TÜVIT

Certificate

The certification body of TÜV Informationstechnik GmbH
hereby awards this certificate to the company

procilon GmbH
Leipziger Straße 110
04425 Taucha, Germany

Certificate validity:
2024-07-08 – 2028-12-05

to confirm that its qualified electronic signature and seal creation device

proNEXT SignatureActivationModule, Version 1.0.0

fulfils the requirements laid down in

Annex II of Reg. (EU) No. 910/2014 (eIDAS).

The requirements are summarized in the appendix to the certificate.

The appendix is part of the certificate with the ID 9804.24 and consists of 6 pages.

Essen, 2024-07-08

Dr. Christoph Sutter, Head of Certification Body

TÜV Informationstechnik GmbH
Am TÜV 1 ▪ 45307 Essen, Germany
tuvit.de

Certification scheme

The certification body of TÜV Informationstechnik GmbH is notified as certification body according to article 30.2 of "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" by "Bundesnetzagentur" (Germany).

The certification body performs its certification for qualified signature / seal creation devices (QSCD) based on the following certification scheme:

- "Certification Process for eIDAS conformant QSCDs of the certification body of TÜV Informationstechnik GmbH", Version 1.3 as of 2024-06-26; the current version can be downloaded at: www.tuvit.de/en/services/eid-trust-services/qscd/

The Certification Process for eIDAS conformant QSCDs makes use of the alternative method according to article 30.3 (b) of eIDAS.

Evaluation / Certification report

- "Evaluation Technical Report Summary (ETR Summary) proNEXT SignatureActivationModule, Version 1.0.0" V1 as of 2024-07-01, TÜV Informationstechnik GmbH – Evaluation Body for IT Security

Evaluation requirements

The evaluation requirements are defined in:

- Annex II of 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

Evaluation target

Evaluation target is the Qualified Electronic Signature and Seal Creation Device (QSCD) „proNEXT SignatureActivationModule", Version 1.0.0.

Description of the evaluation target

The QSCD consist of a software component (short TOE) in a dedicated protected environment and a cryptographic module (HSM) certified against EN 419 221-5. It is a remote QSCD where the qualified trust service provider manages the electronic signature or seal creation data on behalf of a signatory.

The TOE is the software product "proNEXT SignatureActivationModule" (SAM), which implements the signature activation Protocol (SAP). The SAM ensures that the signer has the sole control of his signing keys. It uses the Signature Activation Data (SAD) to activate the corresponding signing or seal key for use in a cryptographic module. The SAD binds together the signer authentication with the signing or sealing key and the data to be signed or sealed.

The TOE is deployed in a dedicated tamper protected environment that is connected to the HSM via a trusted channel.

Delivery of the evaluation target

The TOE including the TOE documentation is composed in a software zip-archive, which is handed over by the TOE manufacturer via personal delivery on a DVD to the customer or is made available via download. The integrity of the delivered TOE has to be checked comparing the SHA512 hash values of the TOE.

No.	Type	Item / SHA-512 Hash Value	Description
1.	SW	SAM Service (file name: SAMService1.1.1.tar.gz) 4ded16bd3625c5b24f83bfee028a2091b013c70d67fc917edfeab22a6412685a965627f46654353e0e2acb323558a4caa8e0167f8b92b414780ba550a7b799f9	TOE binaries: Source code
2.	SW	SAM Firmware (file name: SAMFirmware-1.0.0.tar.gz) fe24d04eeeb5b1fb366f7078042ce8b00f7ca64b77c3448437539b31cfc36d6a0275ed5e115508cd34d042c1e0e77db17a0af4a2c13e20cf2e9634def1fad3d4	TOE binaries: Firmware module for the cryptographic module
3.	SW	ManagementCLI of SAM (file name: manageSAM-1.0.0.tar.gz) 039ea49567d1a19efbba969253ad635fbd5f1cf3fc68b2cf41976dbaaae528f8b70ac6a5bc89751adf775252f23ce75f93cd3265a5dcbfcc4a7abd7df67c3aff	ManagementCLI: manageSAM.sh manageFW.sh checksumSAM.sh
4.	SW	TimeStatusMonitor (file name: TimeStatusMonitor.zip) ecddd3089b9dde877afb671ea5f75449d7aa930ac0eaf5e37b4df5be5bf6cdab51398e6cd978808cc204c2731579b05ee0adfbbf477d26db585b2f3682a27aef	Shell script (v1.2.0)
5.	DOC	Installation Guide (file name: AGD_proNEXT-SAM_Installation-Guide_1.5c.pdf) 4bd1074b8ac9ab6c2a0a11543a9603817b286f1e663b31ee68c848e77558972b9ea78de4c13bb03b409ef92a54a125adb5fb4a4688a20e558896cac00bf148ac	Part of guidance documentation
6.	DOC	Operational User Guide (file name: AGD_proNEXT-SAM_Operational-User-Guide_1.5b.pdf) b2f39d05bb3833edfc94cbddd4ccabc9fee5db17ea6369812e6ba1416024ec56ac278b61f4b1f494b5f503e6d642ad48f6470ca25f0d5ba532cf73ecb3f68374	Part of guidance documentation
7.	DOC	TOE Specification (file name: ADV_proNEXT-SAM_TOE-Specification_1.4b.pdf) 6c73e5761bbeb4d780c1cebb6638cf2eed6abef4a2d1810886686cf74c5d40cf550240258393064bb46d1aeeeec36a6b6a5fe8621ef150	Part of guidance documentation

No.	Type	Item / SHA-512 Hash Value	Description
		251edb97e938624a26	

Evaluation result

- The target of evaluation fulfills all applicable evaluation requirements
- The certification requirements defined in the certification scheme are fulfilled.
- The operating conditions listed in the certification report shall be respected.

Summary of the Evaluation Requirements

Annex II of eIDAS contains the following requirements for QSCDs:

1. Qualified electronic signature and seal creation devices shall ensure, by appropriate technical and procedural means, that at least:

   (a) the confidentiality of the electronic signature or seal creation data used for electronic signature or seal creation is reasonably assured;

   (b) the electronic signature or seal creation data used for electronic signature or seal creation can practically occur only once;

   (c) the electronic signature or seal creation data used for electronic signature or seal creation cannot, with reasonable assurance, be derived and the electronic signature or seal is reliably protected against forgery using currently available technology;

   (d) the electronic signature or seal creation data used for electronic signature or seal creation can be reliably protected by the legitimate signatory against use by others.

2. Qualified electronic signature and seal creation devices shall not alter the data to be signed or prevent such data from being presented to the signatory prior to signing.

Operational conditions

The following operational conditions must be fulfilled:

- The TOE must be implemented within the environment of a qualified Trust Service Provider, which fulfils the requirements as specified in the eIDAS.
- The TOE must be operated as a part of server signing system as specified in EN 419 241-1:2018; Trustworthy Systems Supporting Server Signing - Part 1: General System Security Requirements.

- For the cryptographic key generation and cryptographic operations the CC certified HSM of the model family 'CryptoServer Se-Series Gen2 CP5' (CC certificate with number NSCIB-CC-2300142-01, valid until 2028-12-05 must be installed and configured and used as randomness source for the proNEXT SignatureActivationModule.

- In the local environment of the signer the software component proNEXT SAK Operations must be used to generate the signature activation data and to communicate with the Signature Signing Application. This software component is used as Signers Interaction Component (SIC) (as defined in EN 419241-2:2019).

- An installed Signature or Seal Creation Application which consists of the Server Signing Service and the user interface which displays documents to be signed or sealed and other relevant data for the Signer such as the document hash, the signing keys chosen for the signature or seal creation and the assigned signer certificate.

- The proNEXT SignatureActivationModule server must be synchronized to a trusted time source.

- Only trustworthy, well-trained personal must be assigned to perform administrator duties.

- Administration tasks must be performed with dual control.

- The network-based and channel-based security must be configured in order to protect the transmitted DTBS/R from the disclosure.

- Signers which must be identified and registered for server signing and sealing have to authenticate themselves at the TOE for each server signing or sealing process.

- Before starting the initial operation of the TOE, TÜVIT evaluation body has to repeat the evaluation tests in the environment of the qualified TSP and has to provide an evaluation report to the TÜVIT certification body. The evaluation tests were successfully passed for the following qualified TSP(s):

    - Bundesnotarkammer, Burgmauer 53, 50667 Köln, Germany.

Algorithms and associated parameters

For the creation of qualified electronic signatures and seals, the TOE uses the cryptographic algorithms:

- RSASSA-PSS with 3072/4096 Bit Key Length according to PKCS#1: RSA Cryptography Specifications, Version 2.2 as of November 2016 (RFC8017)

- ECDSA with 256/384/512 Bit Key Length using Brainpool Curves

Evaluation Assurance Level

The TOE has been evaluated according to Common Criteria by the TÜV Informationstechnik GmbH – Evaluation Body for IT Security. The results of the evaluation are documented in the evaluation technical report (ETR) “Evaluation Technical Report Summary”, Version 1 as of 2024-07-01. This was based on the security target “Security Target proNext Signature Activation Module”, Version 1.6 as of 2024-06-12 which is attached to this certificate in Annex I. It contains information on the certification of the HSM and take into account the requirements from the certified protection profiles:

- EN 419 221-5:2018, Protection profiles for TSP Cryptographic Modules - Part 5: Cryptographic Module for Trust Services

- EN 419241-2:2019 Trustworthy Systems Supporting Server Signing - Part 2: Protection Profile for QSCD for Server Signing.

The TOE security assurance requirements are based entirely on the assurance components and classes defined in part 3 of Common Criteria (see part C of this report or [CC] Part 3 for details). The TOE meets the assurance requirements of assurance level EAL 1 (Evaluation Assurance Level 1) augmented with ADV_FSP.2 (Security-enforcing functional specification) and ADV_TDS.1 (Basic Design).

Validity of the certificate

The certificate with the number TUVIT.9804.QSCD.07.2024 is valid to a maximum of 5 years, provided that a vulnerability assessment is carried out every two years. If vulnerabilities are identified and not remedied, the certification will be cancelled.

The validity period of the certificate with the number TUVIT.9804.QSCD.07.2024 depends on the validity of the CC certificate of the crypto module, the strength of security mechanisms and algorithms that are implemented in the product and is limited 05th December 2028 at maximum.

The period of validity may be extended or shortened at a certain point in time if there is new knowledge about the validity of the crypto module's CC certificate and the suitability of the security mechanisms or algorithms.

Attachments

Attached is the public version of the security target:

"Security Target proNEXT SignatureActivationModule".

Author: procilon GmbH

Datum: 2024-06-12

Version: 1.6

.

Security Target

proNEXT SignatureActivationModule

State: 12.06.2024

On behalf of
procilon GROUP

procilon GmbH
Leipziger Strasse 110
04425 Taucha bei Leipzig
Germany

Versioning

Version	Date	Description	Edited by
1.0	10.09.20	Initial Creation, Adaption, Review, Extension, Finalizing	H. Werner, O. Kube
1.1	11.09.20	Adaption according to Observation Reports	H. Werner, O. Kube
1.2	15.09.20		H. Werner, O. Kube
1.3	28.10.20	Adaption Physical Scope of the TOE, Cryptographic Support, Abbreviations	H. Werner, O. Kube
1.4	29.10.20		H. Werner, O. Kube
1.5	14.11.23	Adaptions regarding new CC Cert of CP5	H. Werner, O. Kube
1.6	12.06.24	Adaptions regarding Seals, Authentication Mechanisms, Delivery, RSA Key Length, Signer Enrolment, Wording, References, Abbreviations	H. Werner, O. Kube

Table of Contents

List of tables

List of Figures

1. Introduction (ASE_INT)

This ST should serve as a basis for a process evaluation in the field of remote signatures/seals.

ST Application Note 1

This document is based, among others, on the specification [EN419241-2] regarding 'Trustworthy Systems Supporting Server Signing' and uses its notation. Identifiers defined in [EN419241-2] and used in this document, in particular those for operations and subjects as well as objects of such a system, are, even contrary to their signature-oriented names, always to be understood for providing seals. Thus, as an example, the term 'Signer' for the signatory also denotes the user who creates and applies seals, just as the term 'Generate_Signer_Key_Pair' is not limited to the creation of a key pair for signatories alone, but also includes the creation of key pairs for sealing users.

1.1 Security Target Reference

This Security Target (ST) is identified by the following unique reference:

ST Title:	proNEXT SignatureActivationModule Security Target
ST Version:	1.6
ST Date:	2024-03-12
ST Author:	procilon GmbH

1.2 TOE Reference

The Target of Evaluation (TOE) is identified by the following unique reference:

TOE Name:	proNEXT SignatureActivationModule
TOE Version:	1.0.0
TOE Developer:	procilon GmbH
Confirmation ID:	TUVIT-9804-QSCD

1.3 TOE Overview

1.3.1 General Requirements

A trustworthy system supporting server signing (TW4S) is a system that offers remote digital signatures/seals as a service. It ensures Signer’s signing key or keys are only used under sole control of the Signer for the intended purpose.

ST Application Note 2

Sole control is no longer strictly guaranteed in the sense of [EN419241-2] when signing keys are used to provide seals. In contrast to the provision of a signature, where access and use of a specific signing key is restricted exclusively to a specific user or signatory, a group of users is authorised to access a specific signing key and use it for sealing.

In the case of creating seals, the term Signer refers to a user or signatory who is a member of a sealing group and thus authorised to create seals using signing keys assigned to the sealing group on behalf of a company or institution in accordance with existing organizational regulations.

Terms such as server signing, signing key, signing process/operation/service/architecture/system as well as terms based on [EN419241-2] (e.g. Server Signing Application) are to be understood in the sense of seals and their provision.

The TW4S uses a cryptographic module to generate the signing key and create the digital signature/seal value.

The system consists of a local and a remote environment. The Signer is in the local environment and interacts with the Server Signing Application (SSA) in the remote environment.

The purpose of the interaction between the Signer and SSA is to utilize the SSAs signing service. The signing operation is performed using a Signature Activation Protocol (SAP), which requires that Signature Activation Data (SAD) is provided at the local environment. The SAD binds together three elements: signer authentication with the signing key and the data to be signed/sealed (DTBS/R(s)).

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

One of the three SAD elements is the signer authentication. The signer authentication is assumed to be conducted according to [EN419241-1] SCAL.2 for qualified signatures/seals. This means signer authentication can be carried out in one of the following ways:

- Directly by the SAM or
- Indirectly by the SAM or
- by a combination of the direct and indirect schemes

The authentication is carried out indirectly by the SAM, an external authentication service as part of the TW4S or a delegated party, which verifies the Signer's authentication factor(s) and issues an assertion that the Signer has been authenticated. The SAM verifies the assertion. In the case there is a combination of the direct and indirect scheme, a part of the signer authentication is done directly by the SAM and another part is done indirectly by the SAM.

The SAM has to assume (on the environment) that part of or complete authentication has taken place and rely on an assertion. In this ST signer authentication means that the Signer has been authenticated in one of the three ways mentioned above.

The Signer is located in the local environment with a user interface. The user interface can display documents for the Signer. The Signer Interaction Component (SIC) is used to communicate with the Server Signing Application (SSA). The SSA forwards the communication from the SIC to the QSCD. Inside the QSCD the SAM receives the messages and optionally communicates with the SSA to obtain relevant data. When the SAM module has verified SAD, it can authorize the activation of the signing key within the cryptographic module and produce a digital signature/seal value. The value is returned to the SSA and may be further delivered to the SCA or SIC.

The SAM module is the TOE of this ST. The TOE and a cryptographic module certified against [EN419221-5] is required to obtain a QSCD.

The TOE generates audit records. It relies on the SSA to store audit records.

The TW4S relies on other services:

- Signers shall be identified and registered. This may involve the establishment of authentication mechanism for a Signer.
- Signing keys are certified by a Certification Authority.
- The Signature Creation Application is responsible for creating the signed/sealed document using the signature/seal values provided by the TW4S.

1.3.2 TOE type

The TOE is a software component, which implements the Signature Activation Protocol (SAP). It is deployed within a dedicated protected environment and can be used with the core components the CC certified BNotK Trustcenter 2.0 (CC certificate TUVIT.93204.TE.12.2015) is based on. The TOE is connected to the cryptographic module via a trusted channel.

It uses the Signature Activation Data (SAD) from the Signer to activate the corresponding signing key for use in a cryptographic module.

Together the TOE and cryptographic module are a QSCD.

1.3.3 TOE life cycle

The TOE life cycle consists of successive phases

- **Development**: the TOE developer develops the TOE application and its guidance documentation using any appropriate guidance documentation for components working with the TOE, including the cryptographic module.
- **Delivery**: The TOE is securely delivered from the TOE developer to the TSP.
- **Preparation**: the TSP installs and configures the TOE with the appropriate configuration and initialization data. Installation may allow creating the Privileged Users.
- **Operational use**: In operation, the TOE can be used by Privileged Users to create Privileged Users and Signers. Privileged Users can maintain TOE configuration. Privileged Users and Signers may generate signing keys for a Signer. Signers can supply the data to be signed/sealed to the TOE and authorize a signature/seal creation.

The TOE end of life is out of the scope of this document.

1.3.4 Usage and major security features of the TOE

The major security features of the TOE are:

System management

- Privileged User Admins can handle system configuration.

User management

- Privileged Users can create other Privileged Users
- Privileged Users and Privileged Users Technical can create Signers.
- Privileged Users or Signers can generate signing keys and signature verification data using a cryptographic module and assign the signing key identifier and signature verification data to a Signer.
- Privileged Users or Signers can update user data assigned to a Signer.

Signing operation

- Signers can supply a DTBS/R(s) to be signed/sealed.
- The SAD is securely exchanged with the TOE.
- Within the TOE the following actions are performed:
  - The SAD is verified in integrity.
  - The SAD is verified that it binds together Signer authentication, DTBS/R(s) and signing key identifier.
  - The Signer identified in the SAD is authenticated.
  - The signing key identifier is assigned to the Signer.
  - The TOE uses Authorization Data to activate the signing key within the cryptographic module.
  - The TOE uses the cryptographic module to create signatures/seals.

Audit

- An audit trail is produced of all security relevant events within the TOE. Management access to audit trail is outside the scope of the TOE.

1.3.5 TOE environment general overview

The TOE is expected to:

- operate as parts of server signing system as specified in [EN419241-1]
- be used by a TSP applying security policies as required by TSPs providing signature/seal creation services
- used in conjunction with TSPs issuing certificates

1.3.6 Required non-TOE hardware/software/firmware

The TOE needs, at least, the following hardware/software/firmware to operate:

- A Signature Creation Application (SCA) as mentioned in [EN419241-1] and [EN419241-2] that
  - manages the document to be signed/sealed and
  - transfers that to the SSA, either directly or through the SIC.
- A Server Singing Application (SSA) according to [EN419241-1] and [EN419241-2] that in particular handles the communication between the SAM and the SIC.
- A Signer Interaction Component (SIC) according to [EN419241-1] and [EN419241-2] used locally by the Signer to communicate with the remote systems.
- A cryptographic module as specified in [EN419221-5], supporting the operation of the TOE.
- An external Identity Provider that
  - is delegated by the TOE to perform the authentication of a Signer and
  - returns an ID token as result of a performed successful authentication.

1.4 TOE Description

1.4.1 Physical Scope of the TOE

The TOE is provided as a software archive accompanied by its guidance documentation.

The TOE is handed over by the manufacturer either by personal delivery or by provision via download. In the case of personal delivery, an employee of the manufacturer hands over a DVD with all the delivery components to the customer. If the TOE is provided via download, the manufacturer provides the customer with a link to a file containing all delivery components and appropriate access rights. To check the integrity and authenticity of the TOE and to run the TOE in secure operation the customer has to follow the instructions provided in the guidance documentation.

The TOE[1] is embedded into the following environment:

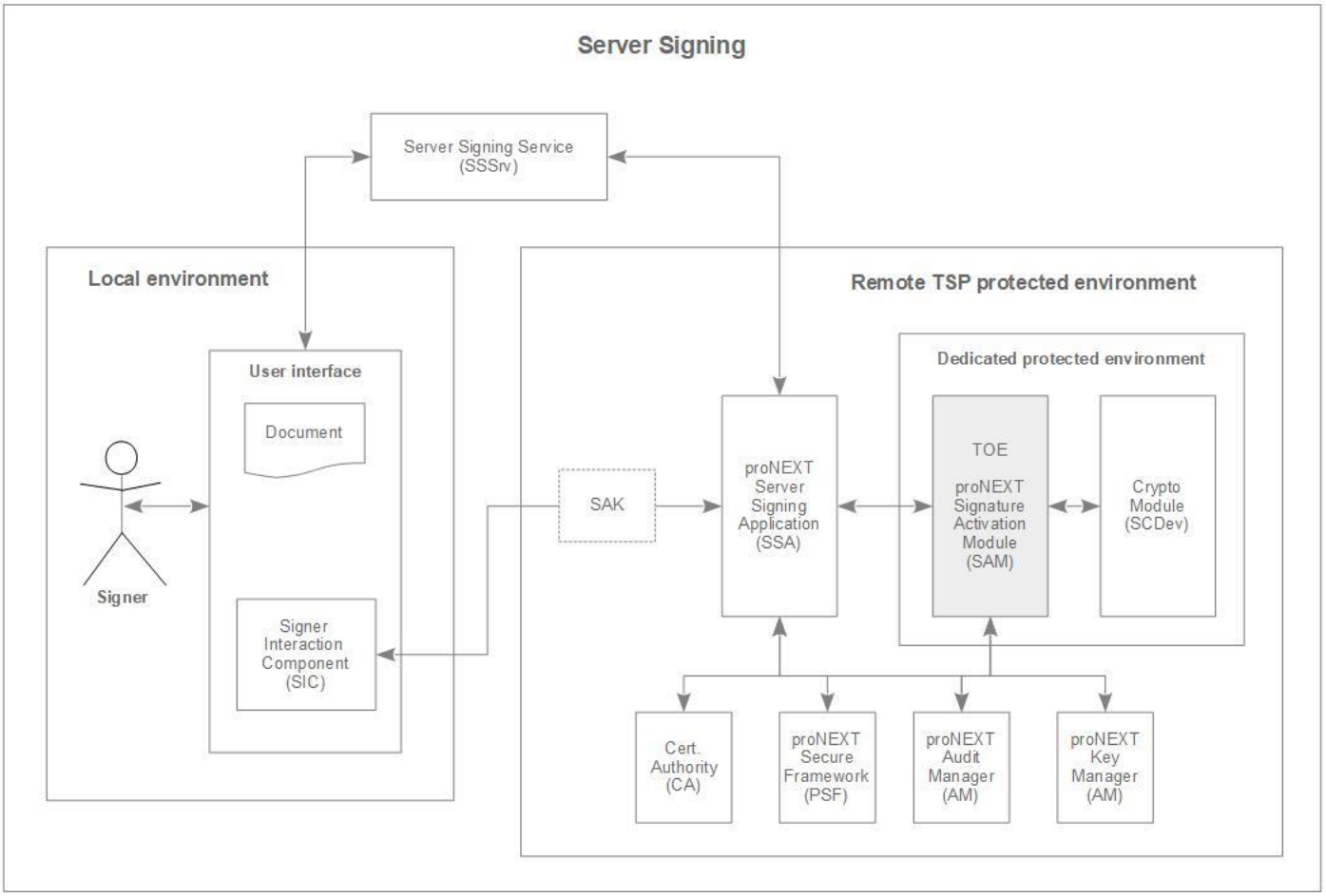


Figure 1: Overview of the TOE and its operational environment

The Signer is located in the local environment and uses a user interface which is provided by a module called Server Signing Service (SSSrv). In the context of remote signatures/seals, the SSSrv acts as the Signature Creation Application (SCA). The user interface displays the document to be signed/sealed and other relevant data for the Signer such as the document hash, the signing keys chosen for the signature/seal creation and the assigned signature/seal certificate.

The SIC provides the Signature Activation Data (SAD) and communicates with the SSA. The SSA interacts directly with the SAM and forwards communications from the SIC to it. It requires Signers to successfully identify and authenticate itself before allowing actions that may affect the SAM or signing keys.

Inside the dedicated protected environment the SAM receives messages send by the SSA, optionally communicates with the SSA to obtain relevant data and processes requests after verification. When the SAM has verified the SAD delivered by a Signer, it may authorize the activation of the signing key assigned to the signing operation within the cryptographic module and produce a digital signature/seal value. The value is returned to the SSA and after verification is further delivered to the SIC.

[1] It consists of three modules: the SAM Service, the SAM Firmware and the SAM Management. The SAM Firmware is integrated into the cryptographic module.

There are two variants for generating the Signature Activation Data (SAD). These differ both in where parts of the generation actually takes place and in the components that are used for it:

(SADGEN1) in this case, the SAD is generated completely by the SIC. In a first step, the SIC generates a SAD signing key. Next, the SAD is generated. Finally, the SAD data structure is signed with the SAD signing key and transmitted to the SSA.

(SADGEN2) In this case, the SIC partially uses a remote signature application component (SAK) to generate and transmit the SAD. The SIC first generates a SAD signing key. The SIC then requests the generation of the SAD at the remote SAK. The SAD supplied by the SAK is then signed by the SIC with the SAD signing key and finally transmitted to the remote SAK which forwarded it to the SSA.

In both cases, the SIC is responsible for generating the SAD signing key and signing the SAD and thus linking the remote signature/seal-relevant components (document(s) to be signed, remote signing key used by the signer to sign/seal, data that authenticates the signer) together.

As can be seen in the figure above the TOE environment consists of the following modules:

Module	Description
proNEXT Audit Manager (AM)	Is a service for providing audit management functions for the components of the server signing architecture. Its functions include the delivery of audit entries via the REST interface, verification and export of the audit log, ensuring integrity protection in databases and configurations. The Audit Manager is used in particular by the Server Signing Application (SSA) and the Signature Activation Module (SAM).
Certification Authority (CA)	Provides certificate services in the sense of a certification authority. In particular, it provides additional services for a TW4S, such as • identification and registration of signers and • certification of signing keys
Cryptographic Module (SCDev)	Used to create both signer signing keys and signatures/seals (signature/seal values) requested by the signer. Is located in a specially secured TW4S remote environment. A HSM of the model family 'CryptoServer Se-Series Gen2 CP5' (CC certificate number NSCIB-CC-2300142-01) is to be used.
proNEXT Key Manager (KM)	Provides functions that enable the creation, management and retrieval of key material. Objects managed by KM, so called managed objects, consist of a unique ID and a binary, which can represent certificates, public keys and user objects, among other things. Links between the managed objects are used to link them and can, for example, represent relationships such as the ownership.
SAK	Signature application component for creating and verifying electronic signatures. Is to be used when the SIC partly uses a remote SAK for generating the SAD. Is situated in the remote TSP protected environment. Manufacturer-independent SAKs can be used, provided that they implement the corresponding interfaces to the SIC as well as the SSA.
proNEXT SecureFramework (PSF)	CC certified signature application component (CC certificate: TUVIT.93200.TE.07.2016). Verifies the certificates generated during key pair generation. Collects certificate information for this purpose, evaluates it, and generates reports based on the checks.
Signers Interaction Component (SIC)	Software that is installed in the signer's environment. Participates in the signature activation protocol (SAP) and the provision of the SAD. Establishes the link between the signer and the signing process.
proNEXT Signature Activation Module (SAM)	A control unit for the cryptographic modules that is located within a dedicated environment. Registers users, initiates the generation of signing keys. Is responsible for executing the signing process and verifying the SAD. Provides its own database. With the help of the information stored there about signers and authentication factors, it is

	ensured that only the actual owner of a key can access it and thus use it for remote signing/sealing. The SAM further activates the signing key against the cryptographic module. It consists of three modules: the SAM Service, the SAM Firmware and the SAM Management. The SAM Firmware is integrated into the cryptographic module.
Server Signing Application (SSA)	Acts as a kind of proxy for the controlled addressing of the functionality of the SAM and provides via it an interface to the cryptographic module for generating, holding and using signing keys. All requests to the SAM by the SIC are received, pre-screened, and routed appropriately by the SSA. Signatories shall successfully identify and authenticate themselves before the SSA permits any actions involving the SAM. The SSA may maintain signer authentication for a specified period of time and/or for a specified number of signatures. In addition, the SSA creates audit records and passes them to the Audit Manager to manage audit logs.
Signature Creation Application (SCA)	A Service which makes it possible to perform the registration for Server Signing. Provides the UI for the user and the functions to manage them. Software represented by the Server Signing Service (SSSrv) in figure 1.

Table 1: Modules of the TOE environment

In addition, together the modules Server Signing Application (SSA), Signature Activation Module (SAM) and Cryptographic Module (SCDev) are the trustworthy system supporting server signing (TW4S).

1.4.2 Logical Scope of the TOE

The TOE is a software component, which implements the functionalities of a Signature Activation Module (SAM) within a trustworthy system supporting server signing (TW4S).

The main usage of the TOE is for the management of users, the signing operation and the system which provides the signing operation remotely. This results in the following major security features.

System management

Privileged User Admins can handle the system configuration. To do this, they must authenticate themselves against the TOE and use a secure channel to transmit information to the TOE to manage the configuration of the TOE. Managing the TOE configuration corresponds to the TOE usage scenario TOE Maintenance.

User management

Privileged Users and Privileged Users Technical are able to create Signers. The TOE allows users to register for the use of the remote signature/seal service to become a Signer. When the identification of the user is performed successfully, the TOE initiates the creation of the Signer.
Following this the key material for the Signer is generated within the SCDev. Based on the generated key material, the Signer's certificate is then issued by the CA and is assigned to the user. Creating a Signer corresponds to the Enrolment of Signers more specifically the TOE usage scenarios Signer Creation and Signer Key Pair Generation.

Privileged Users are able to create other Privileged Users. The Privileged User has to authenticate before performing the creation of another Privileged User and then initiates the registration process for a new Privileged User. The TOE checks the request for the registration of a new Privileged User and when valid the TOE creates a new entry to register the new Privileged User. Creating a Privileged User corresponds to the TOE usage scenario Privileged User Creation.

Privileged Users or Signers can generate signing keys and signature verification data using a cryptographic module. Signing key identifier and signature verification data can be assigned to a Signer. Both actions corresponds to the TOE usage scenario Signer Key Pair Generation that can be performed by a Privileged User or Signer. Signer Key Pair Generation performed by a Privileged User consists of the authentication of the Privileged User, the selection of the Signer, the signing key pair generation within the SCDev as also the issuance of the Signer's certificate by the CA. The TOE assigns the signing key identifier and signature verification data to a Signer. When a Signer performs Signer Key Pair Generation its part of the Enrolment of the Signer. The Signer also has to authenticate before performing the action but there is no need to select the Signer separately.

Privileged Users or Signers can update user data assigned to a Signer. Updating user data assigned to a Signer corresponds to the TOE usage scenario Signer Maintenance. The Signer Maintenance performed by the Privileged User consists of the authentication of the Privileged User, the selection of the Signer and the update of signer attributes. The TOE is returning a list of Signers to the Privileged User, checks the request for the Signer Maintenance and updates the entry of the Signer. When performed by a Signer the Signer also has to authenticate before performing the action but there is no need to select the Signer separately.

Signing operation

Performing the creation of remote signatures/seals is represented as TOE usage scenario Signing. Signing contains the authentication of the Signer, the SAD generation, the activation of the signing key and the signature/seal value creation. The SAD generation is done by the SAK, Signers can supply DTBS/R(s) during this process step, then the SAD is securely exchanged with the TOE. The TOE checks whether the Signer is authenticated, checks the validity of the signature of the SAD, checks the binding of the SAD parts Signer authentication, supplied DTBS/R and the signing key identifier, whether the signing key identifier within the SAD is assigned to the Signer. If the verification is successful, the signing key assigned to the signing process is activated within the cryptographic module based on authorization data. The cryptographic module is requested by the TOE to create signatures/seals.

Audit

The TOE does security audit. An audit trail is produced of all security relevant events within the TOE. Management access to audit trail is outside the scope of the TOE.

The main security functionalities the TOE provides to fulfill the major security features are

- Security Audit
- Cryptographic Operations
- Access Control
- Information Flow Control
- Self-Protection
- Trusted Paths/Channels

which handle the TOE usage scenarios, assigned users and operations more in detail as follows:

Signer

Security functionality	Description
Identification and Authentication	Requires that the signer is maintained by the TOE.
User Data Protection	Describes requirements for protecting signer assigned data in integrity when handled.
Security Management	Describes rules for creation, maintaining and usage of signer as well as requirements to its values.
Protection of the TSF	Requires the TOE to be able to interpret signer related data when shared with SSA.

Table 2: Security functionalities for the Signer

Authentication

Security functionality	Description
Identification and Authentication	Limits the amount of authentication attempts. Require that each user is identified and authenticated before any action on behalf of the user can take place. Describe the list of possible authentication mechanisms.
User Data Protection	Ensures that access control and information flow data are transmitted in a confidential way.

Table 3: Security functionalities for Authentication

Create Signer

Security functionality	Description
Identification and Authentication	Defines authorization rules for creating new signer.
User Data Protection	Describes access control requirements for creating a signer.

Table 4: Security functionalities for Create Signer

Signer Key Pair Generation

Security functionality	Description
Cryptographic Support	Describes rules for how signing key pair are generated.
User Data Protection	Describes access control requirements for creating a signer

Table 5: Security functionalities for Signer Key Pair Generation

Signer Key Pair Deletion

Security functionality	Description
Cryptographic Support	Requires that keys be securely destroyed.
User Data Protection	Describes the access control requirements for deleting signing key pairs.

Table 6: Security functionalities for Signer Key Pair Deletion

Signer Maintenance

Security functionality	Description
User Data Protection	Describes access control requirements for updating authentication related data of signer.

Table 7: Security functionalities for Signer Maintenance

Signing

Security functionality	Description
Cryptographic Support	Requires the TOE to perform cryptographic operation conformant with a ST specified list of algorithms.
User Data Protection	Describes requirements on preconditions for a signing operation to be carried out. Requires the SAD to be protected from modification and replay. Describes access control requirements for signing.

Table 8: Security functionalities for Signing

Privileged User

Security functionality	Description
Identification and Authentication	Requires that a privileged user is maintained by the TOE.
User Data Protection	Describes requirements for protecting privileged user assigned in integrity when handled.
Security Management	Describes rules for creation, maintaining and usage of the privileged user as well as requirements to its values.
Protection of the TSF	Requires the TOE to be able to interpret privileged user data when shared with a trusted IT product.

Table 9: Security functionalities for the Privileged User

Privileged User Creation

Security functionality	Description
Identification and Authentication	Defines authorization rules for creating a new privileged user.
User Data Protection	Describes access control requirements for creating a privileged user.

Table 10: Security functionalities for Privileged User Creation

TOE Maintenance

Security functionality	Description
User Data Protection	Describes access control requirements for maintaining the TOE.
Security Management	Requires the TOE to be able to carry out management functions and maintain users and roles.

Table 11: Security functionalities for TOE Maintenance

Audit

Security functionality	Description
Security Audit	Describes what shall be audited.

Table 12: Security functionalities for Audit

Communication

Security functionality	Description
Trusted Paths/Channels	Requires that all communication to the TOE comes from the SSA. Requires that either the Privileged User or the Signer initiates the communication.

Table 13: Security functionalities for Communication

More information on the security functionalities of the TOE is provided in chapter 6. The security objectives of the operational environment are described in chapter 4. The subjects that interact with the TOE as well as the assets which are protected by the TOE against threats are characterized in chapter 3.

2. Conformance Claims (ASE_CCL)

2.1 CC Conformance Claim

This ST is conformant to Common Criteria version 3.1 revision 5, referenced hereafter as [CC31R5].

More precisely, this security target is

- CC Part 2 extended,
- CC Part 3 conformant.

Which means that:

- For the description of the functional requirements addressed by the TOE, the security functional requirements of CC part 2 and additional security functional requirements introduced as extended component definition were used.

- For the description of the requirements due to the trustworthiness of the TOE, only security assurance requirements of CC part 3 were used.

2.2 PP Claim

This ST does not claim conformance with any Protection Profile (PP).

Nevertheless, the ST is based on the following PP:

- *Title: Vertrauenswürdige Systeme, die Serversignaturen unterstützen – Teil 2: Schutzprofil für qualifizierte Signaturerstellungseinheiten zur Serversignierung; Deutsche Fassung EN 419241-2:2019*
- *CC revision: v3.1 Veröffentlichung 4*
- *PP version: 1.0*
- *Authors: WG17*
- *Publication Date: 2019-05*
- *Keywords: Serversignatur*
- *Registration: DIN EN 419241-2:2019-05 (D)*

referenced hereafter as [EN419241-2].

2.3 Package Claim

The ST claims conformance to the Evaluation Assurance Level (EAL) 1, augmented by ADV_FSP.2 and ADV_TDS.1.

2.4 Conformance Rationale

As the ST does not claim conformance to a Protection Profile (PP), a conformance rationale is not required for that.

The conformance to Evaluation Assurance Level (EAL) 1, augmented by ADV_FSP.2 and ADV_TDS.1 was chosen to support a process-based evaluation of a remote signature scenario.

The conformance to [CC31R5] was chosen because it is the current revision and therefore is to be used.

3. Security Problem Definition (informal)

3.1 Assets

The TOE has the following assets which must be protected in terms of integrity and confidentiality as described below. The TOE shall ensure that whenever a value is outside the TOE, the TOE has performed the necessary encryption operations to enforce confidentiality and can detect whether a value has been changed. Access control to TOE values outside the TOE are to be enforced by the environment.

R.SIGNING_KEY_ID

The signing key is the private key of an asymmetric key pair for creating a digital signature/seal under the sole control of the Signer. The signing key can only be used through the cryptographic module. The TOE uses the value R.SIGNING_KEY_ID, which denotes a signing key in the cryptographic module. The binding of the R.SIGNING_KEY_ID with R.SIGNER shall be protected with regard to integrity.

R.AUTHORISATION_DATA

This is data used by the TOE to activate a signing key in the cryptographic module. The signing key is designated by R.SIGNING_KEY_ID. It shall be protected in terms of integrity and confidentiality.

R.SVD

Signature verification data is the public part associated with the signing key to perform the verification of the digital signature/seal. The R.SVD shall be protected with respect to integrity. The TOE uses a cryptographic module to generate the signing key pair. As part of the signing key pair generation, the cryptographic module provides the TOE with the values R.SIGNING_KEY_ID and R.SVD. The TOE provides the SSA with the R.SVD for further handling so that the key pair can be certified.

R.DTBS/R

A data set transmitted to the TOE for the creation of the digital signature/seal on behalf of the Signer. The DTBS/R(s) is transmitted to the TOE. The R.DTBS/R must be protected with regard to integrity. The transmission of the DTBS/R(s) to the TOE must require that the sending party is authenticated.

R.SAD

Signature activation data is a record involved in the signature activation protocol that activates the signature creation data to create a digital signature/seal under the sole control of the signer. R.SAD shall combine the following:

- the strong authentication of the signer as specified in [EN419241-1];
- if no special key is implied (e.g. a standard or unique key), a unique reference to R.SIGNING_KEY_ID;
- a given R.DTBS/R.

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

R.SIGNATURE

Is the result of the signing process and is a value of a digital signature/seal. R.SIGNATURE is created on the R.DTBS/R using an R.SIGNING_KEY_ID by the cryptographic module, under the control of the Signer as part of SAP. The R.SIGNATURE must be protected for integrity. The R.SIGNATURE can be checked outside the TOE using R.SVD.

R.AUDIT

These are records that contain logs of events that need to be audited. The logs are generated by the TOE and stored externally. R.AUDIT shall be protected with regard to integrity.

R.SIGNER

Is a TOE subject containing the set of data that uniquely identifies the Signer within the TOE. R.SIGNER shall be protected for integrity and confidentiality.

R.REFERENCE_SIGNER_AUTHENTICATION_DATA

This is the set of data used by the TOE to authenticate the Signer. It contains all data (e.g. serial number, protocol settings, etc.) and keys (e.g. verification key, etc.) used by the TOE to authenticate the Signer. This may include signature verification data or a certificate to verify a declaration provided as a result of delegated authentication. R.REFERENCE_SIGNER_AUTHENTICATION_DATA shall be protected for integrity and confidentiality.

R.TSF_DATA

This is the TOE configuration dataset used to operate the TOE. It shall be protected with respect to integrity.

R.PRIVILEGED_USER

Is a TOE subject containing the set of data that uniquely identifies a Privileged User within the TOE. It shall be protected for integrity.

R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA

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

R.RANDOM

Random secrets, e.g. keys, used by the TOE for operation and communication with external parties. It shall be protected in terms of integrity and confidentiality.

R.PRIVILEGED_USER_ADMIN

Is a TOE subject containing the set of data that uniquely identifies an Privileged User Admin within the TOE. It shall be protected for integrity.

R.REFERENCE_PRIVILEGED_USER_ADMIN_AUTHENTICATION_DATA

Is the set of data used by the TOE to authenticate a Privileged User Admin. It shall be protected in terms of integrity and confidentiality.

R.PRIVILEGED_USER_TECHNICAL

Is a TOE subject containing the set of data that uniquely identifies an Privileged User Technical within the TOE. It shall be protected for integrity.

R.REFERENCE_PRIVILEGED_USER_TECHNICAL_AUTHENTICATION_DATA

Is the set of data used by the TOE to authenticate a Privileged User Technical. It shall be protected in terms of integrity and confidentiality.

3.2 Subjects

The following subjects interact with the TOE.

Signer

The natural or legal person using the TOE through SAP, where it provides the SAD and can sign/seal DTBS/R(s) using its own signing key in the cryptographic module.

Privileged User

Performs administrative functions of the TOE and therefore is able to create users, for example.

Privileged User Admin

Privileged User, who is only authorised to install, configure and maintain the TOE. This role is maintained by the operating system of the server environment where the TOE is installed, not by the TOE itself.

Privileged User Technical

Privileged User, who is only authorised to create Signers.

3.3 Threats

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

3.3.1 Enrolment

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 signing 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 are threatened. Such data disclosure may allow a potential incorrect signer authentication leading to unauthorised signing operation on behalf of Signer.

T.SVD_FORGERY

An attacker modifies the R.SVD during transmission to the RA or CA. This results in loss of R.SVD integrity in the binding 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 [EN319411-1], 6.3.4 d) then an attacker can forge signatures/seals masquerading as the Signer.

3.3.2 User 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 signing operation on behalf of.

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/seal.

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 signing operation on behalf of Signer.

3.3.3 Usage

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.

The asset R.REFERENCE_SIGNER_AUTHENTIFICATION_DATA are threatened. Such data modification may allow a potential incorrect signer authentication leading to unauthorised signing 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/seal 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/seal without the Signer having authorised the operation. The asset R.SAD is threatened.

T.SAD_FORGERY

An attacker forges or manipulates R.SAD during transfer in SAP and is able to create a signature/seal 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.

T.DTBSR_FORGERY

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

T.SIGNATURE_FORGERY

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

3.3.4 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 are threatened.

T.AUTHORISATION_DATA_UPDATE

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

T. AUTHORISATION_DATA _DISCLOSE

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

The assets R.AUTHORISATION_DATA and R.SIGNING_KEY_ID are threatened.

T.CONTEXT_ALTERATION

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

The assets R.SIGNING_KEY_ID, R.SVD, R.SAD, R.REFERENCE_SIGNER_AUTHENTICATION_DATA and R.TSF_DATA are threatened.

T.AUDIT_ALTERATION

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

The assets R.SVD, R.SAD, R.SIGNER, R.REFERENCE_SIGNER_AUTHENTICATION_DATA, R.DTBS/R, R.SIGNATURE, R.AUDIT and R.TSF_DATA are threatened.

T.RANDOM

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

3.4 Relation between Threats and Assets

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

Asset	Dimension	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 of 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.SIGNATURE_REQUEST_DISCLOSE 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 14: Relation between threats and assets

3.5 Organisational 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/seals as well as the integrity and confidentiality of TOE assets.

3.6 Assumptions

A.PRIVILIGED_USER

It is assumed that all personal 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. Only algorithm, algorithm parameters and key lengths endorsed by recognized authorities as appropriate by TSPs shall be used. Guidance specifications are given in e.g. [EN319411-1] or for qualified certificate in e.g. [EN319411-2].

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 signing operation, i.e. protected against malicious code.

A.CA

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

A.ACCESS_PROTECTED

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

It is assumed that the operating system of the server where the TOE is installed is configured in such a way that remote access to the server is only possible for Privileged User Admins after a 2-factor authentication via an SSL-protected connection and only from the internal network, where the server is placed.

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

It is assumed that any audit generated by the TOE does not allow signing keys to be used and that any information needed to activate a signing key remains protected in integrity and confidentiality.

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 shall be ensured 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 and audited to be compliant with the requirements for TSP's given by this regulation.

A.SEC_REQ

It is assumed that the TSP establishes an operating environment according to the security requirements for SCAL2 defined in [EN419241-1].

4. Security Objectives (ASE_OBJ)

This section identifies and defines the security objectives for the operational environment of the TOE. These security objectives reflect the stated intent, counter the identified threats, and take into account the assumptions.

4.1 Security Objectives for the Operational Environment

OE.SVD_AUTHENTICITY

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

OE.CA_REQUEST_CERTIFICATE

The operational environment shall issue a certificate including R.SVD, signer information and CA signature.

The operational environment shall use a process for requesting a certificate, including R.SVD and signer information, and CA signature in a way, which demonstrates the Signer is in control of the signing key associated with R.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 management of signer 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 shall ensure that all requirements in [EN419241-1], SRA_SAP.1.1 are met.

In addition, the TSP should ensure that:

- the external 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 and
- if the Signer is only authenticated using a delegated party, 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 [EN419241-1], SRG_KM.1.1.

The evaluation of the qualified TSP in accordance with [EN419241-1] shall demonstrate that a delegated party meets the requirements of [EN419241-1] SRA_SAP.1.1. and optionally SRG_KM.1.1 if the Signer is authenticated by only one delegated party.

OE.DEVICE

The device 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 [EN419241-1]. It may be used to view the document to be signed.

OE.ENV

The TSP deploying the SSA and TOE should 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 evaluation of the qualified TSP shall reflect the safety objectives for the operational environment defined in this section.

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

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

OE.CRYPTOMODULE_CERTIFIED

If the TOE is implemented as a local application within the same physical boundary as the cryptographic module defined in [EN419221-5] then the TOE relies on the cryptographic module for providing a tamper-protected environment and for cryptographic functionality and random number generation.

If the TOE is implemented within a separate physical boundary then the TOE relies on the cryptographic module for cryptographic functionality and random number generation**.** The physical boundaries shall physically protect the TOE.

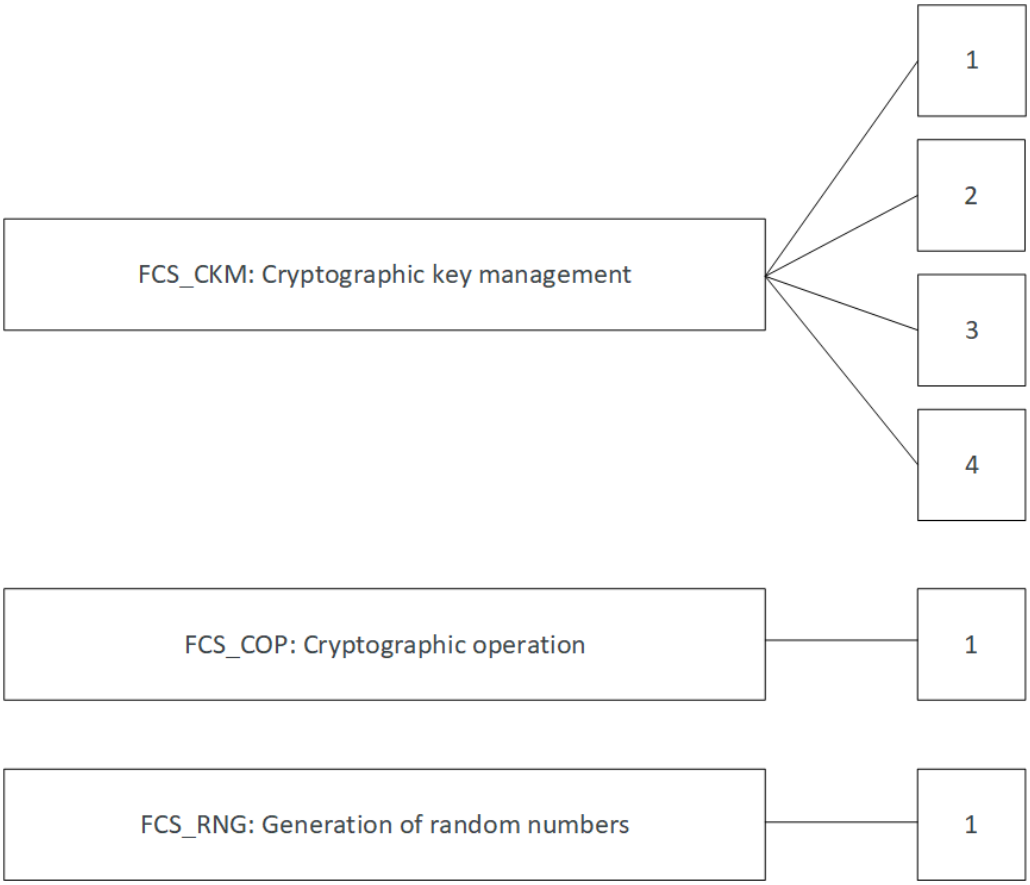
OE.TW4S_CONFORMANT

The TOE shall be operated by a qualified TSP in an operating environment conformant with [EN419241-1].

5. Extended Components Definition (ASE_ECD)

5.1 Class FCS: Cryptographic support

The FCS: Cryptographic support class, as defined in [CC31R5], is extended by a new family: Generation of random numbers (FCS_RNG). The family deals with the generation of random numbers. The following image shows the decomposition of the class FCS with the added family FCS_RNG:



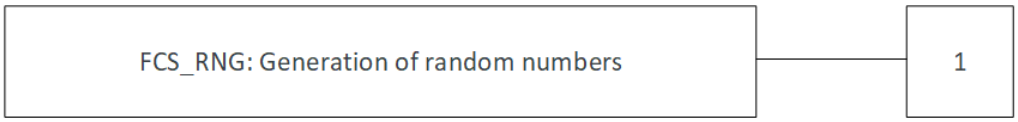
5.1.1 Generation of Random Numbers (FCS_RNG)

This family describes the functional requirements for random number generation used for cryptographic purposes. The description uses the notation as used for the description of SFR families by [CC31R5].

Family behaviour

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

Component levelling



FCS_RNG.1 Generation of random numbers allows the usage of random numbers for performing cryptographic operations e.g. the generation of key material.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

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

6. Security Requirements (ASE_REQ)

This section comprises security functional and security assurance requirements that shall be fulfilled by the TOE.

6.1 Typographical specifications

Operations on the SFRs are identified as follows:

- Iterations are denoted by a slash "/" followed by an iteration identifier
- Assignments performed are printed in **bold** text
- Selections made are indicated in underlined text
- An assignment which is performed as part of a selection is printed in **bold underlined** text
- Refinements are marked in ***bold italic*** text

Footnotes list the original [CC31R5] based text. When only assignments and selections are performed the number referencing a footnote is placed at the performed operation and each footnote shows the single operation. When refinements are performed one footnote list the whole text of the SFR element showing all performed operations.

6.2 Subjects, objects and operations

This section describes subjects, objects and operations supported by the TOE.

Subject	Description
Signer	Natural or legal person who uses the TOE doing server signing / sealing.
Privileged User	User, who performs the administrative functions of the TOE and some Signer related functions.
Privileged User Admin	Privileged User, who only performs installation, configuration and maintenance of the TOE.
Privileged User Technical	Privileged User, who only create Signers.

Table 15: Subjects and their descriptions

Object	Description
R.SIGNER	Represents the user who wants to generate a signature/seal.
R.PRIVILEGED_USER	Represents in the TOE a Privileged User who can manage the TOE and a few processes relevant to R.SIGNER.
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 signing key pair by R.SIGNER.
R.SIGNING_KEY_ID	An identifier that represents the private part of a signing key pair of R.SIGNER.
R.SAD	Data used to activate signature/seal creation under the Signer sole control. Contains R.DTBS/R, R.SIGNING_KEY_ID
R.DTBS/R	Representation of data to be signed/sealed.
R.AUTHORISATION_DATA	Data used by the cryptographic module to activate the

	private part of R.SIGNER's signing key pair.
R.SIGNATURE	The result of a signing process.
R.TSF_DATA	Configuration data of the TOE.
R.PRIVILEGED_USER_ TECHNICAL	Represents in the TOE a Privileged User Technical who can create Signers.
R.REFERENCE_PRIVILEGED_ USER_TECHNICAL_ AUTHENTICATION_DATA	Data used by the TOE to authenticate a Privileged User Technical.

Table 16: Objects and their descriptions

Operation	Description	Subject	Object
Create_New_Privileged_ User	A new Privileged User can be created that includes both the object representing the new Privileged User and the object used to authenticate the newly created Privileged User.	R.PRIVILEGED_USER	R.SIGNER R.REFERENCE_ PRIVILEGED_USER_ AUTHENTICATION_DATA
Create_New_Signer	A new Signer can be created that includes both the object representing the new Signer and the object used to authenticate the newly created Signer.	R.PRIVILEGED_USER R.PRIVILEGED_USER _TECHNICAL	R.SIGNER R.REFERENCE_SIGNER_ AUTHENTICATION_DATA
Signer_Maintenance	A key pair can be deleted by a Signer.	R.PRIVILEGED_USER R.SIGNER	R.SIGNER R.SVD R.SIGNING_KEY_ID
Generate_Signer_Key_Pair	A key pair can be generated by a Signer.	R.PRIVILEGED_USER R.SIGNER	R.SIGNER R.SVD R.SIGNING_KEY_ID
Delete_Signer_Key_Pair	A key pair can be separated from a Signer.	R.PRIVILEGED_USER R.SIGNER	R.SIGNER R.SVD R.SIGNING_KEY_ID
Signing	A Signer can sign/seal data to be signed/sealed and thus generate a signature/seal.	R.SIGNER	R.AUTHORISATION_DATA R.SIGNER R.SIGNING_KEY_ID R.DTBS/R R.SIGNATUR
TOE_Maintenance	The TOE configuration can be managed by a administrator.	R.PRIVILEGED_USER_ ADMIN	R.TSF_DATA

Table 17: Operations and their descriptions

6.3 Security Policies

6.3.1 Access Control Policies (TSP_ACC)

6.3.1.1 Privileged User Creation SFP

The TOE shall control the access to user data according to the following rules:

- Only a securely identified and authenticated Privileged User who
  - provides valid R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA and
  - uses valid Create_New_Privileged_User requests

  will get permission for creating new Privileged User and the security attributes for them.

- Only a securely identified and authenticated Privileged User who
  - provides valid R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA and
  - uses valid Create_New_Privileged_User requests

  will get permission for accessing the security attributes of Privileged User for querying them.

6.3.1.2 Signer Creation SFP

The TOE shall control the access to user data according to the following rules:

- Only a securely identified and authenticated Privileged User or Privileged User Technical who
  - provides valid R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA or R.REFERENCE_PRIVILEGED_USER_TECHNICAL_AUTHENTICATION_DATA and
  - uses valid Create_New_Signer requests

  will get permission for creating new Signer and the security attributes for them.

- Only securely identified and authenticated Privileged User or Signer who
  - provides valid R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA or R.REFERENCE_SIGNER_AUTHENTICATION_DATA and
  - uses valid Create_New_Signer requests

  will get permission for accessing the security attributes of Signer for querying them.

6.3.1.3 Signer Maintenance SFP

The TOE shall control the access to user data according to the following rules:

- Only a securely identified and authenticated Privileged User who
  - provides valid R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA and
  - uses valid Signer_Maintenance requests

  will get permission for maintaining the Signer security attributes R.SVD and R.SIGNING_KEY_ID.

- Only a securely identified and authenticated Signer who
  - provides valid R.REFERENCE_SIGNER_AUTHENTICATION_DATA and
  - uses valid Signer_Maintenance requests

  will get permission for maintaining their own security attributes R.SVD and R.SIGNING_KEY_ID.

6.3.1.4 Signer Key Pair Generation SFP

The TOE shall control the access to user data according to the following rules:

- Only a securely identified and authenticated Privileged User who
  - provides valid R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA and
  - uses valid Generate_Signer_Key_Pair requests

  will get permission for generating a new key pair and the Signer security attributes R.SVD and R.SIGNING_KEY_ID.

- Only securely identified and authenticated Signer who
  - provides valid R.REFERENCE_SIGNER_AUTHENTICATION_DATA and
  - uses valid Generate_Signer_Key_Pair requests

  will get permission for generating a new key pair and the Signer security attributes R.SVD and R.SIGNING_KEY_ID.

6.3.1.5 Signer Key Pair Deletion SFP

The TOE shall control the access to user data according to the following rules:

- Only a securely identified and authenticated Privileged User who
  - provides valid R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA and
  - uses valid Delete_Signer_Key_Pair requests

  will get permission for deleting a key pair and the Signer security attributes R.SVD and R.SIGNING_KEY_ID.

- Only a securely identified and authenticated Signer who
  - provides valid R.REFERENCE_SIGNER_AUTHENTICATION_DATA and
  - uses valid Delete_Signer_Key_Pair requests

  will get permission for deleting a key pair and the Signer security attributes R.SVD and R.SIGNING_KEY_ID.

6.3.1.6 Signing SFP

The TOE shall control the access to user data according to the following rules:

- Only a securely identified and authenticated Signer who
  - provides valid R.SAD and
  - uses valid Signing requests

  will get permission for creating a signature/seal.

6.3.1.7 TOE Maintenance SFP

The TOE shall control the access to TOE data according to the following rules:

- Only a securely identified and authenticated Privileged User Admin who
  - uses valid TOE_Maintenance requests and
  - provides valid R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA

  will get permission for maintaining the TOE configuration data R.TSF_DATA.

6.3.2 Information Flow Control Policies (TSP_IFC)

6.3.2.1 Signer Flow SFP

The TOE shall implement an information flow control policy which follows the following rules:

- The TOE shall be initialized with TOE_Maintenance before performing requests for other operations.
- All rules specified for Signing shall be performed by the TOE.
- The TOE shall not perform any request, if an operation defined by the rules deposited in the TOE cannot be performed successfully.
- The TOE shall only allow a Signer or Privileged User to request for
    - maintaining Signer security attributes
    - the generation of a key pair

  when the Signer is already created in the TOE.
- The TOE shall only allow a Signer and Privileged User to request for the deletion of a signing key pair when the Signer is already created in the TOE and a signing key pair is already created and assigned to the Signer.
- The TOE shall only allow a Signer to request for the creation of a signature/seal when the Signer is already created in the TOE followed by the creation of a key pair for the Signer.
- The TOE shall perform a signing request based on the accessed Signer security attributes.
- The TOE shall return the signature/seal as result of a successful signing request.

6.3.2.2 Privileged User Flow SFP

The TOE shall implement an information flow control policy which follows the following rules:

- The TOE shall be initialized with TOE_Maintenance before performing any request for other operations.
- All rules specified for operations shall be performed by the TOE.
- The TOE shall not perform any request, if an operation defined by the rules deposited in the TOE cannot be performed successfully.
- The TOE shall perform requests for
    - creating Signer
    - creating Privileged User

  by Privileged User based on the accessed Privileged User security attributes.
- The TOE shall perform requests for maintaining the TOE configuration by Privileged User Admin based on the accessed Privileged User Admin security attributes.
- The TOE shall perform requests for
    - creating Signer

  by Privileged User Technical based on the accessed Privileged User Technical security attributes.

6.4 Security Functional Requirements

6.4.1 Security Audit (FAU)

FAU_GEN.1 Audit data generation

Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps

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

- **a)** Start-up and shutdown of the audit functions
- **b)** All auditable events for the not specified [2] 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**
- **j)** **the hash of the DTBS/R(s) and**
- **k)** **R.SIGNATURE**
- **l)** **change of TOE configuration**.[3]

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***, **type of action performed (success or failure), identity of the role which performs the operation**.[4]

FAU_GEN.2 User identity association

Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification

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

[2] [selection: minimum, basic, detailed, not specified]
[3] [assignment: other specifically defined auditable events]
[4] [assignment: other audit relevant information]

6.4.2 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic key generation

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 cryptographic key distribution, or FCS_COP.1 cryptographic operation] FCS_CKM.4 cryptographic key destruction]

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **as shown in the Key Generation Table**[5] and specified cryptographic key sizes **as shown in the Key Generation Table**[6] that meet the following: **standards as shown in the Key Generation Table**[7].

Key Generation Algorithm	Key Sizes	Applicable Standards
RSA PKCS#1 v1.5, RSA PSS	3072 bit to 4096 bit	[RFC8017], [ISO9796]
ECDSA	256 bit to 521 bit	[ISO14888], [FIPS186-4], [ECCBP]

Table 18: Key Generation Table

FCS_CKM.4 Cryptographic key destruction

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **None**[8] that meet the following: **None**[9] .

ST Application Note 3

Key destruction covered by FCS_CKM.4 applies primarily to keys held in the RAM of a cryptographic module (CM).

Furthermore, the TOE uses keys which are used for remote signing/sealing and stored outside the generating CM. Such keys are exported by the generating CM as secured key using the module key of the CM (CM-wrapped key) and are stored as part of a TOE-generated signed container (Wrapped Key) into the [KMIPv20] based Key Manager (see chapter 1.4.1).

Whenever a key is stored outside the generating CM the key is protected in confidentiality and integrity. The integrity of keys is protected by using the Wrapped Key structure which contains a CM-wrapped key and is signed by the TOE before it is stored into the Key Manager.

AES CBC/GCM 128 to 256 Bit is used for the encryption of the CM-wrapped key. The generation of signatures to create Wrapped Keys is done using HMAC-SHA256 with a 256 bit AES key that is derived from the Master Backup Key of the CM.

A Wrapped Key contains the CM-wrapped key, its key ID, and additional metadata specific to remote signing/sealing, such as the ID of the assigned signer.

Keys used for remote signing/sealing are simply destroyed by deleting the according Wrapped Key from the database connected with the Key Manager. Since these items do not contain keys in plaintext they do not require any specific destruction method.

[5] [assignment: cryptographic key generation algorithm]
[6] [assignment: cryptographic key sizes]
[7] [assignment: list of standards]
[8] [assignment: cryptographic key destruction algorithm]
[9] [assignment: list of standards]

FCS_COP.1/Hash Cryptographic operation

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 cryptographic key generation] FCS_CKM.4 cryptographic key destruction
FCS_COP.1.1/ Hash	The TSF shall perform **the computation of hash values**[10] in accordance with a specified cryptographic algorithm **as shown in the Hash Generation Table**[11] and cryptographic key sizes **as shown in the Hash Generation Table**[12] that meet the following: **standards as shown in the Hash Generation Table**[13].

Hash Family	Hash Algorithm	Key Size	Applicable Standards
SHA-2	SHA-256	None	[FIPS180-4]
	SHA-384	None	
	SHA-512	None	
HMAC	HMAC-SHA265	None	[RFC2104], [ISO9797-2]

Table 19: Hash Generation Table

FCS_COP.1/ValSigSea cryptographic operation

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 cryptographic key generation] FCS_CKM.4 cryptographic key destruction
FCS_COP.1.1/ ValSigSea	The TSF shall perform **the verification of electronic signatures/seals**[9] in accordance with a specified cryptographic key generation algorithm **as shown in the Key Generation Table**[10] and specified cryptographic key sizes **as shown in the Key Generation Table**[11] that meet the following: **standards as shown in the Key Generation Table**[12].

[10] [assignment: list of cryptographic operations]
[11] [assignment: cryptographic algorithm]
[12] [assignment: cryptographic key sizes]
[13] [assignment: list of standards]

FCS_RNG.1 Generation of random numbers

Hierarchical to: No other components.
Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a hybrid deterministic[14] random number generator that implements: **RNG class DRG.4 of [AIS 20/31] chapter 4.9**

- **(DRG4.1) The internal state of the RNG shall use {PTRNG of class PTG.2 as random source}.**
- **(DRG4.2) The RNG provides forward secrecy.**
- **(DRG4.3) The RNG provides backward secrecy even if the current internal state is known.**
- **(DRG4.4) The RNG provides enhanced forward secrecy {on condition that 1000 requests for pseudo random bits have been made after last entropy input during instantiation or reseeding}.**
- **(DRG4.5) The internal state of the RNG is seeded by an {PTRNG of class PTG.2}**[15].

FCS_RNG.1.2 The TSF shall provide octets of bits[16] that meet

- **(DRG4.6) The RNG generates output for which {7-10} strings of bit length 128 are mutually different with probability {0.9998}.**
- **(DRG4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A {None}**[17].

[14] [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic]
[15] [assignment: list of security capabilities]

(DRG.4.1) The internal state of the RNG shall [selection: use PTRNG of class PTG.2 as random source, have [assignment: work factor], require [assignment: guess work]].
(DRG.4.4) The RNG provides enhanced forward secrecy [selection: on demand, on condition [assignment: condition], after [assignment: time]].
(DRG.4.5) The internal state of the RNG is seeded by an [selection: internal entropy source, PTRNG of class PTG.2, PTRNG of class PTG.3, [other selection]].

For performed operations of (DRG.4.1/2/3/4) selected/assigned values are positioned within { } .

[16] [selection: bits, octets of bits, numbers [assignment: format of the numbers]]
[17] [assignment: a defined quality metric]

(DRG.4.6) The RNG generates output for which [assignment: number of strings] strings of bit length 128 are mutually different with probability [assignment: probability].
(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A [assignment: additional test suites].

For performed operations of (DRG.4.6 and 4.7) selected/assigned values are positioned within { } .

6.4.3 User Data Protection (FDP)

FDP_ACC.1/Privileged User Creation Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/Privileged User Creation	The TSF shall enforce the **Privileged User Creation SFP**[18] on: **(1) Subjects: Privileged User** **(2) Objects: New security attributes for the Privileged User to be created.** **(3) Operations: Create_New_Privileged_User** **(4) Create_New_Privileged_User: The TOE creates R.PRIVIELEGED_USER and R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA with information transmitted by the Privileged User**[19].

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

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/Privileged User Creation	The TSF shall enforce the **Privileged User Creation SFP**[20] to objects based on the following: **whether the subject is a Privileged User authorised to create a new Privileged User**[21].
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: **only a Privileged User who has been authorised for creation of new users can carry out the Create_New_Privileged_User operation**[22].
FDP_ACF.1.3/Privileged User Creation	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **None**[23].
FDP_ACF.1.4/Privileged User Creation	The TSF shall explicitly deny access of subjects to objects based on the following additional rule: **None**[24].

[18] [assignment: access control SFP]
[19] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]
[20] [assignment: access control SFP]
[21] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]
[22] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects].
[23] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects].
[24] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FDP_ACC.1/Signer Creation Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/Signer Creation	The TSF shall enforce the **Signer Creation SFP**[25] on: **(1) Subjects: Privileged User and Privileged User Technical** **(2) Objects: R.SIGNER and R.REFERENCE_SIGNER_AUTHENTICATION_DATA** **(3) Operations: Create_New_Signer** **(4) Create_New_Signer: The TOE creates R.SIGNER and R.REFERENCE_SIGNER_AUTHENTICATION_DATA with information provided by the Privileged User or Privileged User Technical**[26].

FDP_ACF.1/Signer Creation Security attribute based access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/Signer Creation	The TSF shall enforce the **Signer Creation SFP**[27] to objects based on the following: **whether the subject is a Privileged User or Privileged User Technical authorised to create a new Signer**[28].
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: **only a Privileged User or Privileged User Technical who has been authorised for creation of new users can carry out the Create_New_Signer operation**[29].
FDP_ACF.1.3/Signer Creation	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **None**[30].
FDP_ACF.1.4/Signer Creation	The TSF shall explicitly deny access of subjects to objects based on the following additional rule: **None**[31].

[25] [assignment: access control SFP]
[26] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]
[27] [assignment: access control SFP]
[28] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]
[29] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[30] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]
[31] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FDP_ACC.1/Signer Maintenance Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/Signer Maintenance	The TSF shall enforce the **Signer Maintenance SFP**[32] on: **(1) Subjects: Privileged User and Signer** **(2) Objects: The security attributes R.SIGNING_KEY_ID and R.SVD of R.SIGNER** **(3) Operations: Signer_Maintenance** **(4) Signer_Maintenance: The Privileged User or the Signer instructs the TOE to update R.REFERENCE_SIGNER_AUTHENTICATION_DATA from R.SIGNER**[33].

FDP_ACF.1/Signer Maintenance Security attribute based access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/Signer Maintenance	The TSF shall enforce the **Signer Maintenance SFP**[34] to objects based on the following: **whether the subject is a Privileged User or Signer authorised to maintain the Signer security attributes**[35].
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: **only a Privileged User or Signer who has been authorised to maintain a Signer can carry out the Signer_Maintenance operation**[36].
FDP_ACF.1.3/Signer Maintenance	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **the Signer shall be the owner of the R.SIGNER object to be maintained**[37].
FDP_ACF.1.4/Signer Maintenance	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **if the Signer does not own the R.SIGNER object, it can't be maintained**[38].

[32] [assignment: access control SFP]
[33] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]
[34] [assignment: access control SFP]
[35] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]
[36] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[37] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]
[38] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FDP_ACC.1/Signer Key Pair Generation Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/Signer Key Pair Generation	The TSF shall enforce the **Signer Key Pair Generation SFP**[39] on: **(1) Subjects: Privileged User and Signer.** **(2) Objects: The security attributes R.SVD and R.SIGNING_KEY_ID as part of R.SIGNER.** **(3) Operations: Generate_Signer_Key_Pair** **(4) Generate_Signer_Key_Pair: The Privileged User or the Signer instructs the TOE to request the cryptographic module to generate a pair of signing keys R.SIGNING_KEY_ID and R.SVD and assign them to R.SIGNER**[40].

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

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/Signer Key Pair Generation	The TSF shall enforce the **Signer Key Pair Generation SFP**[41] to objects based on the following: **whether the subject is a Privileged User or Signer authorised to generate a key pair**[42].
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: **only a Privileged User or Signer who has been authorised to generate the key pair can carry out the Generate_Signer_Key_Pair operation**[43].
FDP_ACF.1.3/Signer Key Pair Generation	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **the Signer shall be the owner of the R.SIGNER object where the key pair is to be generated**[44].
FDP_ACF.1.4/Signer Key Pair Generation	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **if the Signer does not own the R.SIGNER object, key pair shall not be generated**[45].

[39] [assignment: access control SFP]
[40] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]
[41] [assignment: access control SFP]
[42] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]
[43] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[44] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]
[45] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FDP_ACC.1/Signer Key Pair Deletion Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/Signer Key Pair Deletion	The TSF shall enforce the **Signer Key Pair Deletion SFP**[46] on: **(1) Subjects: Privileged User and Signer.** **(2) Objects: The security attributes R.SIGNING_KEY_ID and R.SVD as part of R.SIGNER.** **(3) Operations: Delete_Signer_Key_Pair** **(4) Delete_Signer_Krey_Pair: The Privileged User or the Signer instructs the TOE to delete R.SIGNING_KEY_ID and R.SVD from R.SIGNER**[47].

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

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/Signer Key Pair Deletion	The TSF shall enforce the **Signer Key Pair Deletion SFP**[48] to objects based on the following: **whether the subject is a Privileged User or Signer authorised to delete a key pair**[49].
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: **only a Privileged User or Signer who has been authorised to delete the key pair can carry out the Delete_Signer_Key_Pair operation**[50].
FDP_ACF.1.3/Signer Key Pair Deletion	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **the Signer shall be the owner of the R.SIGNER object where the key pair is to be deleted**[51].
FDP_ACF.1.4/Signer Key Pair Deletion	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **if the Signer does not own the R.SIGNER object, key pair shall not be deleted**[52].

[46] [assignment: access control SFP]
[47] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]
[48] [assignment: access control SFP]
[49] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]
[50] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[51] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]
[52] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FDP_ACC.1/Signing Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/Signing	The TSF shall enforce the **Signing SFP**[53] on**:** **(1) Subjects: Signer** **(2) Objects: The security attributes R.SIGNER, R.SIGNING_KEY_ID and R.DTBS/R** **(3) Operations: Signing** **(4) Signing: The Signer instructs the TOE to perform a signing operation with the following steps:** **a. The TOE establishes R.AUTHORISATION_DATA for the R.SIGNING_KEY_ID.** **b. The TOE uses R.AUTHORISATION_DATA and R.SIGNING_KEY_ID to activate a signing key in the cryptographic module and signs the R.DTBS/R and the result is R.SIGNATURE.** **c. The TOE disables the signing key when the signing process is complete**[54].

FDP_ACF.1/Signing Security attribute based access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/Signing	The TSF shall enforce the **Signing SFP**[55] to objects based on the following: **whether the subject is a Signer authorised to create a signature/seal**[56].
FDP_ACF.1.2/Signing	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **(1) The R.SAD is verified in integrity. The R.SAD is verified that it binds together the Signer authentication, a set of R.DTBS/R and R.SIGNING_KEY_ID.** **(2) The R.DTBS/R used for signing operations is bound to the R.SAD.** **(3) The Signer identified in the SAD is authenticated according to the rules specified in FIA_UAU.5/Signer.** **(4) 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/seal**[57].
FDP_ACF.1.3/Signing	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **the Signer shall be the owner of the R.SIGNER object used to generate the signature/seal**[58].
FDP_ACF.1.4/Signing	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **if the Signer does not own the R.SIGNER object, it can't be used to create a signature/seal**[59].

[53] [assignment: access control SFP]
[54] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]
[55] [assignment: access control SFP]
[56] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]
[57] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[58] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]
[59] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FDP_ACC.1/TOE Maintenance Subset access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/ TOE Maintenance	The TSF shall enforce the **TOE Maintenance SFP**[60] on: **(1) Subjects: Privileged User Admin** **(2) Objects: R.TSF_DATA.** **(3) Operations: TOE_Maintenance** **(4) TOE_Maintenance: The administrative user transfers information to the TOE to manage R.TSF_DATA**[61].

FDP_ACF.1/TOE Maintenance Security attribute based access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/TOE Maintenance	The TSF shall enforce the **TOE Maintenance SFP**[62] to objects based on the following: **whether the subject is a Privileged User Admin authorised to maintain the TOE configuration data**[63].
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: **only a Privileged User Admin who has been authorised to maintain the TOE can carry out the TOE_Maintenance operation**[64].
FDP_ACF.1.3/TOE Maintenance	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **None**[65].
FDP_ACF.1.4/TOE Maintenance	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **None**[66].

[60] [assignment: access control SFP]
[61] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]
[62] [assignment: access control SFP]
[63] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]
[64] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[65] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]
[66] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FDP_IFC.1/Signer Subset information flow control

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1/Signer	The TSF shall enforce the **Signer Flow SFP**[67] on **Privileged User and Signer accessing Signer security attributes for all operations**[68].

FDP_IFF.1/Signer Simple security attributes

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
FDP_IFF1.1/Signer	The TSF shall enforce the **Signer Flow SFP**[69] based on the following types of subject and information security attributes: **Privileged User and Signer accessing the Signer security attributes**[70].
FDP_ IFF1.2/Signer	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **(1) The TOE shall be initialized with FDP_ACC.1/TOE Maintenance.** **(2) To allow a Signer to sign/seal, the Signer shall be created in the TOE by FDP_ACC.1/Signer Creation followed by FDP_ACC.1/Signer Key Pair Generation.** **(3) After Signer is created the following operations can be done: FDP_ACC.1/Signer Key Pair Generation, FDP_ACC.1/Signer Maintenance and FDP_ACC.1/Signing**[71].
FDP_ IFF1.3/Signer	The TSF shall enforce the: **None**[72].
FDP_ IFF1.4/Signer	The TSF shall explicitly authorize an information flow based on the following rules: **None**[73].
FDP_ IFF1.5/Signer	The TSF shall explicitly deny an information flow based on the following rules: **None**[74].

[67] [assignment: information flow control SFP]
[68] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]
[69] [assignment: information flow control SFP]
[70] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]
[71] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]
[72] [assignment: additional information flow control SFP rules]
[73] [assignment: rules, based on security attributes, that explicitly authorise information flows]
[74] [assignment: rules, based on security attributes, that explicitly deny information flows]

FDP_IFC.1/Privileged User Subset information flow control

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1/ Privileged User	The TSF shall enforce the **Privileged User Flow SFP**[75] on **Privileged User accessing Privileged User security attributes for all operations**[76].

FDP_IFF.1/Privileged User Simple security attributes

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
FDP_IFF1.1/ Privileged User	The TSF shall enforce the **Privileged User Flow SFP**[77] based on the following types of subject and information security attributes: **Privileged User accessing the Privileged User security attributes**[78].
FDP_ IFF1.2/ Privileged User	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **The TOE shall be initialized with FDP_ACC.1/TOE Maintenance**[79].
FDP_ IFF1.3/ Privileged User	The TSF shall enforce the: **None**[80].
FDP_ IFF1.4/ Privileged User	The TSF shall explicitly authorize an information flow based on the following rules: **None**[81].
FDP_ IFF1.5/ Privileged User	The TSF shall explicitly deny an information flow based on the following rules: **None**[82].

[75] [assignment: information flow control SFP]
[76] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]
[77] [assignment: information flow control SFP]
[78] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]
[79] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]
[80] [assignment: additional information flow control SFP rules]
[81] [assignment: rules, based on security attributes, that explicitly authorise information flows]
[82] [assignment: rules, based on security attributes, that explicitly deny information flows]

FDP_UCT.1 Basic data exchange confidentiality

Hierarchical to:	No other components.
Dependencies:	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_UCT.1.1	The TSF shall enforce the **Privileged User SFP, Signer Creation SFP, Signer Key Pair Generation SFP, Signer Key Pair Deletion SFP, Signer Maintenance SFP, Signing SFP, Signer Flow SFP and Privileged User Flow SFP**[83] to transmit and receive[84] user data in a manner protected from unauthorised disclosure.

FDP_UIT.1/SecAttUsr Data exchange integrity

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted Path
FDP_UIT.1.1/ SecAttUsr	The TSF shall enforce the **access control and information flow control as defined in FDP_IFC.1/Signer and FDP_IFC.1/Privileged User** to transmit and receive user data in a manner protected from modification and insertion ***for all security attributes for R.SIGNER and R.PRIVILEGED_USER***.[85]
FDP_UIT.1.2/ SecAttUsr	The TSF shall be able to determine on receipt of user data, whether modification, deletion and insertion ***for all security attribute as defined in R.SIGNER and R.PRIVILEGED_USER*** has occurred.[86]

FDP_UIT.1/SAD Data exchange integrity

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted Path
FDP_UIT.1.1/SAD	The TSF shall enforce the **access control and information flow control as defined in FDP_IFC.1/Signer and FDP_IFC.1/Privileged User** to transmit and receive user data in a manner protected from modification and replay ***for R.SAD***.[85]
FDP_UIT.1.2/SAD	The TSF shall be able to determine on receipt of user data, whether modification and replay ***for R.SAD*** has occurred.[86]

[83] [assignment: access control SFP(s) and/or information flow control SFP(s)]
[84] [selection: transmit, receive]
[85] The TSF shall enforce the [assignment: access control SFP(s) and/or information flow control SFP(s)] to [selection: transmit, receive] user data in a manner protected from [selection: modification, deletion, insertion, replay] errors.
[86] The TSF shall be able to determine on receipt of user data, whether [selection: modification, deletion, insertion, replay] has occurred.

6.4.4 Identification and Authentication (FIA)

FIA_AFL.1 Authentication failure handling

Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication
FIA_AFL.1.1	The TSF shall detect when **3**[87] unsuccessful authentication attempts occur related to the **Privileged User and Signer and Privileged User Admin and Privileged User Technical authentication**[88].
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been met[89], the TSF shall **suspend the Privileged User and the Signer and Privileged User Admin and Privileged User Technical**[90].

ST Application Note 4

Suspending the Signer means that the Signer's R.SIGNING_KEY_IDs are also suspended and cannot be used for server signing for the time the Signer is suspended.

FIA_ATD.1 User attribute definition

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users: **the security attribute as defined in FIA_USB.1**[91].

FIA_UAU.1 Timing of authentication

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FIA_UAU.1.1	The TSF shall allow **requests for Create_New_Signer, Create_New_Privileged_User, Signer_Maintenance, Generate_Signer_Key_Pair, Delete_Signer_Key_Pair, Signing**[92] 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.

[87] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[88] [assignment: list of authentication events]
[89] [selection: met, surpassed]
[90] [assignment: list of actions]
[91] [assignment: list of security attributes]
[92] [assignment: list of TSF mediated actions]

FIA_UAU.5/Signer Multiple authentication mechanisms

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.5.1/ Signer	The TSF shall provide **as authentication mechanism indirectly by the TOE: Username/Password, Hardware Token, eID Card or any other authentication mechanism according to prEN 419241-1:2017 SRA_SAP.1.1** to support ***Signer*** authentication.[93]
FIA_UAU.5.2/ Signer	The TSF shall authenticate any ***Signer's*** claimed identity according to the ***following rules***: **(1) A Signer always authenticates itself by means of an ID Token.** **(2) To get an ID Token the Signer authenticates itself against an Identity Provider using one of the authentication mechanisms listed in FIA_UAU.5.1/Signer.** **(3) An ID Token only is to be generated by an Identity Provider as the result of a successful performed authentication of a Signer.** **(4) The Signer gives in an ID Token to initiate authentication.** **(5) Authentication is performed by validating the signature of the given ID Token and checking the assertions contained with regard to role permissions.** **(6) Only when the given ID Token is validated and checked successfully and the TOE trusts the Identity Provider the claimed identity is authenticated successfully and the Signer gets access to the relevant R.SIGNER object as the owner**[94].

FIA_UAU.5/Privileged User Multiple authentication mechanisms

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.5.1/ Privileged User	The TSF shall provide **as authentication mechanism indirectly by the TOE: Username/Password, Hardware Token, eID Card or any other authentication mechanism according to prEN 419241-1:2017 SRA_SAP.1.1** to support ***Privileged User*** authentication.[93]
FIA_UAU.5.2/ Privileged User	The TSF shall authenticate any ***Privileged User's*** claimed identity according to the ***following rules***: **(1) A Privileged User always authenticates itself by means of an ID Token.** **(2) To get an ID Token the Privileged User authenticates itself against an Identity Provider using one of the authentication mechanisms listed in FIA_UAU.5.1/Privileged User.** **(3) An ID Token only is to be generated by an Identity Provider as the result of a successful performed authentication of a Privileged User.** **(4) The Privileged User gives in an ID Token to initiate authentication.** **(5) Authentication is performed by validating the signature of the given ID Token and checking the assertions contained with regard to role permissions.** **(6) Only when the given ID Token is validated and checked successfully and the TOE trusts the Identity Provider the claimed identity is authenticated successfully and the Privileged User gets access to the relevant R.PRIVILEGED_USER object as the owner**[94].

[93] The TSF shall provide [assignment: list of multiple authentication mechanisms] to support user authentication.
[94] The TSF shall authenticate any user's claimed identity according to the [assignment: rules describing how the multiple authentication mechanisms provide authentication].

FIA_UAU.5/Privileged User Admin Multiple authentication mechanisms

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.5.1/ Privileged User Admin	The TSF shall provide **as authentication mechanism directly by the operation system of the TOE: Username/Password** to support ***Privileged User Admin*** authentication.[93]
FIA_UAU.5.2/ Privileged User Admin	The TSF shall authenticate any ***Privileged User Admin's*** claimed identity according to the ***following rules***: **(1) Privileged User Admin always authenticates itself by using the Username/Password mechanism.** **(2) The Privileged User Admin gives in a Username/Password combination to initiate authentication.** **(3) Authentication is performed by validating the given Username/Password combination against these kept by the operation system of the TOE.** **(4) Only when the given Username/Password combination is known by the operation system of the TOE the claimed identity is authenticated successfully and the Privileged User Admin gets access to the relevant R.PRIVILEGED_USER_ADMIN object as the owner**[94].

FIA_UAU.5/Privileged User Technical Multiple authentication mechanisms

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.5.1/ Privileged User Technical	The TSF shall provide **as authentication mechanism directly by the TOE: X.509 Certificate** to support ***Privileged User Technical*** authentication.[93]
FIA_UAU.5.2/ Privileged User Technical	The TSF shall authenticate any ***Privileged User Technical's*** claimed identity according to the ***following rules***: **(1) Privileged User Technical always authenticates itself by using the X.509 Certificate mechanism.** **(2) The Privileged User Technical gives in a request which data is signed using a X.509 certificate to initiate authentication.** **(3) Authentication is performed by validating the signature of the given request using the corresponding X.509 certificate kept by the TOE.** **(4) Only when the corresponding X.509 certificate is kept by the TOE and the signature is validated by the TOE the claimed identity is authenticated successfully and the Privileged User Technical gets access to the relevant R.PRIVILEGED_USER_TECHNICAL object as the owner**[94].

ST Application Note 5

In the case of X.509 Certificate mechanism, the authentication of the Privileged User Technical is done by means of a signature on the data of his requests. The signatures are generated using the private key of the Privileged User Technical. The public key assigned to the private key of the Privileged User Technical is stored as X.509 certificate in the TOE configuration.

FIA_UID.2 User identification before any action

Hierarchical to:	FIA_UID.1 Timing of identification
Dependencies:	No dependencies.
FIA_UID.2.1	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_USB.1 User-subject binding

Hierarchical to:	No other components.
Dependencies:	FIA_ATD.1 User attribute definition
FIA_USB.1.1	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **(1) R.REFERENCE_SIGNER_AUTHENTICATION_DATA, R.SIGNING_KEY_ID, R.SVD, R.SIGNER to Signer and** **(2) R.REFERENCE_PRIVILEGED_USER_AUTHENTICATION_DATA, R.PRIVILEGED_USER to Privileged User and** **(3) R.REFERENCE_PRIVILEGED_USER_ADMIN_AUTHENTICATION_DATA, R.PRIVILEGED_USER_ADMIN to Privileged User Admin and** **(4) R.REFERENCE_PRIVILEGED_USER_TECHNICAL_AUTHENTICATION_DATA, R.PRIVILEGED_USER_TECHNICAL to Privileged User Technical**[95].
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: **whether the subject is a Privileged User or Privileged User Technical authorised to create a new Signer. Whether the subject is a Privileged User authorised to create a new Privileged User**[96].
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: **whether the subject is a Privileged User authorised to modify an R.SIGNER object. Whether the subject is a Signer authorised to modify his own R.SIGNER object**[97].

[95] [assignment: list of user security attributes]
[96] [assignment: rules for the initial association of attributes].
[97] [assignment: rules for the changing of attributes].

6.4.5 Security Management (FMT)

FMT_MSA.1/Signer Management of security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/ Signer	The TSF shall enforce the **(1) Signer Creation SFP**[98] to restrict the ability to **create**[99] the security attributes **listed in FIA_USB.1 for Signer**[100] to **authorised Privileged User and Privileged User Technical**[101]. **(2) Signer Key Pair Generation SFP**[98] to restrict the ability to **generate**[99] the security attributes **R.SVD and R.SIGNING_KEY_ID**[100] to **authorised Privileged User and Signer**[101]. **(3) Signer Key Pair Deletion SFP**[98] to restrict the ability to delete[99] the security attributes **R.SVD and R.SIGNING_KEY_ID**[100] to **authorised Privileged User and Signer**[101]. **(4) Signing SFP**[98] to restrict the ability to **create**[99] the security attributes **R.DTBS/R as part of R.SIGNER**[100] to **authorised Signer**[101]. **(5) Signing SFP**[98] to restrict the ability to query[99] the security attributes **as listed in FIA_USB.1**[100] to **authorised Signer**[101]. **(6) Signer Maintenance SFP**[98] to restrict the ability to **destruct**[99] the security attributes **R.SVD and R.SIGNING_KEY_ID as part of R.SIGNER**[100] to **authorised Privileged User and Signer**[101].

FMT_MSA.1/Privileged User Management of security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/ Privileged User	The TSF shall enforce the **Privileged User Creation SFP**[98] to restrict the ability to query **and create**[99] the security attributes **listed in FIA_USB.1 for Privileged User**[100] to **authorised Privileged User**[101].

FMT_MSA.2 Secure security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for **all security attributes listed in FIA_USB.1**[102].

[98] [assignment: access control SFP(s), information flow control SFP(s)]
[99] [selection: change_default, query, modify, delete, [assignment: other operations]]
[100] [assignment: list of security attributes]
[101] [assignment: the authorised identified roles]
[102] [assignment: list of security attributes]

FMT_MSA.3/Signer Static attributes initialisation

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1/ Signer	The TSF shall enforce the **Signer Creation SFP**[103] to provide restrictive[104] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/ Signer	The TSF shall allow the **Privileged User or Privileged User Technical**[105] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3/Privileged User Static attributes initialisation

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1/ Privileged User	The TSF shall enforce the **Privileged User Creation SFP**[103] to provide restrictive[104] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/ Privileged User	The TSF shall allow the **Privileged User**[105] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1 Management of TSF data

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1	The TSF shall restrict the ability to modify[106] the **R.TSF_DATA**[107] to **Privileged User Admin**[108].

FMT_SMF.1 Specification of management functions

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: **(1) Signer management,** **(2) Privileged User management and** **(3) Configuration management**[109]

[103] [assignment: access control SFP, information flow control SFP]
[104] [selection, choose one of: restrictive, permissive, [assignment: other property]]
[105] [assignment: the authorised identified roles]
[106] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[107] [assignment: list of TSF data]
[108] [assignment: the authorised identified roles]
[109] [assignment: list of management functions to be provided by the TSF]

FMT_SMR.2 Restrictions on security roles

Hierarchical to:	FMT_SMR.1 Security roles
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.2.1	The TSF shall maintain the roles: **Signer and Privileged User and Privileged User Admin**[110].
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 **(1) Signer can't be a Privileged User and** **(2) Signer can't be a Privileged User Admin**[111] are satisfied.

6.4.6 Protection of the TSF (FPT)

FPT_PHP.1 Passive detection of physical attack

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_PHP.1.1	The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.
FPT_PHP.1.2	The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

ST Application Note 6

The TOE is a software solution. It is operated in a tamper resistant environment. The tamper resistant environment is provided by the TOE environment. In particular, the requirements mentioned under OE.ENV related to physical tampering are implemented by the TOE environment. Specifically, the TOE is operated in a separately secured network zone of a qualified trusted service provider (TSP) that meets the requirements of [EN419241-2] "7.3 Security objectives for the operating environment".

FPT_RPL.1 Replay detection

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_RPL.1.1	The TSF shall detect replay for the following entities: **R.SAD**.[112]
FPT_RPL.1.2	The TSF shall perform **the rejection of the signing operation**[113] when replay is detected.

[110] [assignment: authorised identified roles].
[111] [assignment: conditions for the different roles]
[112] [assignment: list of identified entities]
[113] [assignment: list of specific actions]

FPT_STM.1 Reliable time stamps

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

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret **(1) R.SIGNER** **(2) R.REFERENCE_SIGNER_AUTHENTICATION_DATA** **(3) R.SAD, R.DTBS/R and** **(4) R.SVD, R.PRIVILEGED_USER, R.REFERENCE_PRIVILEGED_USER _AUTHENTICATION_DATA**[114] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use **data integrity either on data or on communication channel**[115] when interpreting the TSF data from another trusted IT product.

6.4.7 Trusted Paths/Channels (FTP)

FTP_TRP.1/SSA Trusted Path

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1.1/SSA	The TSF shall provide a communication path between itself and ***Privileged User through SSA*** users that is logically distinct from other communication paths and provides ensured identification of its end points and protection of the communicated data from modification.[116]
FTP_TRP.1.2/SSA	The TSF shall permit ***Privileged User through SSA*** to initiate communication via the trusted path.[117]
FTP_TRP.1.3/SSA	The TSF shall require the use of the trusted path for **FDP_ACC.1.1/Privileged User Creation, FDP_ACC.1/Signer Creation, FDP_ACC.1/Signer Maintenance, FDP_ACC.1/Signer Key Pair Generation, FDP_ACC.1/Signer Key Pair Deletion**[118].

[114] [assignment: list of TSF data types]
[115] [assignment: list of interpretation rules to be applied by the TSF]
[116] The TSF shall provide a communication path between itself and [selection: remote, local] 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, disclosure, [assignment: other types of integrity or confidentiality violation]].
[117] The TSF shall permit [selection: the TSF, local users, remote users] to initiate communication via the trusted path.
[118] [selection: initial user authentication, [assignment: other services for which trusted path is required]]

FTP_TRP.1/SIC Trusted Path

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1.1/SIC	The TSF shall provide a communication path between itself and ***Remote Signer through SIC*** users that is logically distinct from other communication paths and provides ensured identification of its end points and protection of the communicated data from modification.[119]
FTP_TRP.1.2/SIC	The TSF shall permit ***Remote Signer through SIC*** to initiate communication via the trusted path.[120]
FTP_TRP.1.3/SIC	The TSF shall require the use of the trusted path for **FDP_ACC.1/Signing**[121].

FTP_TRP.1/RSSA Trusted Path

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1.1/RSSA	The TSF shall provide a communication path between itself and ***Remote Signer through SSA*** users that is logically distinct from other communication paths and provides ensured identification of its end points and protection of the communicated data from modification.[119]
FTP_TRP.1.2/ RSSA	The TSF shall permit **Remote Signer through SSA** to initiate communication via the trusted path. [120]
FTP_TRP.1.3/ RSSA	The TSF shall require the use of the trusted path for **FDP_ACC.1/Signer Maintenance, FDP_ACC.1/Signer Key Pair Generation, FDP_ACC.1/Signer Key Pair Deletion**[121].

FTP_TRP.1/Admin Trusted Path

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1.1/Admin	The TSF shall provide a communication path between itself and ***Privileged User Admin*** users that is logically distinct from other communication paths and provides ensured identification of its end points and protection of the communicated data from modification.[119]
FTP_TRP.1.2/Admin	The TSF shall permit ***Privileged User Admin*** to initiate communication via the trusted path. [120]
FTP_TRP.1.3/Admin	The TSF shall require the use of the trusted path for **FDP_ACC.1/TOE Maintenance**[121].

ST Application Note 7

The operating system of the server where the TOE is installed is configured in such a way that remote access to the server is only possible for Privileged User Admins after a 2-factor authentication via an SSL-protected connection and only from the internal network, where the server is placed.

[119] The TSF shall provide a communication path between itself and [selection: remote, local] 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, disclosure, [assignment: other types of integrity or confidentiality violation]].

[120] The TSF shall permit [selection: the TSF, local users, remote users] to initiate communication via the trusted path.

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

FTP_TRP.1/Technical Trusted Path

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1.1/Technical	The TSF shall provide a communication path between itself and ***Privileged User Technical*** users that is logically distinct from other communication paths and provides ensured identification of its end points and protection of the communicated data from modification.[122]
FTP_TRP.1.2/Technical	The TSF shall permit ***Privileged User Technical*** to initiate communication via the trusted path.[123]
FTP_TRP.1.3/Technical	The TSF shall require the use of the trusted path for **FDP_ACC.1/Signer Creation**[124].

FTP_ITC.1/CM Inter-TSF trusted channel

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1.1/CM	The TSF shall provide a communication channel between itself and ***a cryptographic module certified according to [EN419221-5]*** that is logically distinct from other communication channels and provides ***ensured authentication*** of its end points and protection of the ***communicated data*** from modification or disclosure.[125]
FTP_ITC.1.2/CM	The TSF shall permit ***the TSF and a cryptographic module certified according to [EN419221-5]*** to initiate communication via the trusted channel.[126]
FTP_ITC.1.3/CM	The TSF shall initiate communication via the trusted channel for **the operations Generate_Signer_Key_Pair, Signing**[127].

[122] The TSF shall provide a communication path between itself and [selection: remote, local] 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, disclosure, [assignment: other types of integrity or confidentiality violation]].
[123] The TSF shall permit [selection: the TSF, local users, remote users] to initiate communication via the trusted path.
[124] [selection: initial user authentication, [assignment: other services for which trusted path is required]]
[125] 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.
[126] The TSF shall permit [selection: the TSF, another trusted IT product] to initiate communication via the trusted channel.
[127] [assignment: list of functions for which a trusted channel is required]

6.5 Security Assurance Requirements

The following Table gives an overview on the security assurance requirements that have to be fulfilled by the TOE. They correspond to the Evaluation Assurance Level (EAL) 1, augmented by ADV_FSP.2 and ADV_TDS.1.

Assurance Class	Assurance Components
Development (ADV)	Security-enforcing functional specification (ADV_FSP.2)
	Basic Design (ADV_TDS.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life-cycle support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Security target evaluation (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	TOE summary specification (ASE_TSS.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

Table 20: TOE security assurance requirements

6.6 SFR Dependencies

SFR	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/Hash	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	The hash algorithms as defined in FCS_COP.1/Hash do not need any key material. As such the dependency to the generation or destruction of key material is omitted for this SFR.
FCS_COP.1/ValSigSea	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1, FCS_COP.1/ValSigSea refers to a cryptographic operation which uses public keys. Public keys do not need to be destructed.
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/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/Privileged User
FDP_ACF.1/Signer Maintenance	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signer Maintenance FMT_MSA.3/Signer FMT_MSA.3/Privileged User
FDP_ACF.1/Signer Key Pair Generation	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signer Key Pair Generation FMT_MSA.3/Signer FMT_MSA.3/Privileged User
FDP_ACF.1/Signer Key Pair Deletion	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Signer Key Pair Deletion FMT_MSA.3/Signer FMT_MSA.3/Privileged User

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_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_UCT.1	[FTP_ITC.1 or FTP_TRP.1] [FDP_ACC.1 or FDP_IFC.1]	FTP_TRP.1 FDP_IFC.1/Signer FDP_IFC.1/Privileged User
FDP_UIT.1/SecAttUsr	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	FDP_IFC.1/Signer, FTP_TRP.1, FDP_IFC.1/Privileged User
FDP_UIT.1/SAD	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	FDP_IFC.1/Signer, FTP_TRP.1, FDP_IFC.1/Privileged User
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1	None	No dependents
FIA_UAU.1	FIA_UID.1	FIA_UID.2
FIA_UAU.5/Signer	None	No dependents
FIA_UAU.5/Privileged User	None	No dependents
FIA_UAU.5/Privileged User Admin	None	No dependents
FIA_UAU.5/Privileged User Technical	None	No dependents
FIA_UID.2	None	No dependents
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	No dependents
FMT_SMR.2	FIA_UID.1	FIA_UID.2
FPT_PHP.1	None	No dependents
FPT_RPL.1	None	No dependents
FPT_STM.1	None	No dependents
FPT_TDC.1	None	No dependents
FTP_TRP.1/SSA	None	No dependents
FTP_TRP.1/SIC	None	No dependents
FTP_TRP.1/RSSA	None	No dependents
FTP_TRP.1/Admin	None	No dependents
FTP_TRP.1/Technical	None	No dependents
FTP_ITC.1/CM	None	No dependents

Table 21: Rationale for SFR Dependencies

7 TOE Summary Specification (ASE_TSS)

7.1 SF1 – Security Audit

The TOE produces audit logs for security relevant events as a reliable supporting evidence of operations. The TOE supports audit logging of the following events (FAU_GEN.1):

- Start-up and shutdown of the audit functions
- All auditable events for the not specified level of audit
- Privileged User management
- Privileged User authentication
- Signer management
- Signer authentication
- Signing key generation
- Signing key destruction
- Signing key activation and usage including
- the hash of the DTBS/R(s)
- R.SIGNATURE
- change of TOE configuration.

The audit functionality cannot be stopped during the operation of the TOE. Therefor the start of the audit functionality is implicitly logged by logging the start-up of the TOE. The shutdown of the TOE is logged by the application server as part of the operational environment of the TOE.

Whenever a security-relevant event occurs during execution of an operation, a new log entry is produced containing the following information (FAU_GEN.1 and FAU_GEN.2):

- Date and time of the event: system time synchronized with the reliable time source
- Type of event: description of the event
- Subject identity: identification of the TOE user that originated the event
- Result: description of the result type (success or failure).

The TOE writes produced audit logs to the SSA which is connected to the Audit Manager (AM) that manages audit log in a secured way. Security audit logs are protected by the AM from modification and deletion. There is no operation for audit records other than exporting them to authorised administrators. The audit logs are written to a database. The protection of the audit logs is achieved by generating a separate digital signature for every row stored in the database where audit logs are stored. The signature is verified when reading and using any signed data upon the request of an authorised administrator.

Please note that the records of the events shutdown of the TOE are stored within the file system of the server where the TOE is installed. These servers are assumed to be physically protected from unauthorised access. Thereby the protection of these audit logs from unauthorised modifications and deletion is provided by the operational environment of the TOE.

The audit logs can be exported to a file upon the request on an authorised administrator. The administrator can use this file to analyze the audit records of interest.

7.2 SF2 – Cryptographic Support

7.2.1 Key Generation and Destruction

The TOE supports the generation of key material using the following algorithms (see [SOGISACM], [TS119312]):

- RSA PKCS#1 v1.5 with 3072/4096 Bit Key Length
- RSA PSS with 3072/4096 Bit Key Length
- ECDSA with 256/384/512 Bit Key Length using Brainpool Curves

using the random number generation function of a cryptographic module (FCS_RNG.1).

The TOE does not support a specific destruction method as keys which are generated by a cryptographic module are only stored outside this module in an encrypted container while the container is encrypted with the functions and module key of the cryptographic module. Keys used for remote signature/seal are simply destroyed by deleting the according encrypted container from the database where it is stored according to an implementation based on [KMIPv20] (FCS_CKM.4).,

7.2.2 Signature/Seal Creation

For the signing process the TOE uses the attached cryptographic module certified according to [EN419221-5] and reliable time stamps (FPT_STM.1).

To launch the signing process the following information has to be provided via the Signer through SIC to the TOE:

- the Signer's authentication data (as specified in [EN419241-1])
- a unique reference to the key that shall be used for signature/seal creation (R.SIGNING_KEY_ID)
- a DTBS/R of the data to be signed (including the identifier of the hash algorithm that shall be used)

The Signer is authenticated indirectly by the TOE validating ID Token and contained assertions supported by the request. ID Token and contained assertions are the result of a successful authentication of the Signer against an Identity Provider using one of the possible authentication mechanisms (FIA_UAU.5).

To provide a unique reference to the key that shall be used for server signing the Signer has to select a signing key along with the corresponding singing certificate gained by the enrolment process using modules of the environment of the TOE.

ST Application Note 8

PKCS11 based key identifiers are used to represent R.SIGNING_KEY_ID.
An example of a possible PKCS11 property is CKA_ID.[128]

The hash algorithm used to get DTBS/R shall correspond to SHA-256, SHA-384 or SHA-512 (FCS_COP.1/Hash).

The SIC generates a hash value of the selected data to be signed/sealed using the chosen algorithm. At this time the Signer is informed about the pending start of the signing process by a notification. In addition, the following information will be displayed:

- Name of the document to be signed,
- SHA-512 hash value of the document to be signed

To launch the signing process the Signer has to confirm the start of the signing process.

To prevent the re-use of SAD for the creation of remote signatures/seals, the TOE detects already used SAD and rejects signing operations requested with already used SAD (FPT_RPL.1).

[128] the exactly PKCS11 based key identifier used by the TOE is to be specified

In case of (SADGEN1) once the user has confirmed the start of the signing process, the SIC generates R.SAD and transmits R.SAD to the SSA. The SSA checks the request for signing. If the request for signing is correct, the SSA transmits R.SAD to the TOE requesting signing of R.SAD by the cryptographic module.
In the case (SADGEN2) the SIC generates the SAD signing key and requests the remote SAK for generating the SAD. The remote SAK transmits the generated SAD back to the SIC. Then the SIC signs the SAD resulting in R.SAD and transmits R.SAD to the SAK which forwards it to the SSA.

If the signing of R.SAD is done successfully the SSA requests signing at the TOE using one of the following algorithms (FCS_CKM.1) according to according to [SOGISACM], [TS119312]:

- RSA PKCS#1 v1.5 with 3072/4096 Bit Key Length
- RSA PSS with 3072/4096 Bit Key Length
- ECDSA with 256/384/512 Bit Key Length using Brainpool Curves

The TOE checks the request for signing whether the Signer is authenticated (FIA_UAU.5/Signer) then requests R.AUTHORISATION_DATA corresponding to the supplied R.SIGNING_KEY_ID an checks whether R.SIGNER of the supplied R.SAD matches to R.SIGNER contained in the R.SIGNING_KEY_ID specific R.AUTHORISATION_DATA.

If the checks were successful the attached cryptographic module certified according to [EN419221-5] then signs the DTBS/R and gives back R.SIGNATURE.

7.2.3 Signature Verification

To verify an electronic signature/seal the TOE performs the following action:

- mathematical verification of the electronic signature/seal (FCS_COP.1/ValSigSea).

To validate a signature/seal mathematically, the TOE first performs a mathematical operation to calculate the hash value from the signature/seal which is the result of the server signing. Therefor the TOE uses the cryptographic algorithm and the public key of the given signing certificate. Afterwards the TOE calculates the hash value of the original data which was server signed/sealed. For this operation the TOE uses the algorithm that was specified within the signature/seal. In the following the TOE checks whether both hash values are identical. If the hash values differ, an error message is returned. Otherwise, the signature/seal verification is performed successfully.

Permitted hash algorithms are (FCS_COP.1/Hash):

- SHA-256,
- SHA-384,
- SHA-512.

7.3 SF3 – Access Control

The TOE is able to manage Signer, Privileged User, Privileged User Technical, their security attributes as also as its own configuration (FIA_ATD.1, FIA_USB.1, FMT_MSA.2, FMT_SMF.1, FMT_SMR.2). Only the Privileged User Admin is permitted to modify configuration data (FMT_MTD.1).

The following operations can be performed before (i.e. without) user identification (FIA_UAU.1.1):

- Requesting Create_New_Signer
- Requesting Signer_Maintenance
- Requesting Generate_Signer_Key_Pair
- Requesting Delete_Signer_Key_Pair
- Requesting Signing
- Requesting Create_New_Privileged_User

Any other Operations the TOE provides can only be performed after successful identification and authentication of the Signer or Privileged User (FIA_UAU.1, FIA_UID.2). Transmitting and receiving user data is performed in a manner protected from unauthorised disclosure (FDP_UCT.1). The TOE is able to associate users with roles (FMT_SMR.2).

The TOE authenticates the identity of a Signer or Privileged User indirectly by validating the signature of the given ID Token supplied by the request. In addition, the TOE checks the assertions contained with regard to role permissions. To get an ID Token the Signer or Privileged User always authenticate against an Identity Provider using a suitable authentication mechanism (Username/Password, Hardware Token , eID Card or any other authentication mechanism according to prEN 419241-1:2017 SRA_SAP.1.1). An ID Token is only be generated by an Identity Provider as the result of a successful performed authentication of a Signer or Privileged User. Only when the signature of the given ID Token is validated and the assertions are checked successfully and the TOE trusts the Identity Provider the claimed identity is authenticated successfully. (FIA_UAU.5.1/Signer and FIA_UAU.5.1/Privileged User).

In difference to the Signer and Privileged User the Privileged User Admin its identity is authenticated directly by TOE by using Username/Password authentication mechanism. Only when the given Username/Password combination is known by the operating system of the TOE the claimed identity is authenticated successfully (FIA_UAU.5.1/Privileged User Admin).

The Privileged User Technical its identity is authenticated directly by TOE by using X.509 Certificate authentication mechanism. Only when the given X.509 is known and validated the claimed identity is authenticated successfully (FIA_UAU.5.1/Privileged User Technical).

The TSF detects when 3 unsuccessful authentication attempts occur suspends the requesting user whether this is a Privileged User, a Signer or a Privileged User Admin (FIA_AFL.1). The TOE ensures that a Signer can’t be a Privileged User or a Privileged User Admin (FMT_SMR.2).

ST Application Note 9

Appendix A – Authentication describes the mentioned means of identification Username/Password, Hardware Token, eID Card or any other authentication mechanism according to prEN 419241-1:2017 SRA_SAP.1.1 that can be used for authentication more in detail.

The users are then associated to the relevant object which uniquely identifies them and their role within the TOE in order to acquire privileges. The TOE defines the roles Signer and Privileged User and Privileged User Admin and Privileged User Technical (FMT_SMR.2).

Only users assigned to the role Signer are allowed to use the following operations:

- Signing

Only users assigned to the role Privileged User are allowed to use the following operation:

- Create_New_Privileged_User

Only users assigned to the role Privileged User or Privileged User Technical are allowed to use the following operation:

- Create_New_Signer

Further, only users assigned to the role Signer or Privileged User are allowed to use the following operations:

- Signer_Maintenance
- Generate_Signer_Key_Pair
- Delete_Signer_Key_Pair

Additionally, only users assigned to the role Privileged User Admin are allowed to use the following operations:

- TOE_Maintenance

When a controlled resource is accessed the TOE verifies that the caller meets the required access rules for the resource and grants or denies access (FDP_ACF.1/*, FDP_ACC.1/*).

Only authorised Privileged User will get permission for:

- creating new Privileged User and the security attributes for them,
- accessing the security attributes of Signer or Privileged User for querying them.

Only authorised Privileged User or Privileged User Technical will get permission for:

- creating new Signer and the security attributes for them.

Only authorised Signer will get permission for:

- creating a signature/seal.

Further, only users assigned to the role Signer or Privileged User will get permission for:

- maintaining the (own) Signer security attributes R.SVD and R.SIGNING_KEY_ID,
- generating a new key pair and the Signer security attributes R.SVD and R.SIGNING_KEY_ID,
- deleting a key pair and the Signer security attributes R.SVD and R.SIGNING_KEY_ID.

Additionally, only users assigned to the role Privileged User Admin will get permission for:

- maintaining the TOE configuration data R.TSF_DATA.

If the subject does not have sufficient rights to perform the operation on the object, the TOE denies access and generates an error. If no access rules are defined for a resource, the access is denied (FMT_MSA.1/*, FMT_MSA.3/*).

7.4 SF4 – Information Flow Control

The TOE implements an information flow control for the subjects Signer and Privileged User and Privileged User Admin, Privileged User Technical and assigned operations while performing requests (FDP_IFC.1/Signer and FDP_IFC.1/Privileged User).

The information flow control is based on security attributes of the subjects, the identity of the subject and the type of request (FDP_IFF.1/Signer and FDP_IFF.1/Privileged User, FDP_UIT.1/SecAttUsr, FDP_UIT.1/SAD).

The following information flow is permitted by the TOE for Signer and Privileged User (FDP_IFF.1/Signer):

- The TOE shall be initialized with TOE_Maintenance before performing requests for other operations.
- All rules specified for Signing shall be performed by the TOE.
- The TOE shall not perform any request, if an operation defined by the rules deposited in the TOE cannot be performed successfully.

- The TOE shall only allow a Signer or Privileged User to request for
  - maintaining Signer security attributes
  - the generation of a key pair

  when the Signer is already created in the TOE.
- The TOE shall only allow a Signer and Privileged User to request for the deletion of a signing key pair when the Signer is already created in the TOE and a signing key pair is already created and assigned to the Signer.
- The TOE shall only allow a Signer to request for the creation of a signature/seal when the Signer is already created in the TOE followed by the creation of a key pair for the Signer.
- The TOE shall perform a Signing request based on the accessed Signer security attributes.
- The TOE shall return the signature/seal as result of a successful Signing request.

The following information flow is permitted by the TOE for Privileged User and Privileged User Admin and Privileged User Technical (FDP_IFF.1/Privileged User):

- The TOE shall be initialized with TOE_Maintenance before performing any request for other operations.
- All rules specified for operations shall be performed by the TOE.
- The TOE shall not perform any request, if an operation defined by the rules deposited in the TOE cannot be performed successfully.

- The TOE shall perform requests for
  - creating Signer
  - creating Privileged User

  by Privileged User based on the accessed Privileged User security attributes.
- The TOE shall perform requests for maintaining the TOE configuration by Privileged User Admin based on the accessed Privileged User Admin security attributes.
- The TOE shall perform requests for
  - creating Signer

  by Privileged User Technical based on the accessed Privileged User Technical security attributes.

For all requests, the TOE must select and execute the appropriate TOE configuration data and rules based on the subject's identity and/or the request type.

7.5 SF5 – Self-Protection

The TOE is a software solution. It is operated in a tamper resistant environment. The tamper resistant environment is provided by the TOE environment. In particular, the requirements mentioned under OE.ENV are implemented by the TOE environment. Specifically, the TOE is operated in a separately secured network zone of a qualified trusted service provider (TSP) that meets the requirements of [EN419241-2] "7.3 Security objectives for the operating environment".

To ensure the integrity of the TOE binaries, a SHA-512 hash value of each TOE binary is generated once during TOE installation. Thereby the TOE calls a function provided by the underlying operation system, to calculate the hash values (SHA-512) and store the hash values in a file. This file is signed by the connected hardware security module (SHA-512 and ECDSA 256 bit).[129] The signature of this file and the hash values themselves get verified on each start-up of the TOE and upon the request of an administrator. Once the mathematical correctness could be verified, the TOE calls a function provided by the underlying operation system, to calculate the SHA-512 hash values of each TOE binary. Afterwards the TOE compares the calculated hash values to those stored within the signed file. If the signature verification fails or any hash value does not correspond to the hash values stored within the signed file, the start of the TOE will abort.

Furthermore the operator of the system receives a digitally signed configuration file (SHA-512 and ECDSA 256 bit) used for a secure operation of the TOE. This file is delivered together with the server component binaries. The TOE initiates the verification of the mathematical correctness of the signature on each start-up of the TOE. Therefor the TOE provides the signature to the HSM, where the signature verification is performed. After a successful verification the configuration is loaded into the application memory so that a change of the configuration file causes no effect to the behavior of the application.

[129] The way in which the signature on the checksum file is generated must be specified finally.

7.6 SF6 – Trusted Paths/Channels

The TOE provides per TOE subject

- Privileged User
- Signer
- Privileged User Admin
- Privileged User Technical

and for

- a cryptographic module certified according to [EN419221-5]

a communication channel between itself and the TOE subject which is logically distinct from other communication channels (FTP_TRP.1/SSA, FTP_TRP.1/SIC, FTP_TRP.1/RSSA, FTP_TRP.1/Admin, FTP_TRP.1/Technical, FTP_ITC.1/CM).

The TOE may initiate communication via a trusted channel to

- a cryptographic module certified according to [EN419221-5].

The TOE permits

- a cryptographic module certified according to [EN419221-5]
- Signer remotely through SIC or SSA
- Privileged User through SSA
- Privileged User Admin
- Privileged User Technical

to initiate communication via the trusted channel (FTP_TRP.1/SSA, FTP_TRP.1/SIC, FTP_TRP.1/RSSA, FTP_TRP.1/Admin, FTP_TRP.1/Technical, FTP_ITC.1/CM).

A trusted channel is maintained as long and used for as many (even parallel) transactions as desired. If a trusted channel is aborted during the processing of an operation the permitted subjects or a cryptographic module certified according to [EN419221-5] establish a new trusted channel and determines the status of the request or transaction in order to continue with it.

Trusted communication channels in the form of TLS tunnels with mutual certificate-based authentication are set up before any communication between the permitted subjects and the TOE. Between a [EN419221-5] certified cryptographic module and the TOE, the CM provided 'Secure Messaging' mechanism (see [CMDS]) is used to secure the channels. Trusted communication channels used in the following in order to protect integrity and confidentiality during transmission and to authenticate requests and responses (FPT_TDC.1).

The implementation of the TLS tunnels complies with the requirements of [RFC8446]. The TOE uses sufficiently strong cryptographic algorithms according to [TR02102-2] to secure the trusted channels.

That means only the following cipher suites are allowed and recommended to use:

- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_AES_128_CCM_SHA256

The TOE does not accept weaker algorithms during TLS tunnel setup. The TLS configuration required for implementing the TLS tunnel is stored in the TOE configuration. On the client side, the TLS configuration is configured in the JRE environment.

When using 'Secure Messaging' provided by the CM a session between the cryptographic module and the TOE is negotiated using the Diffie-Hellman key agreement protocol resulting in a session encryption key and a session MAC key. Both keys are AES-based and have got a length of 256 bit. Additionally the CM provides signatures over the answer data calculated with the HSM Authentication Key (a 3072 bit RSA key). The Signatures can be used for authentication of the CM towards the TOE. Using 'Secure Messaging' every command and answer sent to or received from the CM is encrypted and protected with a MAC (AES with CMAC).

8. References

See attached document "CC_proNEXT-SAM_References*".

9. Abbreviations

See attached document „CC_proNEXT-SAM_Abbreviations*“.

Appendix

Appendix A – Authentication

If a User is registered with the SSASC for Server Signing to become a Signer and thus possesses a Signer Certificate, he is basically able to initiate Server Signing with the SSASC. For each process of Server Signing, the Signer has to authenticate himself at the SSASC, more precisely the SAM.

The following authentication methods are particularly suitable:

- Username/Password
- Hardware Token
- eID Card
- any other authentication mechanism according to prEN 419241-1:2017 SRA_SAP.1.1

Username/Password

This authentication method uses user ID password pairs (one factor: knowledge) to prove identities. The basic version of this authentication method is vulnerable to compromise (recording, replay, social engineering, etc.). In the context of Server Signing and to comply with SCAL2 it must therefore be extended by a second factor. Procedures for assigning TANs are suitable for this purpose. For example, this can be implemented by sending TANs via SMS to mobile phones (factor: possession).

Hardware Token

This SCAL2 compliant method uses 2 factors to authenticate a user. The hardware token used (e.g. a smart card) represents one factor (possession). The second factor (knowledge) is realized by the so called PIN, which must be entered during authentication using the hardware token. If a factor is lost (forgotten, spied on, etc.), the user is protected against unauthorised authentication.

eID Card

eID cards are a SCAL2 compliant 2-factor based authentication method. The factors of possession and knowledge are provided through the use of a physical eID card and an additional PIN to be applied. eID cards are issued by the governments and cryptographic protocols are used to secure communication (e.g. for mutual authentication and secure reading of user data between the involved component). In principle, all national eID cards of european countries are suitable for use in the context of server signing, provided there is mutual recognition and appropriate technical implementation.