1) a delegated scheme (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; or (2) a mixed scheme (external factor plus an signer's secret). In case (1), the TOE verifies the assertion. In case (2), the TOE verifies the assertion and it also verifies the signer´s secret.*].

Application Note 69

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).

Application Note 70

Section Identification and Authentication *includes all the authentication factors type used to authenticate signer in accordance with [EN 419 241-1]. Successful authentication gives Signer access to the relevant R.Signer object as the owner.*

5.3.4.4 FIA_UAU.5/Privileged User - Multiple authentication mechanisms

FIA_UAU.5.1/Privileged User The TSF shall provide [assignment: *TLS client certificate, OCS cards*] to support Privileged User authentication.

FIA_UAU.5.2/Privileged User The TSF shall authenticate any user's claimed identity according to the [assignment: *TLS protocol authentication and verification that the public keys of the PUs are registered in the TOE configuration, an authentication by means of OCS cards is enforced*].

5.3.4.5 FIA_UID.2 User identification before any action

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.

5.3.4.6 FIA_USB.1 User-subject binding

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: *To Signer: R.Reference_Signer_Authentication_Data, R.Signing_Key_Id, R.SVD, R.Signer, R.Authorisation_Data, R.DTBS/R; To Privileged User: R.Reference_Privileged_User_Authentication_Data*].

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: [assignment: *(1) Whether the subject is a Privileged User authorized to create a new Signer; (2) Whether the subject is a Privileged User authorized to create a new Privileged User; (3) [assignment: 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: [assignment: *(1) Whether the subject is a Privileged User authorized to modify an R.Signer object; (2) Whether the subject is a Signer authorized to modify his own R.Signer object; (3) [assignment: None]*.].

Application Note 71

In FIA_USB.1.2 several attributes including R.Signing_Key_ID, R.SVD and R.DTBS/R may initially be empty.

Application Note 72

Section Identification and Authentication *includes all the Signer attribute.*

5.3.5 Security Management (FMT)

5.3.5.1 FMT_MSA.1/Signer - Management of security attributes

FMT_MSA.1.1/ Signer The TSF shall enforce the

(1) [assignment: *Signer Creation SFP*] to restrict the ability to [selection: [assignment: *create*]] the security attributes [assignment: *listed in FIA_USB.1 for Signer*] to [assignment: *authorised Privileged User*];

(2) [assignment: *Generate Signer Key Pair SFP*] to restrict the ability to [selection: [assignment: *generate*]] the security attributes [assignment: *R.SVD and R.Signing_Key_Id*] to [assignment: *authorised Privileged User and Signer*];

(3) [assignment: *Signer Key Pair Deletion SFP*] to restrict the ability to [selection: [assignment: *destruct*]] the security attributes [assignment: *R.SVD and R.Signing_Key_Id as part of R.Signer*] to [assignment*: authorised Signer*];

(4) [assignment: *Supply DTBS/R SFP*] to restrict the ability to [selection: [assignment: *create*]] the security attributes [assignment: *R.DTBS/R as part of R.Signer*] to [assignment: *authorised Privileged User*];

(5) [assignment: *Signing SFP*] to restrict the ability to [selection: [assignment: *create*]] the security attributes [assignment: *R.DTBS/R as part of R.Signer*] to [assignment: *authorised Signer*];

(6) [assignment: *Signing SFP*] to restrict the ability to [selection: *query*] the security attributes [assignment: *as listed in FIA_USB.1*] to [assignment: *authorised Signer*];

(7) [assignment: *Signer Maintenance SFP*] to restrict the ability to [selection: [assignment: *change*]] the security attributes [assignment: *R.Reference_Signer_Authentication_Data as part of R.Signer*] to [assignment: *authorised Privileged User and Signer*];

Application Note 73: Regarding point (1) of the requirement, the TOE does not store the related information, therefore, it does not force the associated restriction; guaranteeing in the same way the related security requirements.

5.3.5.2 FMT_MSA.1/Privileged User - Management of security attributes

FMT_MSA.1.1/ Privileged User The TSF shall enforce the [assignment: *Privileged User Creation SFP*] to restrict the ability to [selection: *query, [assignment: create]*] the security attributes [assignment: *listed in FIA_USB.1 for Privileged User]* to [assignment: *the authorised Privileged User*].

5.3.5.3 FMT_MSA.2 Secure security attributes

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

5.3.5.4 FMT_MSA.3/Signer - Static attribute initialisation

FMT_MSA.3.1/ Signer The TSF shall enforce the [assignment: *Signer Creation SFP*] to provide [selection, choose one of: *restrictive*] default values for security attributes that are used to enforce the SFP.

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

5.3.5.5 FMT_MSA.3/Privileged User - Static attribute initialisation

FMT_MSA.3.1/ Privileged User The TSF shall enforce the [assignment: *Privileged User Creation SFP*] to provide [selection, choose one of: *restrictive*] default values for security attributes that are used to enforce the SFP.

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

5.3.5.6 FMT_MTD.1 Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *modify*] the [assignment: *R.TSF_DATA*] to [assignment: *Privileged User*].

Application Note 74

The TSF data includes configuration of administrator roles.

5.3.5.7 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [assignment: *(1) Signer management, (2) Privileged User management, (3) Configuration management*].

5.3.5.8 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1 The TSF shall maintain the roles: [assignment: *First Privileged User (FPU), Operation Privileged User (OPU), Configuration Privileged User (CPU), Signing Service Privileged User and Signer*].

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 [assignment: *Signer can't be a Privileged User*] are satisfied.

Application Note 75

Section Security Management *describe which roles are defined in the TOE and which operations the role can perform.*

5.3.6 Protection of the TSF (FPT)

5.3.6.1 FPT_PHP.1 Passive

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

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

Application Note 76

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

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

5.3.6.2 FPT_PHP.3 Resistance

FPT_PHP.3.1 The TSF shall resist [assignment: *physical tampering through the chassis intrusion*] to the [assignment: *hardware on which the TOE is installed*] by responding automatically such that the SFRs are always enforced.

Application Note 77

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

5.3.6.3 FPT_RPL.1 Replay detection

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

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

5.3.6.4 FPT_STM.1 Reliable time stamps

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

Application Note 78

The TOE may receive a reliable time source from its environment.

5.3.6.5 FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret [assignment: *(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*] when shared between the TSF and another trusted IT product.

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

Application Note 79

The SFR is used to handle the situation where the whole or part of the above data are stored outside the TOE.

5.3.7 Trusted Paths/Channels (FTP)

5.3.7.1 FTP_TRP.1/SSA Inter-TSF Trusted path

FTP_TRP.1.1/SSA The TSF shall provide a communication path between itself and [selection: *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 [selection: *modification*].

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

FTP_TRP.1.3/SSA The TSF shall require the use of the trusted path for [selection: *(1) FDP_ACC.1.1/Privileged User Creation; (2) FDP_ACC.1/Signer Creation; (3) FDP_ACC.1/Signer Maintenance; (4) FDP_ACC.1/Signer Key Pair Generation; (5) FDP_ACC.1/Signer Key Pair Deletion; (6) FDP_ACC.1/Supply DTBS/R; (7) FDP_ACC.1/TOE Maintenance*].

Application Note 80

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.

5.3.7.2 FTP_TRP.1/SIC Inter-TSF Trusted path

FTP_TRP.1.1/SIC The TSF shall provide a communication path between itself and [selection: *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 [selection: *modification*].

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

FTP_TRP.1.3/SIC The TSF shall require the use of the trusted path for [selection: *(1) FDP_ACC.1/Signer Maintenance; (2) FDP_ACC.1/Signer Key Pair Generation; (3) FDP_ACC.1/Signer Key Pair Deletion; (4) FDP_ACC.1/Signing*].

Application Note 81

The R.SAD generated by the signer during signature activation protocol (SAP) execution is protected in integrity by a digital signature and, when it contains R.Authorisation_Data from the signer, it is encrypted (using the SAM key) for confidentiality.

The TOE is not expected to verify the SIC as a communication end point and it may rely on the signer authentication.

5.3.7.3 FTP_ITC.1/CM Inter-TSF trusted channel

FTP_ITC.1.1/CM The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/CM The TSF shall permit [selection: *the TSF and a cryptographic module certified according to [CEN 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 [assignment: *request for key pair generation, request for certificate insertion, request for signature generation, request for key pair deletion*].

Application Note 82

In this Security Target, the FTP_ITC.1.1/CM requirement maps the "trusted IT product" to the "cryptographic module certified according to [CEN EN 419 221-5]".

5.4 Extended Components Definitions

5.4.1 Class FCS: Cryptographic Support

The Class FCS: Cryptographic Support as defined in [CC2] is extended with a new family: Generation of Random Numbers (FCS_RNG). The family is concerned with generation of random numbers. The following picture illustrates the decomposition of the Class FCS: Cryptographic Support with the added family FCS_RNG.

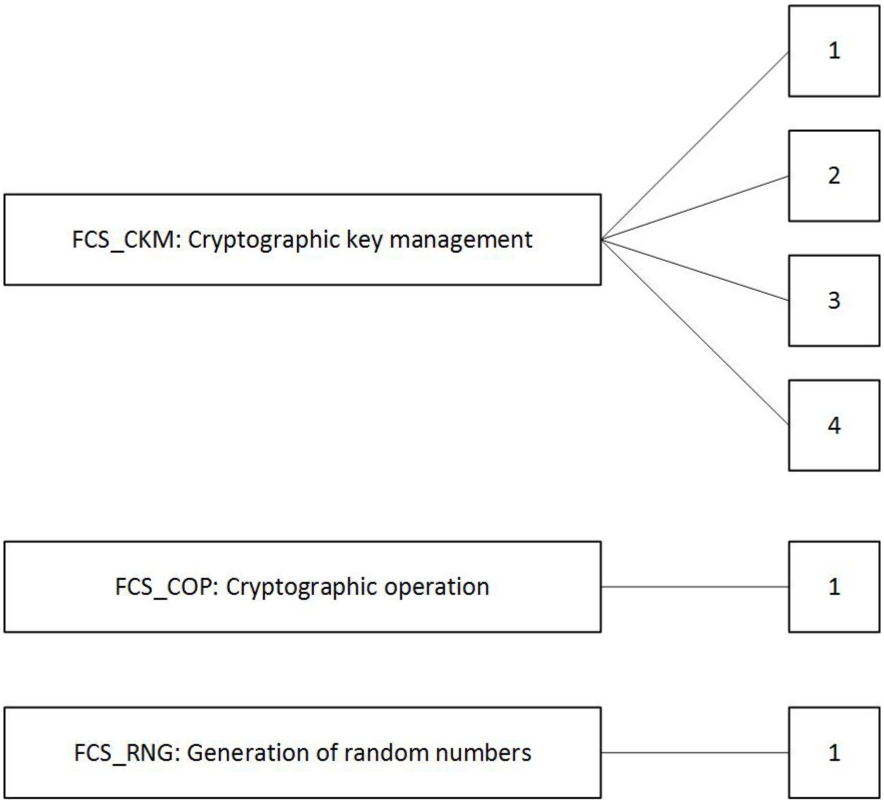

FCS_CKM: Cryptographic key management
1
2
3
4
FCS_COP: Cryptographic operation
1
FCS_RNG: Generation of random numbers
1

Figure 1. Class FCS: Cryptographic Support

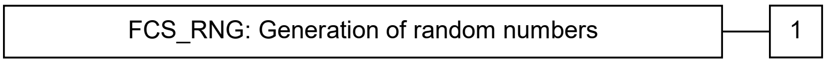
5.4.1.1 FCS_RNG.1 Generation of random numbers

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

Family behaviour

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

Component levelling



Management: FCS_RNG.1

There are no foreseen management activities.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].

FCS_RNG.1.2 The TSF shall provide [selection: bits, octets of bits, numbers [assignment: *format of the numbers*]] that meet [assignment: *a defined quality metric*].

Application Note 83

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.

5.5 Security Assurance Requirements

The security assurance requirement level is EAL4 augmented with ALC_FLR.2 and AVA_VAN.5. The assurance components are identified in the table below with the augmented item in bold.

Since the TOE is operated in a physically protected environment as described in OE.ENV an evaluation against this PP will probably not include physical attacks.

Assurance Class	**Assurance Components**
Development (ADV)	Security architecture description (ADV_ARC.1)
	Complete functional specification (ADV_FSP.4)
	Implementation representation of the TSF (ADV_IMP.1)
	Basic modular design (ADV_TDS.3)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life-cycle support (ALC)	Production support, acceptance procedures and automation (ALC_CMC.4)
	Problem tracking CM coverage (ALC_CMS.4)
	Delivery procedures (ALC_DEL.1)
	Identification of security measures (ALC_DVS.1)
	Developer defined life-cycle model (ALC_LCD.1)

	Well-defined development tools (ALC_TAT.1)
	Flaw reporting procedures (ALC_FLR.2)
Security Target evaluation (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives (ASE_OBJ.2)
	Derived security requirements (ASE_REQ.2)
	Security problem definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Tests (ATE)	Analysis of coverage (ATE_COV.2)
	Testing: basic design (ATE_DPT.1)
	Functional testing (ATE_FUN.1)
	Independent testing - sample (ATE_IND.2)
Vulnerability assessment (AVA)	**Advanced methodical vulnerability analysis (AVA_VAN.5)**

Table 2. Security Assurance Requirements

5.6 Security Rationale

5.6.1 Security Requirements Rationale

5.6.1.1 Security Requirements Coverage

The following table is used to demonstrate that every SFR is used to cover an objective and that every objective is covered by an SFR. The table is not complete in the sense that all possible crosses are created.

	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
Security Audit									
FAU_GEN.1									
FAU_GEN.2									
Cryptographic Support									
FCS_CKM.1			X						
FCS_CKM.4			X						
FCS_COP.1			X						
FCS_RNG.1			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									
FDP_ACF.1/ Supply DTBS/R									
FDP_ACC.1/Signing									
FDP_ACF.1/Signing									
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_ATD.1	X				X		X		
FIA_UAU.1						X			

FIA_UAU.5/Signer									
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									
FPT_STM.1									
FPT_TDC.1	X				X				
Trusted Path/Channels									
FTP_TRP.1/SSA									X
FTP_TRP.1/SIC									
FTP_ITC.1/CM			X						

Table 3. Security Requirements Rationale I

	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	
FCS_CKM.4								
FCS_COP.1						X	X	
FCS_RNG.1								X
User Data Protection								
FDP_ACC.1/Privileged User Creation								
FDP_ACF.1/Privileged User Creation								
FDP_ACC.1/Signer Creation								
FDP_ACF.1/Signer Creation								
FDP_ACC.1/ Signer Maintenance								
FDP_ACF.1/ Signer Maintenance								
FDP_ACC.1/Signer Key Pair Generation								

FDP_ACF.1/Signer Key Pair Generation								
FDP_ACC.1/Signer Key Pair Deletion								
FDP_ACF.1/Signer Key Pair Deletion								
FDP_ACC.1/ 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								
FDP_ACF.1/TOE Maintenance								
FDP_ETC.2/Signer								
FDP_IFC.1/Signer								
FDP_IFF.1/Signer								
FDP_ETC.2/Privileged User								
FDP_IFC.1/Privileged User								
FDP_IFF.1/privileged User								
FDP_ITC.2/Signer								
FDP_ITC.2/Privileged User								
FDP_UCT.1								
FDP_UIT.1								
Identification and authentication								
FIA_ATD.1								
FIA_UAU.1		X						
FIA_UAU.5/Signer		X						
FIA_UAU.5/Privileged User								
FIA_UID.2								

FIA_USB.1								
Security Management								
FMT_MSA.1/Signer								
FMT_MSA.1/Privileged User								
FMT_MSA.2								
FMT_MSA.3/Signer								
FMT_MSA.3/Privileged User								
FMT_MTD.1								
FMT_SMF.1								
FMT_SMR.2								
Protection of the TSF								
FPT_PHP.1								
FPT_PHP.3								
FPT_RPL.1			X					
FPT_STM.1	X							
FPT_TDC.1								
Trusted Path/Channels								
FTP_TRP.1/SSA					X			
FTP_TRP.1/SIC			X	X	X			
FTP_ITC.1/CM						X		

Table 4. Security Requirements Rationale II

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

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

OT.SIGNER_KEY_PAIR_GENERATION is handled by the requirements for key generation and cryptographic algorithms in FCS_CKM.1 and FCS_COP.1. FCS_RNG.1 provides a random source for key generation. FCS_CKM.4 describes the requirements for key destruction. FDP_ACC.1/Signer Key Pair Generation and FDP_ACF.1/Signer Key Pair Generation describes access control for creating a key pair. FIA_USB.1 describes that R.Signing_Key_Id is associated with Signer. FTP_ITC.1/CM can be used to communicate securely with a Cryptographic Module.

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

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

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

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

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

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

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

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

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

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

OT.DTBSR_INTEGRITY is covered by FTP_TRP.1/SSA and FTP_TRP.1/SIC requiring data transmission to be protected in integrity.

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

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

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

5.6.2 SFR Dependencies

The dependencies between SFRs are addressed as shown in.

Requirement	Dependencies	Fullfilled by
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1 FIA_UID.2
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1 and FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1
FCS_RNG.1	None	No dependents
FDP_ACC.1/Privileged User Creation	FDP_ACF.1	FDP_ACF.1/Privileged User Creation
FDP_ACC.1/Signer Creation	FDP_ACF.1	FDP_ACF.1/Signer Creation
FDP_ACC.1/Signer Maintenance	FDP_ACF.1	FDP_ACF.1/Signer Maintenance
FDP_ACC.1/Signer Key Pair Generation	FDP_ACF.1	FDP_ACF.1/Signer Key Pair Generation
FDP_ACC.1/Signer Key Pair Deletion	FDP_ACF.1	FDP_ACF.1/Signer Key Pair Deletion
FDP_ACC.1/Supply DTBS/R	FDP_ACF.1	FDP_ACF.1/Supply DTBS/R
FDP_ACC.1/Signing	FDP_ACF.1	FDP_ACF.1/Signing
FDP_ACC.1/TOE Maintenance	FDP_ACF.1	FDP_ACF.1/TOE Maintenance

FDP_ACF.1/Privileged User Creation	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Privileged User Creation FMT_MSA.3/Privileged User
FDP_ACF.1/Signer Creation	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signer Creation FMT_MSA.3/Signer
FDP_ACF.1/Signer Maintenance	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signer Maintenance FMT_MSA.3/Signer
FDP_ACF.1/Signer Key Pair Generation	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signer Key Pair Generation FMT_MSA.3/Signer
FDP_ACF.1/Signer Key Pair Deletion	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signer Key Pair Deletion FMT_MSA.3/Signer
FDP_ACF.1/Supply DTBS/R	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Supply DTBS/R FMT_MSA.3/Signer
FDP_ACF.1/Signing	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signing FMT_MSA.3/Signer
FDP_ACF.1/TOE Maintenance	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/TOE Maintenance FMT_MSA.3/Privileged User
FDP_ETC.2/Signer	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1/Signer
FDP_ETC.2/Privileged User	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1/Privileged User
FDP_IFC.1/Signer	FDP_IFF.1	FDP_IFF.1/Signer
FDP_IFF.1/Signer	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/Signer FMT_MSA.3/Signer
FDP_IFC.1/Privileged User	FDP_IFF.1	FDP_IFF.1/Privileged User
FDP_IFF.1/Privileged User	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/Privileged User

		FMT_MSA.3/Privileged User
FDP_ITC.2/Signer	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1] FTP_TDC.1	FDP_IFC.1/Signer FTP_TRP.1/SSA and FTP_TRP.1/SIC FPT_TDC.1
FDP_ITC.2/Privileged User	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1] FTP_TDC.1	FDP_IFC.1/Privileged User FTP_TRP.1/SSA FPT_TDC.1
FDP_UCT.1	[FTP_ITC.1 or FTP_TRP.1] [FDP_ACC.1 or FDP_IFC.1]	FTP_TRP.1/SSA and FTP_TRP.1/SIC FDP_IFC.1/Signer FDP_IFC.1/Privileged User
FDP_UIT.1	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	FDP_IFC.1/Signer FDP_IFC.1/Privileged User FTP_TRP.1/SSA and FTP_TRP.1/SIC
FIA_ATD.1	None	
FIA_UAU.1	FIA_UID.1	FIA_UID.2
FIA_UAU.5/Signer	None	
FIA_UAU.5/Privileged User	None	
FIA_UID.2	None	
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MSA.1/Signer	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_IFC.1/Signer FMT_SMR.2 FMT_SMF.1
FMT_MSA.1/Privileged User	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_IFC.1/Privileged User FMT_SMR.2 FMT_SMF.1
FMT_MSA.2	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	FDP_IFC.1/Signer FDP_IFC.1/Privileged User FMT_MSA.1/Signer

		FMT_MSA.1/Privileged User FMT_SMR.2
FMT_MSA.3/Signer	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/Signer FMT_SMR.2
FMT_MSA.3/Privileged User	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/Privileged FMT_SMR.2
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	FMT_SMR.2 FMT_SMF.1
FMT_SMF.1	None	
FMT_SMR.2	FIA_UID.1	FIA_UID.2
FPT_PHP.1	None	
FPT_PHP.3	None	
FPT_RPL.1	None	
FPT_STM.1	None	
FPT_TDC.1	None	
FTP_TRP.1/SSA	None	
FTP_TRP.1/SIC	None	
FTP_ITC.1/CM	None	

Table 5. SFR Dependencies

5.6.2.1 Rationales for SARs

EAL4 permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, through rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest level at which it is likely to be economically feasible to retrofit to an existing product line.

As the TOE manages signature creation data generation and authorises it's use it manage security attributes which can only be ensured by the TOE. While the TOE is assumed to be in a physically protected environment, it is still subject to logical remote attacks and should be evaluated to deal with High attack potential.

EAL4 is therefore augmented with AVA_VAN.5.

6 TOE Summary specification

6.1 Security Audit

FAU_GEN.1 Audit Generation, FAU_GEN.2 User identity association

The TOE generates audit records for all the security events involved in the TSFs, and associates each record with the identity of the user that caused the related event. The TOE issues and signs an event log for any given service request, and then it stores them in a permanent repository. Audit records can be verified and securely stored externally by the SSA. The TOE signs the audit records using a keyed-hash message authentication code.

The audit records include but are not limited to, the date and time of the event, the type of event, the subject identity who performs the operation, the success or failure of the event, and a mechanism that guarantees the integrity of the configuration.

6.2 Cryptographic Support

FCS_CKM.1 Cryptographic key generation, FCS_CKM.4 Cryptographic key destruction, FCS_COP.1 Cryptographic operation, FCS_RNG.1 Generation of random numbers

The TOE functional requirements that implement the FCS requirements included in this Security Target, use a HSM certified according to [CEN EN 419 221-5].

6.3 User Data Protection

FDP_ACC.1/Privileged User Creation Subset access control, FDP_ACF.1/Privileged User Creation Security attribute based access control

The First Privilege User that must be created in the system is the FPU (First Privileged User). This role is unique in the TOE and is responsible for the start-up of the component (creation of the private configuration), as well as the administration of the initial public configuration composed by the static [3] and dynamic [4] configuration. The First Privilege User creates the Operation Privileged User (OPU), the Configuration Privileged User (CPU) and the Signing Service Privileged Users (SSPU) roles. The CPU role is also responsible for defining dynamic configurations. The OPU role can only load dynamic configurations signed by the CPU. The OPU role is the only one who can run the TOE.

The first step for the creation of Privileged Users is to create the Security World of the HSM and define the ACSs (Administrator Card Set) and the necessary OCSs (Operator Card Set). Both the FPU role and the OPU role require OCS cards in order to access the TOE. When the

[3] The static configuration defines the behavior of the SAM including the list of Configuration Privileged User and Operation Privileged User roles. To load this configuration it is necessary to restart the SAM.

[4] The dynamic configuration defines the list of Signing Service Privileged User and Authorization Server roles. This resources can be loaded dynamically, it is not necessary to stop the SAM.

TOE is created an event is registered in the logs repository. The FPU (N/M) and the OPU (N/M) protect the TOE infrastructure keys.

The SSPU role recognized by the TOE is included in the dynamic configuration of the TOE. The FPU can directly import dynamic configuration and an event is registered in the logs repository. The CPU role can sign a dynamic configuration and the OPU role makes the signed dynamic configuration effective by loading it. The OPU role cannot directly create Privileged Users.

All SAM components require specific Privileged User authentication before executing its security functions.

FDP_ACC.1/Signer Creation Subset access control, FDP_ACF.1/Signer Creation Security attribute based access control

The TOE functionality related to the signer creation requirement creates the signer and associates its authentication data. That is, the TOE creates users in the same operation that assigns them a key (first the keys are created and later they are assigned to a Signer that is managed by a SSPU). The SSPU privileged user is the only one who can create the asset R.Signer; in this operation it associates this asset with the key pair and also with the SSPUs that can manage these keys. A key pair that has already been assigned cannot be assigned to another user.

The FDP_ACC.1 / Signer Creation component is responsible for creating the R.Reference_Signer_Authentication_Data attribute, which includes the public keys of the Authorization Server component, and optionally the secret key of the signer. The creation of a new signer does not entail the creation of new values for the keys of an Authorization Server component, but it does create a security attribute R.Reference_Signer_Authentication_Data for that specific signer, containing the value of the keys of the AS that it has. assigned that particular signer.

The R.Reference_Signer_Authentication_Data attribute is represented in the TOE as part of the Signer User Information resource described in the ENTRUST_SAM document and corresponding to R: Signer, since it contains the signing_service_privileged_user_id property that indicates the Signing Service Privileged User who created the corresponding resource to the signer and that he will be able to operate with him. In turn, this user must appear as accepted in the dynamic configuration of the TOE. In this configuration, also described in the ENTRUST_SAM document, the Signing Service Privileged User is specified in the signing_service_privileged_users property, within which the list of authorization servers is registered with their public keys in the authorization_servers property, and those accepted by that user (property allowed_authorization_servers).

Optionally, the user can authorize the assignment of keys through the presence of a signed assertion. This optionality is due to the fact that a Privileged User must be able to perform this assignment administratively (for example, in the functionality associated with these signer creation requirements), but also a signer must be able to perform the functionality of key assignment to this signer by means of an authorization from him (signed assertion).

The signer creation operation is always authenticated through the TLS protocol (with client authentication) and also in this operation it is verified that the SSPU Privileged User that invokes it is an accepted Signing Service Privileged User.

FDP_ACC.1/Signer Maintenance Subset access control, FDP_ACF.1/Signer Maintenance Security attribute based access control

In this implementation, the First Privileged User, the Configuration Privileged User (through the OPU) and the Signer (through de SSPU) can invoke operations in the TOE that allow modifying the data used by the TOE to authenticate the signer, and which are contained in the asset R.Reference_Signer_Authentication_Data of the R.Signer.

Specifically, these operations are the following:

<u>Update of the secret that the signer can use in a mixed authentication scheme</u>. This operation requires the explicit consent of the signer by means of an assertion signed by an authorized IdP/Authorization Server.

The operation requires authentication and external authorization: (1) this is authenticated from the TLS protocol with client authentication, and it is invoked through a previously accepted Signing Service Privileged User (SSPU); and (2) it is only possible to perform this operation based on the explicit consent of the signer.

Only the owner of the asset R.Signer can authorize a maintenance operation of its asset R.Reference_Signer_Authentication_Data.

<u>Update of the public keys necessary to verify the assertions</u>. This operation can be performed directly by the FPU. Furthermore, this operation can be performed by the CPU, since the updated information is stored in the dynamic configuration signed by the Configuration Privileged User (CPU) and which is loaded dynamically (with the server on and processing requests) in the TOE by the OPU. This operation does not require the explicit authorization of the signer.

The operation is authenticated through the TLS protocol with client authentication and the Privileged User must be an Operation Privileged User accepted by the TOE.

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

Both the SSPU Privileged User and the Signer can request the generation of the signing key pair of a Signer. When a Signer requests this operation, the invocation is carried out by the SSPU, but the explicit authorization of the Signer is required (signed assertion) without which the key generation operation cannot be carried out.

Only the Signing Service Privileged User role accepted in the system, and authenticated by the TLS protocol with client authentication, can generate a key pair of a Signer.

The key pair generation operation for a Signer has the following two parts:

- Generation of a key pair. The key pair will be generated according to the algorithm and size parameters of the request, and according the parameters supported by the TOE. For this part, the explicit authorization of the signer is not necessary, allowing cases of use such as pre-generation of keys. At the end of this key generation process, the private key will not be assigned to any user, nor will it be able to sign since it has not yet been certified.
- Assignment of the keys created to a specific Signer. This functionality associates a pre-generated key pair with a specific Signer. Optionally, the Signer must authorize the assignment through the presence of a signed assertion. This optionality is due to the fact

that the Privileged User can administratively assign keys to Signers, or the request can come from a Signer who requests this assignment making use of an explicit authorization.

A key pair assigned to a Signer cannot be assigned to another user, thus respecting the uniqueness of the Signers' keys.

This functionality sets the R.Authorisation_Data asset, since the user is allowed to explicitly authorize the assignment of the keys and activate the secret, by means the SKAD assertion (this would be optional, since the SSPU could assign keys directly).

The TOE that identifies and describes this Security Target can interact with several Cryptographic Modules at the same time if all these Modules share the same Security World. The keys generated by the TOE and stored outside the TOE can be used in multiple TOEs, as long as 1) the TOE instance is the same and 2) it is the same security world of the Cryptographic Module.

In the process of generating a key pair of a Signer, the TOE generates an internal Key_Pair structure where control information is stored together with the key pair (for example, the status of the key, information on the type of the key, user password activation secret if used, ...). This entire structure is signed by the TOE infrastructure key, so integrity is guaranteed by the TOE. The TOE protection system guarantees that no role of the system by itself can make an attack that compromises the user's Sole Control to their keys. The main security feature is that the TOE infrastructure keys are protected by the Operation Privileged User's OCS, and stored in the SAM's Private Store that the Operation Privileged User by itself cannot access.

When the key pair is assigned to a Signer, the R.Signer asset is created or modified. The R.Signer structure is signed by the TOE infrastructure key, so the TOE itself maintains the link between the R.Signer and the Key_Pair structure. The TOE controls that a key pair can only be assigned to one user and it is the only one that can change the status of the assigned keys.

Regarding the R.Authorisation_Data, the TOE internal structure related to this asset, it contains an identifier of the signature key that is tried to be activated and also user information (which must coincide with that included in the R.Signer asset).

The Key_Pair structure contains information about the status of the key. If the status is not "assigned", then the key pair is a pre-generated key pair that is not yet assigned; only the TOE can modify the status information because the Key_Pair structure is signed by the TOE infrastructure key, and only the TOE can use this key.

FDP_ACC.1/Signer Key Pair Deletion Subset access control, FDP_ACF.1/Signer Key Pair Deletion Security attribute based access control

The functionality associated with these requirements removes a key pair of a signer from the TOE. Both the Signing Service Privileged User (SSPU) and the Signer can request the deletion of the signing key pair of a Signer. When a Signer requests this operation, the invocation is carried out by the SSPU, but the explicit authorization of the Signer is required (signed assertion) without which the key deletion operation cannot be carried out. The operation receives the R.Signer asset and the identifier of the key pair to eliminate, and disassociate the R.Signer asset from the key pair of the Signer (private and public key). This functionality re-signs the R.Signer asset to ensure that the keys can no longer be used. In this operation, the HSM does not eliminate the keys since they are not found within the HSM (they are externally persisted and protected by the SAM). Only the signer who owns their keys can delete their own keys.

Only the Signing Service Privileged User role accepted in the system, and authenticated by the TLS protocol with client authentication, can delete a key pair of a Signer.

FDP_ACC.1/Supply DTBS/R Subset access control, FDP_ACF.1/Supply DTBS/R Security attribute based access control, FDP_ACC.1/Signing Subset access control, FDP_ACF.1/Signing Security attribute based access control

The functionality associated with these requirements generates a qualified electronic signature for a specific Signer. The Signer that is the owner of the related signature key is the only one who can request the generation of this qualified signature. When a Signer requests this operation, the invocation is carried out by the Signing Service Privileged User (SSPU), but the explicit authorization of the Signer is required (signed assertion) without which the key signing operation cannot be carried out.

Only the Signing Service Privileged User role accepted in the system, and authenticated by the TLS protocol with client authentication, can generate a signing request for a Signer. In addition, the SSPU involved in the signature operation must have been the same as the one involved in the key generation operation.

All the signatures generated by the TOE must be authorised though the R.Authorisation_Data, that is used in order to activate the signing key in the Cryptographic Module. This asset consists basically of the signed assertion (SAD) and all the necessary data to ensure that the assertion comes from an authorized Authorization Server (Authorization Server public keys). The TOE verifies the SAD assertion before the R.Authorisation_Data is used to activate the signing key in the Cryptographic Module.

The SAD can include a user secret (user credential) that activate the user´s signing key. This part of the SAD is protected in confidentiality by an infrastructure asymmetric key specific for the encryption of SAD's confidential data. When the SAM verifies a signer's authentication factor (mixed scheme), it verifies a secret that the Signer's only knows, and that has been included in the R.SAD asset.

The TOE ensures that 1) the Signer is authorised to sign this specific DTBS/R using an specific key; 2) the key is authorised for signing this specific DTBS/R by this specific Signer; and 3) this DTBS/R is authorised to be signed by this specific Signer using an specific signing key. All these guarantees are ensured because the SAD includes the user identifier, the specific signing key identifier to be used, and the DTBS/R data. In addition, before generating the signature, the TOE verifies that the specific key included in the SAD is a signing key assigned previously to this user.

Before the signing process, the integrity of the SAD structure is verified, as well as the integrity of the R.Signer and R.Key_pair structures. The SAD is signed by an Authorisation Server (AS) that must be previously authorised by the TOE (it must be included in the TOE dynamic configuration). The ASs that can sign SADs are explicitly configured for each SSPU; if an AS is registered, but it is not associated with the SSPU involved in the signing operation, the signed SAD will not be accepted. Before the signing process, the TOE also guarantees that the signing key to be used is enabled and its associated public key has been certified.

When the signing process is completed, in order to carry out another signature, the key is no longer loaded in the CM, but it can only be activated by repeating a new invocation of the signing process, which will require a new R.Authorisation_Data asset, and therefore a new authorization of the signer.

The R.Signing_key_id asset is always mandatory in this TOE, both in the SAD and in the Signing request for verification. The TOE does not work with default keys or single-use keys. A user can have N keys, and always has to authorize the use of each one.

The TOE also supports batch signature. In this case, the signature operation includes several DTBS/R, and the SAD includes the authorization for the complete set of data to be signed.

FDP_ACC.1/TOE Maintenance Subset access control, FDP_ACF.1/TOE Maintenance Security attribute based access control

The TOE allows maintenance operations of the configuration data to the different Privileged Users. Basically, the TOE configuration is composed by public configuration ([1] static configuration and [2] dynamic configuration) and private configuration ([3] TOE infrastructure keys protected by the CM). Operative private and public configuration are stored within the private store. Private configuration is protected in confidentiality when exported.

The FPU can directly import static configuration and an event is registered in the logs repository.

The FPU can directly import dynamic configuration and an event is registered in the logs repository. Additionaly, the CPU role can sign a dynamic configuration and the OPU role makes the signed dynamic configuration effective by loading it. The OPU role is authenticated using a TLS protocol. When OPU synchronizes the signed dynamic configuration it will be verified that the configuration version is higher than the current dynamic configuration version in order to avoid malicious downgrades.

Types of configuration structures

[1] The static configuration defines the behaviour of the SAM including the list of Configuration Privileged User and Operation Privileged User roles. To load this configuration it is necessary to stop the SAM.

[2] The dynamic configuration defines the list of Signing Service Privileged User and Authorization Server roles. This resources can be loaded dynamically, it is not necessary to stop the SAM.

[3] The Infrastructure Keys include the TOE and the FPU infrastructure configuration. The TOE's infrastructure configuration includes the keys used to protect user keys, as well as other keys used for logging, signing of material protected by the TOE, etc. These keys are protected by the OCS of the OPU and therefore cannot be accessed by anyone. The FPU infrastructure configuration includes the keys used by the FPU for authentication when executing administration commands, as well as the proprietary log signature key. These keys are protected by the OCS of the FPU and therefore cannot be accessed by anyone.

FDP_ETC.2/Signer Export of user data with security attributes, FDP_IFC.1/Signer Subset information flow control, FDP_IFF.1/Signer Simple security attributes, FDP_ETC.2/ Privileged User Export of user data with security attributes,

FDP_IFC.1/Privileged User Subset information flow control, FDP_IFF.1/Privileged User Simple security attributes

The SFPs identified in the assignments of the requirements affect the following user data:

- R.Reference_Signer_Authentication_Data: optionally can include the secret of the user (as an authentication factor, if this is used).
- R.Key_Pair: The key pair structure is signed by the SAM. It includes the private key protected in confidentiality by de CM. It contains the SSPU unique id that can manage it.
- Signatures: It is an array of signatures. The TOE can only generate signatures through the signature generation operation of the SAP protocol. This operation requires the presence of the SAD that authorizes the activation of the key identified in R.Key_Pair asset and, in addition, must belong to the user identified in the R.Signer asset. Also, only the SSPU that generated the R.Key_Pair can activate that key. If all of these conditions are met, then the TOE generates the signature and returns it to the SSPU authenticated in the request.
- R. Signer: This structure is signed by the SAM. It contains the SSPU unique id that can manage it.
- R. Audit: Audit events may include user data, but never confidential data. All events are signed by the TOE.
- TOE configurations: signed structure that only can be accessed by specific Privileged Users.

All signed structures ensure that security attributes are unambiguously associated with exported user data. All this data are accessible only through TOE operations, and to access these operations any PU needs TLS access with a correct certificate and recognized by the TOE. Additionally, the privileged user FPU can also manage the TOE configuration, previously authenticating with their OCS.

Only the FPU Privileged User, authenticating through their OCS, can manage the static configuration data. And only the FPU Privileged User, authenticating with their OCS, or the OPU Privileged User, after signing the configuration by a CPU, can manage the dynamic configuration.

The public keys of the CPU, SSPU and OPU Privileged Users are all contained in the TOE configurations, these are always uniquely associated with their identifiers, and this structure is signed. Additionally, the infrastructure key backups are encrypted by the OPU key, protected by its OCS, and also by the FPU key, protected by its OCS; consequently, this backup can only be restored in the event that these two roles are present with their OCSs.

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

The SFPs identified in the assignments of the requirements affect the following user data:

- R.SAD: This asset is protected in integrity by a digital signature and, when it contains a signer´s secret, it is encrypted (using the public key of the TOE) for confidentiality.
- R. Signer: This structure is signed by the TOE. It contains the SSPU unique id that can manage it.
- R.Key_Pair: The key pair structure is signed by the SAM. It includes the private key protected in confidentiality by de CM. It contains the SSPU unique id that can manage it.
- DTBS/R: these data are included in the signed R.SAD asset.

- R. Audit: Audit events may include user data, but never confidential data. All events are signed by the TOE.
- TOE configurations: signed structure that only can be accessed by specific Privileged Users.

All signed structures ensure that security attributes are unambiguously associated with exported user data. All this data are accessible only through TOE operations, and to access these operations any PU needs TLS access with a correct certificate and recognized by the TOE. Additionally, the privileged user FPU can also manage the TOE configuration, previously authenticating with their OCS.

Only the FPU Privileged User, authenticating through their OCS, can manage the static configuration data. And only the FPU Privileged User, authenticating with their OCS, or the OPU Privileged User, after signing the configuration by a CPU, can manage the dynamic configuration.

The public keys of the CPU, SSPU and OPU Privileged Users are all contained in the TOE public configuration, these are always uniquely associated with their identifiers, and this structure is signed. Additionally, the private configuration backups are encrypted by the OPU key, protected by its OCS, and also by the FPU key, protected by its OCS; consequently, this backup can only be restored in the event that these two roles are present with their OCSs.

FDP_UCT.1 Basic data exchange confidentiality, FDP_UIT.1 Data exchange integrity

The TOE provides functionality for providing confidentiality and integrity for user data in transit between the TOE and the SSA/HSM.

The TOE uses the TLS protocol version 1.2, over a mutually authentication channel in order to communicate with the SSA. The communication between the TOE and the HSM is protected by a proprietary secure channel of the HSM. This channel provides data confidentiality and integrity.

Regarding the communication between the TOE and the SSA component, in order for the TOE to generate a signature, it needs a SAD, which includes the DTBS/R structure (cannot be reused to sign another hash), the identifier of the signer's key to be used in the signature (it cannot be used to activate a different key), and it also includes a time stamp (this same SAD cannot be reused after a while).

6.4 Identification and Authentication

FIA_ATD.1 User attribute definition, FIA_UAU.1 Timing of authentication, FIA_UAU.5/Signer Multiple authentication mechanisms, FIA_UAU.5/Privileged User Multiple authentication mechanisms, FIA_UID.2 User identification before any action, FIA_USB.1 User-subject binding

The TOE creates and maintains the association of user security attributes with Signers and Privileged Users, in a secure way by maintaining them in signed structures where this association is guaranteed. The TOE validates the attributes before assigning them to the users.

Regarding the FIA_USB.1.1 requirement, the following user attributes are linked with the users who are acting on their behalf (performing the operation): 1) R.Reference_Signer_Authentication_Data: User Identifier (SAD); 2) R.Signing_Key_Id: The key identifier will be associated with a Signer within the R.Signer; 3) R.SVD: The public key is associated with a Signer within the R.Signer; 4) R.Signer: it has a unique identifier; 5) R.Reference_Priviliged_User_Authentication_Data: The public key of the Privileged User is associated with an identifier of this User.

With respect to the FIA_USB.1.2 requirement, an association is made between: a) The signer with the Privileged User authorized to create new signers. To create any action with the Signer's private keys (association of keys and use) it is necessary to authenticate the Privileged User through TLS. And in turn, the structure of the key pair (R.Signing_Key_ID) contains a field with the identifier of the Privileged User that can manage the keys; b) Only authorized Privileged Users can create new signers. The Privileged Users FPUs (those that are within the configuration and identified) are the only ones who can manage new Privileged Users.

Regarding the FIA_USB.1.3 requirement, the following associations are guaranteed:

a) Only the authorized Privileged User can modify the authorized R.Signer object. The R.Signer structure has the field "signing_service_privileged_user_id" with the identifier of the SSPU Privileged Usuer authorized to perform the modification. The R.Signer update can be performed with the following operations in administrative mode (that is, only with the authorized SSPU present): Assign_Key_Pair, Update_Key_Pair and Delete_key_pair.

All these operations require that the SSPU Privileged User authenticates with TLS and only if the SSPU User matches the SSPU in the "signing_service_privileged_user_id" field of the R.Signer structure will the operation be allowed.

b) The SSPU Privileged User with the presence of the Signer may make the modification of their own R.Signer through the following operations: Assign_Key_Pair, Update_Key_Pair, Sign_DTBS (the R.Signer is re-signed and returned in the response in case the R.Signer of the user is signed with an old key from the history), Delete_key_pair.

For all these operations to be carried out by the authorized SSPU Privileged User with the presence of the Signer, it is necessary that the request be made with SKAD or SAD (in the case of Sign). The Asset R.Signer includes the "subject" field that includes the identified user (with his domain), and this field must coincide with the "sub" field of the SKAD or SAD.

The TOE require each user to be successfully identified and authenticated before allowing any operation implemented by the TSF on behalf of that user. Additionally, a secure channel must be established implemented by the TSF functions associated with FTP_TRP.

With respect to the components FIA_UAU.1 and FIA_UID.2, and regarding the Private Configuration Creation operation, to protect the cryptographic keys that make up this configuration, a quorum of the OCS First Privileged User and another quorum of the OCS Operation Privileged User are requested consecutively, this being the mechanism by which these two OCS (that must have been previously created in the HSM Security World and be different) are defined as such. That is, the authentication of the OPU and FPU users occurs because in the installation and start-up steps it is indicated that they must have been previously created by the HSM, and the TOE is the one who requests that it occur, but it is precisely in this operation where they are assigned to the system. The TOE requires this authentication to occur through the HSM.

With respect to the FIA_UAU.5 component, in relation to the privileged users FPU and OPU, an authentication by means of OCS cards is enforced. These privileged users are precisely those who own the OCS cards of the HSM that identify the role, and allow them to authenticate when the quorum of the cards is achieved.

When a PU needs to authenticate using the TLS protocol with client authentication, the client presents to the server a certificate. If the certificate do not belong to a valid PU (at first attempt), the connection will be rejected. PU users will be identified through the client TLS certificate used during the TLS handshake once it is accepted as a valid Privileged User.

When a user gets wrong in the authentication credential the number of times indicated in the TOE static configuration, then its signature key becomes disabled.

This TOE authenticates any signer's claimed identity according to [EN 419 241-1] SCAL2 for qualified signatures. The signer authentication scheme that the TOE implements can be (1) a delegated scheme (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 signed by an authorized IdP/Authorization Server that the signer has been authenticated); or (2) a mixed scheme (external factor plus an signer’s secret). In case (1), the TOE verifies the assertion. In case (2), the TOE verifies the assertion and it also verifies the signer´s secret. The signed assertion ensures at least two factors of authentication.

6.5 Security Management

FMT_MSA.1/Signer Management of security attributes, FMT_MSA.1/Privileged User Management of security attributes, FMT_MSA.2 Secure security attributes, FMT_MSA.3/Signer Static attribute initialisation, FMT_MSA.3/Privileged User Static attribute initialisation,, FMT_MTD.1 Management of TSF data, FMT_SMF.1 Specification of Management Functions, FMT_SMR.2 Restrictions on security roles

The TOE guarantees that the management of security attributes allows authorised users to manage the specified security attributes, ensures that values assigned to them are valid with respect to their secure state, and ensures that the default values of security attributes are restrictive.

Only Privileged Users, properly identified and authenticated with their credentials, can modify the TOE configuration data.

The TOE associates each user with an identifier and credentials, and guarantees that a Signer registered in the system cannot be a Privileged User.

Regarding the FMT_MSA.1/Signer component, below it is detailed how the TOE fulfills the related requirement, depending on each operation of the requirement:

- For the key generation operation, only an authorized SSPU can perform it. When the key is generated, then the asset R.Key_Pair is created, already containing the security attributes R.Reference_Signer_Authentication_Data (reference to the authorized SSPUs), R.Signing_Key_Id and R.SVD.

- For the Key Assignment operation, only an authorized SSPU can access it. In addition, the permission of this assignment is optionally requested from the Signer through SKAD.
- To perform the signer's key delete operation, the TOE requires the intervention of an SSPU and an authorized Signer.
- The signature generation operation (Supply DTBS / R SFP and Signing SFP) can only be requested by an authorized SSPU with express authorization from the Signer. Through the asset R.SAD you get the R.DTBS/R of the Signer.
- The Signing SFP operation does not apply, since the TOE does not allow queries of the attributes (the TOE does not manage these attributes or structures).
- Regarding the operation of Signer Maintenance SFP, the asset includes the public key of the authorized ASs. The reference to these AS keys are the identifier of the SSPU and are within the R.Signer asset. This is because each SSPU is assigned one or more trusted ASs in the dynamic configuration. And this dynamic setting can only be edited by a trusted OPU. That is: a Signer will never be able to update R.Reference_Signer_Authentication_Data asset (the TOE does not allow it); and only authorized OPUs can change the R.Reference_Signer_Authentication_Data asset by changing the keys of the authorized ASs in the dynamic configuration.

Regarding the FMT_MSA.3 / Privileged User component, clarify that it does not apply to the creation of PKI keys, since default values cannot be provided for this type of key.

TOE ROLES

First Privileged User (FPU)

First Privileged User users are the holders of the HSM OCS First Privileged User cards. Thus, each First Privileged User can be considered to hold a portion of the First Privileged User role. Consequently, a minimum of First Privileged User users are required to perform any of the operations that require a quorum of the OCS First Privileged User.

The main functions that Privileged Users FPUs can carry out are the following: 1) creation of the TOE, 2) initialization of the TOE, 3) deletion of the TOE, 4) renewal of the TOE keys, and 5) import of the TOE configuration.

Operation Privileged User (OPU)

The Operation Privileged User users are those who have the OCS Operation Privileged User cards of the HSM. Consequently, a minimum number of Operation Privileged User users are required to perform any of the operations that require a quorum of the OCS Operation Privileged User.

The functions that an OPU Privileged User can perform are the following: 1) start the TOE, 2) stop the TOE and 3) import the TOE dynamic configuration signed by an OPU Privileged User.

Configuration Privileged User (CPU)

In order to these users be recognized as such by the TOE, their public signing keys must be included in the TOE static configuration, being this inclusion the mechanism by which the TOE recognizes the Configuration Privileged User as such.

Configuration Privileged User privileges carry out the following functions on the TOE: 1) signing the dynamic configuration of the TOE, 2) defining the Privileged Users SSPU (Signing Service Privileged User) that the TOE recognises, and 3) defining the components Authorization Server that recognizes the TOE.

Signing Service Privileged User

In order for the TOE to recognize SSPU Privileged Users, previously, the public authentication keys of these entities must be included in the dynamic configuration of the SAM.

Signing Service Privileged User entities are the ones who can invoke the following TOE service functions: 1) signature generation (with the consent of the signer and the R.DTBS/R), 2) request for the generation of keys, 3) key assignment to the signer, and 4) deletion of the key.

Signer

Signer users are the owners of the signing keys whose use is controlled by the TOE.

These users can perform the following functions: 1) signature generation through the R.SAD that contains the R.DTBS/R, 2) key pair generation, and 3) key deletion.

6.6 Protection of the TSF

FPT_PHP.1 Passive, FPT_PHP.3 Resistance

The TOE ensures unauthorised modification of R.TSF_DATA by means of the detection of physical tampering that might compromise the TSF. The TOE detects the opening of the hardware where the TOE is installed, and in this tampering situation the TOE carries out the following actions:

- Block new incoming requests.
- Complete processing requests that were in progress.
- Delete the keys from CM memory and session.
- Delete the infrastructure keys from CM memory and session.
- Stop the TOE service.
- Stop the operating system (if this is configured), so the computer is stopped.
- Generation of a signed event log by the TOE before stopping the service.

When the hardware is restarted, then an authorized OPU Privileged User has to start the service again (it is the only privileged user who can do this action).

The hardware on which the TOE is installed includes a BMC (Baseboard Management Controller) chipset, which communicates with the Operating System through the standard IPMI (Intelligent Platform Management Interface) interface. The TOE accesses the intrusion sensor through the Operating System libraries.

FPT_RPL.1 Replay detection, FPT_STM.1 Reliable time stamps

Time stamping is a way of preventing a replay attack. The R.SAD asset includes a time stamp, the DTBS/R data and the signing key id. All this information is signed by an Authorization Server trusted by the SAM, and therefore, this is protected with integrity. The TOE only accepts R.SAD assets for which the time stamping is within a reasonable tolerance using the reliable TOI (Time of Interest) from TOE environment time source. The SAD guarantees that the TOE can only sign the specific data with the specific signing key id included in the R.SAD asset for a specific amount of time configured in the static configuration. When the TOI exceeds the amount of time configured from the R.SAD time stamp, the R.SAD is considered as expired and the TOE rejects the signature operation. The TOE environment uses a NTP client that is synchronized with a NTP server in order to issue a reliable time source. The reliable time stamps are used in the events logs generated by the TOE.

In addition to this guarantee, the TOE uses the TLS communication protocol, and establishes a new session for each request.

FPT_TDC.1 Inter-TSF basic TSF data consistency

All assets referenced in the FPT_TDC.1.1 requirement are included in structures signed by the TOE or by the SA component. Furthermore, the communication protocol between the SSA and the TOE is TLS version 1.2 thereby ensuring the integrity of communication between this trusted component of IT and the TOE. Therefore, the TOE uses the integrity of the data both in the structures that store the referenced assets and in the communication channel through which this data is sent.

6.7 Trusted Paths/Channels

FTP_TRP.1/SSA Inter-TSF Trusted path

The TOE ensures that modification of any TOE asset is authorised by a Privileged User and that unauthorised modifications can be detected. A mutual TLS (v1.2) channel is established between the SSA component and the TOE, consisting of an encrypted channel between a Privileged User and the TOE, with origin authentication at both ends. All operations / services offered by TSF from SSA use this protected channel.

FTP_TRP.1/SIC Inter-TSF Trusted path

The TOE implements a server-side endpoint of a SAP (Signature Activation Protocol), which provides signer authentication, integrity and confidentiality of the transmitted SAD. The TOE ensures Signature Activation Data is protected against attacks when transmitted to the TOE which would compromise its use for authentication. In addition, the TOE guarantees that the R.DTBS/R is protected in integrity when transmitted to the TOE.

The R.SAD generated by the signer during signature activation protocol (SAP) execution is protected in integrity by a digital signature and, when it contains R.Authorisation_Data from the signer, it is encrypted (using the public key of the SAM) for confidentiality.

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

FTP_ITC.1/CM Inter-TSF trusted channel

The TOE establishes an exclusive path between the TOE and the CM based on secure communication with origin authentication from both ends, communication integrity and data confidentiality. This secure communication is done through the proprietary protocol of nCipher Impath.

7 Bibliography and acronyms

For the purposes of this document, the symbols, abbreviations, terms and definitions given in [CEN EN 419 241-1], [CEN EN 419 241-2] and [eIDAS] article 3 apply.

7.1 Bibliography

The following documents are referenced in this document:

Reference	*Referenced document*
[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.
[Formats]	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.
[IMPREG 2015/1502]	COMMISSION IMPLEMENTING REGULATION (EU) 2015/1502 of 8 September 2015 on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market.
[AIS31]	A proposal for Functionality classes for random number generators. Version 3.0 BSI.
[CEN EN 419 241-1]	CEN EN 419 241-1 Trustworthy Systems Supporting Server Signing; Part 1: General System Security Requirements.
[CEN EN 419 241-2]	CEN EN 419 241-2 Trustworthy Systems Supporting Server Signing; Part 2: Protection Profile for QSCD for Server Signing.
[CEN EN 419 221-5]	CEN EN 419 221-5 Protection Profiles for TSP Cryptographic Modules; Part 5 - Cryptographic Module for Trust Services.
[ETSI EN 319 411-1]	ETSI, Electronic Signatures and Infrastructures (ESI); Policy and security requirements for trust service providers issuing certificates; Part 1: General requirements.
[ETSI EN 319 411-2]	ETSI, Electronic Signatures and Infrastructures (ESI); Policy and security requirements for trust service providers issuing certificates; Part 2: Requirements for TSP issuing EU qualified certificates.
[ETSI TS 119 312]	ETSI, Electronic Signatures and Infrastructures (ESI); Cryptographic Suites.
[SOGIS]	SOG-IS, SOG-IS Crypto Evaluation Scheme, Agreed Cryptographic Mechanisms.

Reference	*Referenced document*
[ISO/IEC 19790]	ISO/IEC 19790:2012 Information technology – Security techniques – security requirements for cryptographic modules.
[CC2]	Common Criteria for Information Technology; Security Evaluation - Part 2: Security functional components. April 2017, Version 3.1, Revision 5

7.2 Acronyms

The following abbreviations are used in this document:

Acronym	*Meaning*
CA	Certification Authority
RA	Registration Authority
CSR	Certificate Signing Request
SVD	Signature Verification Data
SCD	Signature Creation Data
DTBS	Data To Be Signed
DTBS/R	Data To Be Signed Representation
SAD	Signature Activation Data
SAM	Signature Activation Module
SAP	Signature Activation Protocol
SSA	Server Signing Application
SIC	Signer's Interaction Component
SCAL	Sole Control Assurance Level
QSCD	Qualified Electronic Signature (or Electronic Seal) creation device as defined in the eIDAS Regulation
SCDev	Signature Creation Device
TW4S	Trustworthy System Supporting Server Signing
PP	Protection Profile
ST	Security Target
EAL	Evaluation Assurance Level
TOE	Target Of Evaluation
TSF	TOE Security Function
SFR	Security Functional Requirements
SFP	Security Function Policy

Acronym	*Meaning*
OSP	Organizational Security Policy
TSP	Trust Service Provider
CEN	Comité Européen de Normalization (European Committee for Standardization)
TC	Technical Committee
ETSI	European Telecommunications Standards Institute
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
CC	Common Criteria, ISO/IEC 15408, Evaluation criteria for IT security
RAP	Registration Provider
PKI	Public Key Infrastructure
HSM	Hardware Security Module
CM	Cryptographic Module
SCA	Signature Creation Application
AS	Authorization Server
REST	Representational State Transfer
API	Application Programming Interface
IdP	Identity Provider
SAML	Security Assertion Markup Language
OTP	One-Time password
TLS	Transport Layer Security
LoA	Level of Assurance