TÜVIT

CERTIFICATION REPORT

Certification file:	**TUVIT-TSZ-CC-9265-2023**
Product / System:	software module Smart-ID SecureZone, version 11.5.23
Product manufacturer:	SK ID Solutions AS Pärnu avenue 141 11314 Tallinn, Estonia
Customer:	see above
Evaluation body:	TÜV Informationstechnik GmbH TÜV NORD GROUP Evaluation Body for IT Security Am TÜV 1 45307 Essen, Germany
Evaluation report:	*Version 2 as of 2023-09-22* project-number: *8120584434* authors: Arzu Sarial
Result:	EAL4 augmented by AVA_VAN.5
Evaluation stipulations:	none
Certifier:	Dr. Silke Keller
Certification stipulations:	none
Version / Date:	Version 1.0, 2023-09-26

………………………………………	………………………………………
Dr. Christoph Sutter	Dr. Silke Keller
Head of Certification Body	Certifier

Contents

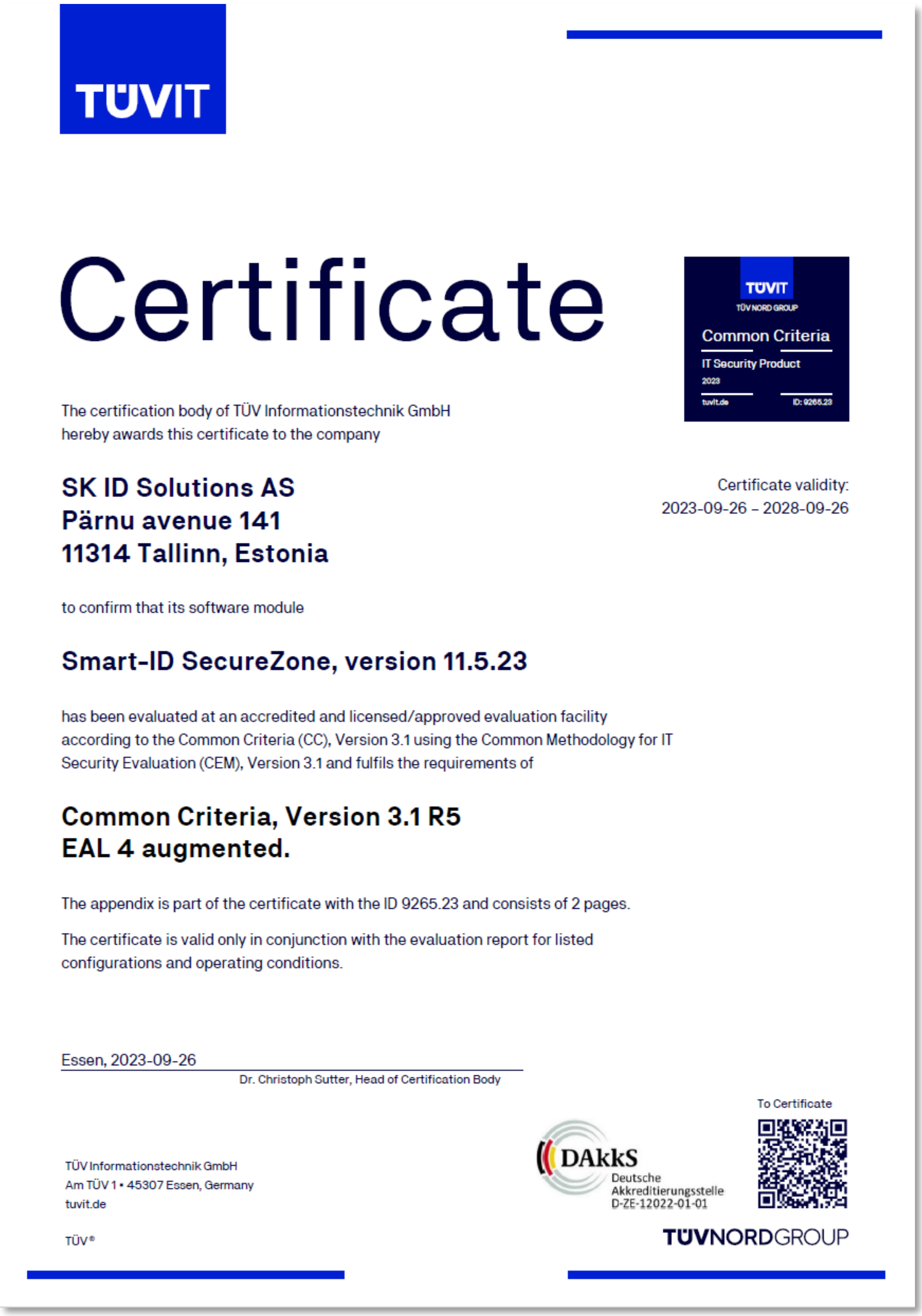
Part A

Certificate and Background of the Certification

Part A presents a copy of the issued certificate and summarizes

- information about the certification body,
- the certification procedure and
- the performance of evaluation and certification.

1 The Certificate



TÜVIT

Certificate

TÜVIT
TÜV NORD GROUP
Common Criteria
IT Security Product
2023
tuvit.de ID: 9265.23

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

SK ID Solutions AS
Pärnu avenue 141
11314 Tallinn, Estonia

Certificate validity:
2023-09-26 – 2028-09-26

to confirm that its software module

Smart-ID SecureZone, version 11.5.23

has been evaluated at an accredited and licensed/approved evaluation facility according to the Common Criteria (CC), Version 3.1 using the Common Methodology for IT Security Evaluation (CEM), Version 3.1 and fulfils the requirements of

Common Criteria, Version 3.1 R5
EAL 4 augmented.

The appendix is part of the certificate with the ID 9265.23 and consists of 2 pages.

The certificate is valid only in conjunction with the evaluation report for listed configurations and operating conditions.

Essen, 2023-09-26
Dr. Christoph Sutter, Head of Certification Body

To Certificate

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

DAkkS
Deutsche
Akkreditierungsstelle
D-ZE-12022-01-01

TÜV®

TÜVNORDGROUP

2 Certification Body – TÜVIT

The Certification Body of *TÜV Informationstechnik GmbH*[1] – TÜV NORD GROUP – was established in 1998 and offers a variety of services in the context of security evaluation and validation.

TÜVIT is accredited for certification of IT security products according to ITSEC and Common Criteria by *Deutsche Akkreditierungsstelle GmbH under* registration no. D-ZE-12022-01-01 and performs its projects under a quality management system certified against ISO 9001.

3 Specifications of the Certification Procedure

The certification body conducts the certification procedure according to the criteria laid down in the following:

- DIN EN ISO/IEC 17065
- TÜVIT Certification Scheme
- TÜVIT Certification Conditions
- Common Criteria for Information Technology Security Evaluation (CC) part 1-3, version 3.1 revision 5, April 2017.
- Common Methodology for Information Technology Security Evaluation (CEM), version 3.1 revision 5, April 2017.
- Application Notes and Interpretations of the Scheme (AIS), published by BSI.

4 Performance of Evaluation and Certification

The certification body monitors each individual evaluation to ensure uniform procedures, interpretations of the criteria, and ratings. The software product Smart-ID SecureZone, version 11.5.23 has undergone the certification procedure at TÜVIT certification body.

The evaluation of the software product Smart-ID SecureZone, version 11.5.23 was conducted by the evaluation body for IT-security of TÜVIT and concluded on September 22, 2023. The TÜVIT evaluation body is recognised by BSI.

Sponsor as well as the developer is SK ID Solutions AS. Distributor of the product is SK ID Solutions AS.

[1] in the following termed shortly TÜVIT

The certification was concluded with

- the comparability check and
- the preparation of this certification report.

This work was concluded on September 26, 2023. The confirmation of the evaluation assurance level (EAL) only applies on the condition that:

- all stipulations regarding generation, configuration and operation, as given in part B of this report, are observed,
- the product is operated – where indicated – in the environment described.

This certification report applies only to the version of the product indicated here. The validity of the certificate can be extended to cover new versions and releases of the product, provided the applicant applies for re-certification of the modified product, in accordance with the procedural requirements, and provided the evaluation does not reveal any security deficiencies.

This certificate is not an endorsement of the IT product by the TÜV Informationstechnik GmbH or any other organisation that recognises or gives effect to this certificate, and no warranty of the IT product by TÜV Informationstechnik GmbH or any other organisation that recognises or gives effect to this certificate, is either expressed or implied.

In order to avoid an indefinite usage of the certificate when evolved attack methods require a re-assessment of the products resistance to state of the art attack methods, the maximum validity of the certificate has been limited. The certificate issued on September 26, 2023 is valid until September 26th 2028. The validity date can be extended by re-assessment and re-certification.

With regard to the meaning of the evaluation assurance levels (EAL), please refer to part C of this report.

Within the last two years, the certifier did not render any consulting or other services for the company ordering the certification and there was no relationship between them that might have an influence on his assessment.

The certifier did not participate at any time in test procedures for the product, which forms the basis of the certification.

5 Publication

The following Certification Results consist of pages B-1 to B-16. The certification report and the certificate for product Smart-ID SecureZone, version 11.5.23 will be included in the TÜVIT certification list (http://www.tuvit.de).

Further copies of this certification report may be ordered from the sponsor of the product. The certification report may also be obtained in electronic form at the internet address of TÜVIT as stated above.

Part B

Certification Result

The following results represent a summary of

- the security target of the sponsor for the target of evaluation,
- the relevant evaluation results from the evaluation facility, and
- complementary notes and stipulations of the certification body.

Contents of the Certification Result

1 Executive Summary

The target of evaluation (TOE) is the software product ***Smart-ID SecureZone, version 11.5.23***

The TOE architecture is described in chapter 5. The TOE is the server-side software implementation of the Smart-ID system, developed to provide a solution for the digital signature creation. The TOE is a Java application server package, which implements the server-side functions of the Threshold Signature Scheme Protocol (TSSP) for the signer and the management functions for the administrators. The signer can use the TOE services to enroll new key pairs, create digital signatures and to destroy the key pairs.

The TOE has the following functions:

- Creation of Qualified Electronic Signatures, complying with eIDAS regulation reg. (EU) 910/2014 [eIDAS];
- Enrolment and destruction of the Signer's key pair;
- Security management and access control functions.

The security target is the basis of this certification. It is not based on a certified protection profile.

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 4 (Evaluation Assurance Level 4) augmented by AVA_VAN.5 (Advanced methodical vulnerability analysis).

The TOE's security functional requirements were taken from CC part 2 (i. e. the set is CC part 2 conformant) [CC]. They are implemented by the following eight security functions:

Security Function	Description
SF.Authentication	SF.Authentication authenticates users with different methods. It also locks the authentication procedure in case of consecutive unsuccessful authentication tries.
SF.AccessControl	SF.AccessControl ensures that all three main groups of users are only allowed to perform operations, which are intended to be able for their role.
SF.Audit	SF.Audit generates audit records of the important system events by using standard Java toolset.
SF.KeyGen	SF.KeyGen ensures the use of the FIPS 140-2-certified HSM to perform the most of the key generation operations. In case the HSM doesn't support generation and management of particular key type, TOE is generating that by himself.

Security Function	Description
SF.CryptoAlgorithms	SF.CryptoAlgorithms ensures the use of the FIPS 140-2-certified HSM to perform most of the key usage operations. In cases the HSM doesn't support operations with the particular key type, TOE is implementing this by himself: • Computation of signatures, • Creation and verification of RSA signatures, • Encryption/decryption of JSON Web Encryption (JWE) messages.
SF.KeyZer	SF.KeyZer enforces the TOE to destroy cryptographic keys after they are no longer used.
SF.TrustedPath	SF.TrustedPath implements JWE messages for the communication between the TOE and the Smart-ID App TSE.
SF.SecureChannel	SF.SecureChannel ensures that vendor-specific proprietary communication channel is used when connecting with HSM or database, such as nCipher impath and PostgreSQL connections.

A more detailed description of the TOE security functions can be found in section 7.3 of the public ST, which is attached as part E of this certification report.
Assets for the TOE comprise the integrity and/or confidentiality security functions of the TOE and the data used like the data to be signed representation, the electronic signature with the different shares, the signature verification data and the cryptographic keys during operation.

The 17 threats comprise threats to create one or more signature or change data to be signed under the name of the signer and to decrease the trust in the signatures created with the service Smart-ID Trust Service Provider (TSP) and the security of the TOE. The threats are organised within the ST in the following subsections in order to present the closely related threats next to each other:

- Threats related to the key enrolment
- Threats related to impersonation of the Signer within the signing process
- Threats related to signature forgery
- Other threats (e.g. attacks to create signatures, attacks to audit logs)

There are 10 organisational security policies for the TOE.

A more detailed description of the threats, organisational security policies and assumptions can be found in sections 4.4, 4.5 and 4.6 of the public ST, which is attached as part E of this certification report. The certification covers the configurations of the TOE as outlined in chapter 8.

2 Identification of the TOE

The Target of Evaluation (TOE) is the software product Smart-ID SecureZone, version 11.5.23.

The TOE delivery consists of the following parts:

1. TOE Documentation (see chapter 6)
2. Smart-ID SecureZone

The TOE including the TOE documentation is composed in a software zip-archive, which is delivered via a delivery system. The integrity of the delivered TOE has to be checked comparing the SHA-384 hash values of the TOE.

No.	Type	Item / Identifier	Form of Delivery
1.	SW	SecureZone binary package (file name: sz-11.5.23_RELEASE-all.jar) 5cfcafdd4ed4dfbd9c414b615985abbb7310bc74b47211c3b541389cbe7b1086eb146a41b39a541b92ed1efd500c94a7	Secure file transfer system
2.	SW	sz-boot-11.5.23_RELEASE-executable.jar (file name: sz-boot-11.5.23_RELEASE-executable.jar) a7fdb96566bad7be962f9095b5bc9d95c76d03e3781213139caa3bf01f4840026ef874de84b20f759801657d8471a924	Secure file transfer system
3.	SW	SecureZone Admin CLI binary package (file name: secure-zone-cli.jar) 2b46995d8fc7b05af99214dbf9a26be935ff6768ac5b5276124bb19b21fcf043f5ac0be1b4833886de73b4a9b84347aa	Secure file transfer system
4.	SW	Liquibase changesets and scripts for initializing and updating the database schema (file name: liquibase.tar) 619252e50b20900cc7e8295b13127903a21407cf98c37ab1b43bd9b4ce687b9fefd14e5c8bf1739672398e05afc96170	Secure file transfer system
5.	DOC	Installation Guide for SecureZone v2.32_v133 f817289ac42b9241b8d648924746d0b67f92a9deb96b5cab164fde02346518d6092a59acbdebeb8b649944006297988a	Secure file transfer system
6.	DOC	Administration Guide for SecureZone v2.15_v78 7e65d4d7a7b366a0b8f4a12958987161ab51e4e2bd6a0c073ab04e1c51e3277138ce70d1677a1fec6c9a2fd55fccfa7a	Secure file transfer system
7.	DOC	Smart-ID SecureZone monitoring guide v1.6_v19 2dac8a1f63952a00759febb0b868556218861857a1b28eda1172debfebcb4a0661da35e33a4d26e1121e71107f39e844	Secure file transfer system
8.	DOC	Signer User Guidance information for SecureZone and TSE library operators v2.8_v11 1850e329825b29918e538fd0181add2137021a085c7bd6764ee06fabc3d0e0d1ff7c7e58545097082043d8b01a31e66d	Secure file transfer system
9	DOC	Release Notes document (file name: smartid-sz-release-notes-11.5.23.txt)	Secure file transfer system, delivered

No.	Type	Item / Identifier	Form of Delivery
			in digitally signed container containing overview of changes and checksums of all delivered components
10.	DOC	Checksums txt (file name: smartid-sz-checksums-11.5.23.txt)	Secure file transfer system, delivered in digitally signed container containing overview of changes and checksums of all delivered components

The delivery of the HSM and mobile client must be performed according to their certification requirements.

3 Security Policy

The security policy enforced is defined by the selected set of Security Functional Requirements and implemented by the TOE. It covers the following issues:

- Security Audit,
- Cryptographic Support,
- User Data Protection,
- Identification and Authentication,
- Security Management,
- Protection of the TSF,
- Trusted Path/Channels.

Specific details concerning the different security policies can be found in section 7 of the public ST, which is attached as part E of this certification report.

4 Assumptions and Clarification of Scope

The assumptions defined in the Security Target and some aspects of threats and Organisational Security Policies are not covered by the TOE itself. These aspects lead to specific security objectives to be fulfilled by the TOE environment (see the following chapters in the Security Target:

- 5.2 Security Objectives for the Environment fulfilled by HSM
- 5.3 Security Objectives for the Environment fulfilled by TSE
- 5.4 Security Objectives for the Environment fulfilled by other components

5 Architectural Information

The TOE consists of six modules:

Name of Element	Description
TSSP module	The TSSP module is responsible for full execution of the TSSP.
Configuration module	This is an administrative command line tool, which contains all the logic for system key generation and other administrative operations.
Command line interface module	This module is responsible for configuration file loading and verification.
JSON-RPC controller module	A glue module for converting JSON-RPC calls to Java calls.
Audit logging module	This is module is called by all the modules above to create audit log records.
Support functionality module	This is a support module for non-security-critical operations.

6 Documentation

The following documentation is provided with the product by the developer to the consumer (see chapter 2).

7 IT Product Testing

The developer’s testing to systematically test the TOE security functionality / TSFI, was executed with the following approach:

- Tests cover the TSFI and their behavioral aspects defined in [ADV], by testing each TSFI.
- Automated and manual, black-box and white, direct and indirect tests are applied.
- Positive and negative tests are executed.
- Tests cover also all TSF modules.

The evaluation body testing started on July 11th, 2023 and was successfully concluded on July 13th, 2023. The evaluator's objective was to test the functionality of the TOE systematically against the security functionality description in [ST] and [ADV]. In order to do this, the evaluation body performed the following tasks:

- Repeat the developer's tests,
- Devise and execute own functional tests.
- Based on a list of potential vulnerabilities applicable to the TOE in its operational environment the evaluators devised the attack scenarios for penetration tests when they were of the opinion, that those potential vulnerabilities could be exploited in the TOE's operational environment. All other evaluation input was used for the creation of the tests as well. Specifically the test documentation provided by the developer was used to find out if there are areas of concern that should be covered by tests of the evaluation body.

8 Evaluated Configuration

The TOE Smart-ID SecureZone, version 11.5.23 is delivered in one fixed configuration and no further generation takes place.

The Security Target [ST] has identified solely one configuration of the TOE under evaluation.

9 Results of the Evaluation

9.1 CC specific results

The Evaluation Technical Report [ETR] was provided by TÜVIT's evaluation body according to the requirements of the Scheme, the Common Criteria [CC], the Methodology [CEM] and the Application Notes and Interpretations of the Scheme [AIS].

As a result of the evaluation the verdict PASS is confirmed for the following assurance components:

- All components of the EAL4 package including the class ASE as defined in the CC (see also part C of this report).
- The component AVA_VAN.5 augmented for this TOE evaluation.

The verdicts for CC, part 3 assurance classes and components (according to EAL4+ augmented by AVA_VAN.5 and the class ASE for the Security Target Evaluation) are summarised in the following table:

Assurance classes and components			**Verdict**
Development		**ADV**	**PASS**
	Security architecture description	ADV_ARC.1	PASS
	Complete functional specification	ADV_FSP.4	PASS
	Implementation representation of the TSF	ADV_IMP.1	PASS
	Basic modular design	ADV_TDS.3	PASS
Guidance documents		**AGD**	**PASS**
	Operational user guidance	AGD_OPE.1	PASS
	Preparative procedures	AGD_PRE.1	PASS
Life-cycle support		**ALC**	**PASS**
	Production support, acceptance procedures and automation	ALC_CMC.4	PASS
	Problem tracking CM coverage	ALC_CMS.4	PASS
	Delivery procedures	ALC_DEL.1	PASS
	Identification of security measures	ALC_DVS.1	PASS
	Developer defined life-cycle model	ALC_LCD.1	PASS
	Well-defined development tools	ALC_TAT.1	PASS
Security Target evaluation		**ASE**	**PASS**
	Conformance claims	ASE_CCL.1	PASS
	Extended components definition	ASE_ECD.1	PASS
	ST introduction	ASE_INT.1	PASS
	Security objectives	ASE_OBJ.2	PASS
	Derived security requirements	ASE_REQ.2	PASS
	Security problem definition	ASE_SPD.1	PASS
	TOE summary specification	ASE_TSS.1	PASS
Tests		ATE	PASS
	Analysis of coverage	ATE_COV.2	PASS
	Testing: security enforcing modules	ATE_DPT.1	PASS
	Functional testing	ATE_FUN.1	PASS
	Independent testing - sample	ATE_IND.2	PASS
Vulnerability Assessment		**AVA**	**PASS**

Assurance classes and components			Verdict
	Advanced methodical vulnerability analysis	AVA_VAN.5	PASS

9.2 Results of the cryptographic assessment

The strength of the cryptographic algorithms was not rated in the course of this certification procedure (see [BSIG], section 9, para. 4, clause 2). But Cryptographic Functionalities with a security level of lower than 100 bits can no longer be regarded as secure without considering the application context. Therefore, for these functionalities it shall be checked whether the related crypto operations are appropriate for the intended system. Some further hints and guidelines can be derived from [TR-02102].

Any Cryptographic Functionality that is marked with 'No' in column 'Security Level above 100 Bits' of the following table achieves a security level of lower than 100 Bits (in general context).

No.	Purpose	Cryptographic Mechanism	Implementation Standard	Key Size in Bits	Security Level above 100 Bits	Evaluator's comments
1.	**Generation of signature verification** data to perform digital signature verification.	RSA PKCS1-v1_5, RSA-PSS	[RFC8017], TSSP	3071, 3072, 4095, 4096, 6143, 6144, 8191, 8192	Yes	FCS_COP.1/RSA_SCD RSA PKCS1-v1_5 is not recommended in new systems and stated as legacy algorithm. But no successful attacks are known and additionally RSA-PSS was implement-ted.
2.	Generation of symmetric encryption/decryption and integrity protection key to create the **secure communication channel** between TSE and TOE.	Diffie-Hellman station-to-station protocol and concatKDF	[RFC2631], [RFC3526], [SP800-56A Rev. 2]	2048 up to 4096	Yes	FCS_CKM.1.1/DH_TEK
3.	**Authentication of the Signer:** Verification of the App's signature share to check if the Signer provided	RSA PKCS1-v1_5, RSA-PSS, RSAESOAEP	[RFC8017], TSSP	3072 bits up to 16384 bits	Yes	FCS_COP.1/RSA_Other

No.	Purpose	Cryptographic Mechanism	Implementation Standard	Key Size in Bits	Security Level above 100 Bits	Evaluator's comments
	the correct PIN (knowledge-based factor). Verifying that the provided OTP matches with the value in the database (possession-based factor).					
4.	**Signature creation:** Generation of RSA signature to generate a compound signature of the Signer	RSA PKCS1-v1_5, RSA-PSS	[RFC8017], TSSP	3071, 3072, 4095, 4096, 6143, 6144, 8191, 8192	Yes	FCS_COP.1/RSA_SCD
5.	**Secure channel:** perform message decryption and generation of signature when securing the communication between TOE and TSE library in the possession of the Signer.	AES	[FIPS 197]	128 bits or longer	Yes	FCS_COP.1.1/AES
6.	**Secure channel:** message encryption and decryption between the TOE and a specific instance of the TSE library used by the Signer.	AES	[FIPS 197]	128	Yes	Presently without maximum validity.
7.	**Secure channel:** Provide and verify the authenticity and integrity of the messages between the TOE and a specific instance of the TSE library used by the Signer	HMAC	[FIPS 198-1]	128	Yes	FCS_COP.1.1/HMAC
8.	**Secure storage:** encryption, decryption along	AES and HMAC	[FIPS 197] [FIPS 198-1]	128	Yes	FCS_CKM.1.1/AES_KWK

No.	Purpose	Cryptographic Mechanism	Implementation Standard	Key Size in Bits	Security Level above 100 Bits	Evaluator's comments
	with integrity protection and verification of the key material in the database.					
9.	**Secure storage:** encryption and decryption of the sensitive data fields in the database.	AES	800-133r2 [23]	128	Yes	FCS_CKM.1.1/AES_DEK
10.	Digest computations for key generation operations.	SHA-256	[FIPS 180-4]	256 bits, 384 bits, 512 bits,	Yes	FCS_COP.1.1/SHA-2, FCS_COP.1.1/SHA-3
11.	Digest computations for integrity verification operations.	SHA-256	[FIPS 180-4]	256 bits, 384 bits, 512 bits	Yes	FCS_COP.1.1/SHA-2, FCS_COP.1.1/SHA-3

10 Evaluation Stipulations, Comments, and Recommendations

The evaluation technical report contains no stipulations or recommendations.

11 Certification Stipulations and Notes

There are no stipulations or notes resulting from the certification report.

12 Security Target

The security target [ST] for *Smart-ID SecureZone, version 11.5.23* is included in part E of this certification report.

13 Definitions

13.1 Acronyms

AGD	Guidance Documents
CC	Common Criteria for Information Technology Security Evaluation (referenced to as [CC])
CEM	Common Methodology for Information Technology Security Evaluation (referenced to as [CEM])
EAL	Evaluation Assurance Level
EEPROM	Electrical Erasable and Programmable Read Only Memory
ES	Embedded Software
EU	European Union
FSP	Functional Specification
FIPS	Federal Information Processing Standard
HLD	High-level Design
HSM	Hardware Security Module
IC	Integrated Circuit
JWE	JSON Web Encryption
JSON	Java Script Object Notation
IF	Interface
IGS	Installation, Generation and Start-up
OS	Operating System
OSP	Organisational Security Policy
PP	Protection Profile
RPC	Remote Procedure Call
RSA	Signature Algorithm of Rivest, Shamir, Adleman
SAR	Security Assurance Requirement
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirement
SIF	Sub-interface
SOF	Strength of Function
SS	Sub-system
SSL	Secure Sockets Layer
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSE	Threshold Signature Engine
TSF	TOE Security Functions
TSFI	TOE Security Function Interfaces

TSP	TOE Security Policy
TSSP	Threshold Signature Scheme Protocol
VLA	Vulnerability Analysis

13.2 Glossary

Augmentation The addition of one or more assurance component(s) from Part3 to an EAL or assurance package.

Extension The addition to an ST or PP of functional requirements not contained in Part 2 and/or assurance requirements not contained in Part 3 of the CC.

Formal Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.

Informal Expressed in natural language.

Object An entity within the TSC that contains or receives information and upon which subjects perform operations.

Protection Profile An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.

Security Function A part or parts of the TOE that have to be relied upon for enforcing a closely related subset of the rules from the TSP.

Security Target A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

Semiformal Expressed in a restricted syntax language with defined semantics.

Strength of Function A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behaviour by directly attacking its underlying security mechanisms.

Subject An entity within the TSC that causes operations to be performed.

Target of Evaluation An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

TOE Security Functions A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

TOE Security Policy A set of rules that regulate how assets are managed, protected and distributed within a TOE.

TSF Scope of Control The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.

14 Bibliography

[AGD-Inst] Installation Guide for SecureZone, Version 2.32_v133 as of 2023-08-18

[AGD-Admin] Administration Guide for SecureZone, Version 2.15_v78 as of 2023-06-21

[AGD-Mon] Signer User Guidance information for SecureZone and TSE library operators, Version 1.6_v19 as of 2018-08-30

[AGD-User] Smart-ID SecureZone monitoring Guide, Version 1.1_v18 as of 2023-06-21

[AIS] Application Notes and Interpretations of the Scheme (AIS) as relevant for the TOE, Bundesamt für Sicherheit in der Informationstechnik

[AIS1] Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 1, Durchführung der Ortsbesichtigung in der Entwicklungsumgebung des Herstellers, Version 13, 2008-08-14, Bundesamt für Sicherheit in der Informationstechnik.

[AIS11] Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 11, Programmiersprachen und Compiler, Version 2.0, 1998-02-02, Bundesamt für Sicherheit in der Informationstechnik.

[AIS14] Anwendungshinweise und Interpretationen zum Schema, AIS 14: Anforderungen an Aufbau und Inhalt der ETR-Teile (Evaluation Technical Report) für Evaluationen nach CC (Common Criteria), Version 7, 2010-08-03, Bundesamt für Sicherheit in der Informationstechnik.

[AIS19] Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 19, Anforderungen an Aufbau und Inhalt der Zusammenfassung des ETR (Evaluation Technical Report) für Evaluationen nach CC (Common Criteria), Version 9, 2014-11-03, Bundesamt für Sicherheit in der Informationstechnik.

[AIS32] Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 32, CC-Interpretationen im deutschen Zertifizierungsschema, Version 7, 2011-06-08, Bundesamt für Sicherheit in der Informationstechnik.

[AIS40] Application Notes and Interpretation of the Scheme (AIS), AIS 40, Use of Interpretation for Security Evaluation and Certification of Digital Tachographs, Version 1, 2005-06-28, Bundesamt für Sicherheit in der Informationstechnik.

[AIS41] Application Notes and Interpretation of the Scheme (AIS) – AIS 41, Guidelines for PPs and STs, Version 2, 2011-01-31, Bundesamt für Sicherheit in der Informationstechnik.

[AIS42] Application Notes and Interpretation of the Scheme (AIS), AIS 42, Guidelines for the Developer Documentation, Version 1, 2008-05-21, Bundesamt für Sicherheit in der Informationstechnik.

[CC] Common Criteria for Information Technology Security Evaluation, Version 3.1,
Part 1: Introduction and general model, Revision 5, April 2017
Part 2: Security functional requirements, Revision 5, April 2017
Part 3: Security assurance requirements, Revision 5, April 2017

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

[eIDAS] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, Aug. 2014.

[ETR] Evaluation Technical Report, TÜV Informationstechnik GmbH, version 2, *2023-09-22*, project-number: 8120584434

[ST] Security Target of the Smart-ID SecureZone, , Version 3.0.8 as of 2023-08-28, SK ID Solutions AS

[TR-02102] BSI - Technische Richtlinie TR-02102 Kryptographische Verfahren: Empfehlungen und Schlüssellängen (consisting of [TR-02102-1]/[TR-02102-2]/[TR-02102-3])

[TR-02102-1] BSI - Technische Richtlinie TR-02102-1, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Version 2023-01, 2023-01-09, Bundesamt für Sicherheit in der Informationstechnik.

[TR-02102-2] BSI - Technische Richtlinie TR-02102-2, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 2 – Verwendung von Transport Layer Security (TLS), 2023-01, 2023-01-17, Bundesamt für Sicherheit in der Informationstechnik.

[TR-02102-3] BSI - Technische Richtlinie TR-02102-3, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 3 – Verwendung von Internet Protocol Security (IPsec) und Internet Key Exchange (IKEv2), 2023-01, 2023-01-17, Bundesamt für Sicherheit in der Informationstechnik.

[TR-02102-4] BSI - Technische Richtlinie TR-02102-3, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 4 – Verwendung von Secure Shell (SSH), Version 2023-01, 2023-01-17, Bundesamt für Sicherheit in der Informationstechnik.

Part C

Excerpts from the Criteria

The excerpts from the criteria are dealing with

- conformance results
- assurance categorization
- evaluation assurance levels
- strength of security function
- vulnerability analysis

CC Part 1:

Conformance Claim

The conformance claim indicates the source of the collection of requirements that is met by a PP or ST that passes its evaluation. This conformance claim contains a CC conformance claim that:

- describes the version of the CC to which the PP or ST claims conformance.
- describes the conformance to CC Part 2 (security functional requirements) as either:

  CC Part 2 conformant - A PP or ST is CC Part 2 conformant if all SFRs in that PP or ST are based only upon functional components in CC Part 2, or

  CC Part 2 extended - A PP or ST is CC Part 2 extended if at least one SFR in that PP or ST is not based upon functional components in CC Part 2.
- describes the conformance to CC Part 3 (security assurance requirements) as either:

  CC Part 3 conformant - A PP or ST is CC Part 3 conformant if all SARs in that PP or ST are based only upon assurance components in CC Part 3, or

  CC Part 3 extended - A PP or ST is CC Part 3 extended if at least one SAR in that PP or ST is not based upon assurance components in CC Part 3.

Additionally, the conformance result may include a statement made with respect to sets of defined requirements, in which case it consists of one of the following:

- Package name Conformant - A PP or ST is conformant to a pre-defined package (e. g. EAL) if:
  - the SFRs of that PP or ST are identical to the SFRs in the package, or
  - the SARs of that PP or ST are identical to the SARs in the package.
- Package name Augmented - A PP or ST is an augmentation of a pre-defined package if:
  - the SFRs of that PP or ST contain all SFRs in the package, but have at least one additional SFR or one SFR that is hierarchically higher than an SFR in the package.
  - the SARs of that PP or ST contain all SARs in the package, but have at least one additional SAR or one SAR that is hierarchically higher than an SAR in the package

Note that when a TOE is successfully evaluated to a given ST, any conformance claims of the ST also hold for the TOE. A TOE can therefore also be e. g. CC Part 2 conformant.

Finally, the conformance result may also include a statement made with respect to Protection Profiles, in which case it includes the following:

- **PP Conformant** - A PP or TOE meets specific PP(s), which are listed as part of the conformance result.
- **Conformance Statement** (Only for PPs) - This statement describes the manner in which PPs or STs must conform to this PP: strict or demonstrable. For more information on this Conformance Statement, see Annex D.

CC Part 3:

Class APE: Protection Profile evaluation

Evaluating a PP is required to demonstrate that the PP is sound and internally consistent, and, if the PP is based on one or more other PPs or on packages, that the PP is a correct instantiation of these PPs and packages. These properties are necessary for the PP to be suitable for use as the basis for writing an ST or another PP.

Assurance Class	Assurance Components
Class APE: Protection Profile evaluation	APE_INT.1 PP introduction
	APE_CCL.1 Conformance claims
	APE_SPD.1 Security problem definition
	APE_OBJ.1 Security objectives for the operational environment APE_OBJ.2 Security objectives
	APE_ECD.1 Extended components definition
	APE_REQ.1 Stated security requirements APE_REQ.2 Derived security requirements

APE: Protection Profile evaluation class decomposition

Class ACE: Protection Profile Configuration evaluation

Evaluating a PP-Configuration is required to demonstrate that the PP-Configuration is sound and consistent. These properties are necessary for the PP-Configuration to be suitable for use as the basis for writing an ST or another PP or PP-Configuration.

Assurance Class	Assurance Components
Class ACE: Protection Profile Configuation evaluation	ACE_INT.1 PP-Module introduction
	ACE_CCL.1 PP-Module conformance claims
	ACE_SPD.1 PP-Module Security problem definition
	ACE_OBJ.1 PP-Module Security objectives
	ACE_ECD.1 PP-Module extended components definition
	ACE_REQ.1 PP-Module security requirements
	ACE_MCO.1 PP-Module consistency
	ACE_CCO.1 PP-Configuration consistency

APE: Protection Profile Configuration evaluation class decomposition

Class ASE: Security Target evaluation

Evaluating an ST is required to demonstrate that the ST is sound and internally consistent, and, if the ST is based on one or more PPs or packages, that the ST is a correct instantiation of these PPs and packages. These properties are necessary for the ST to be suitable for use as the basis for a TOE evaluation.

Assurance Class	Assurance Components	
Class ASE: Security Target evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_SPD.1	Security problem definition
	ASE_OBJ.1	Security objectives for the operational environment ASE_OBJ.2 Security objectives
	ASE_ECD.1	Extended components definition
	ASE_REQ.1 ASE_REQ.2	Stated security requirements Derived security requirements
	ASE_TSS.1 ASE_TSS.2	TOE summary specification TOE summary specification with architectural design summary

ASE: Security Target evaluation class decomposition

Security assurance components

“The following Sections describe the constructs used in representing the assurance classes, families, and components.“ “Each assurance class contains at least one assurance family.” “Each assurance family contains one or more assurance components.”

The following table shows the assurance class decomposition:

Assurance Class	Assurance Components	
ADV: Development	ADV_ARC.1	Security architecture description
	ADV_FSP.1 ADV_FSP.2 ADV_FSP.3 ADV_FSP.4 ADV_FSP.5	Basic functional specification Security-enforcing functional specification Functional specification with complete summary Complete functional specification Complete semi-formal functional specification with additional error information

Assurance Class	Assurance Components	
	ADV_FSP.6	Complete semi-formal functional specification with additional formal specification
	ADV_IMP.1 ADV_IMP.2	Implementation representation of the TSF Implementation of the TSF
	ADV_INT.1 ADV_INT.2	Well-structured subset of TSF internals Well-structured internalsADV_INT.3 Minimally complex internals
	ADV_SPM.1	Formal TOE security policy model
	ADV_TDS.1 ADV_TDS.2 ADV_TDS.3 ADV_TDS.4 ADV_TDS.5 ADV_TDS.6	Basic design Architectural design Basic modular design Semiformal modular design Complete semiformal modular design Complete semiformal modular design with formal high-level design presentation
AGD: Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures

Assurance Class	Assurance Components
ALC: Life cycle support	ALC_CMC.1 Labelling of the TOE ALC_CMC.2 Use of a CM system ALC_CMC.3 Authorisation controls ALC_CMC.4 Production support, acceptance procedures and automation ALC_CMC.5 Advanced support
	ALC_CMS.1 TOE CM coverage ALC_CMS.2 Parts of the TOE CM coverage ALC_CMS.3 Implementation representation CM coverage ALC_CMS.4 Problem tracking CM coverage ALC_CMS.5 Development tools CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures ALC_DVS.2 Sufficiency of security measures
	ALC_FLR.1 Basic flaw remediation ALC_FLR.2 Flaw reporting procedures ALC_FLR.3 Systematic flaw remediation
	ALC_LCD.1 Developer defined life-cycle mode ALC_LCD.2 Measurable life-cycle mode
	ALC_TAT.1 Well-defined development tools ALC_TAT.2 Compliance with implementation standards ALC_TAT.3 Compliance with implementation standards - all parts

Assurance Class	Assurance Components
ATE Tests	ATE_COV.1 Evidence of coverage ATE_COV.2 Analysis of coverage ATE_COV.3 Rigorous analysis of coverage
	ATE_DPT.1 Testing: basic design ATE_DPT.2 Testing: security enforcing modules ATE_DPT.3 Testing: modular design ATE_DPT.4 Testing: implementation representation
	ATE_FUN.1 Functional testing ATE_FUN.2 Ordered functional testing
	ATE_IND.1 Independent testing – conformance ATE_IND.2 Independent testing – sample ATE_IND.3 Independent testing – complete
AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey AVA_VAN.2 Vulnerability analysis AVA_VAN.3 Focused vulnerability analysis AVA_VAN.4 Methodical vulnerability analysis AVA_VAN.5 Advanced methodical vulnerability analysis

Assurance class decomposition

Evaluation assurance levels

The Evaluation Assurance Levels (EALs) provide an increasing scale that balances the level of assurance obtained with the cost and feasibility of acquiring that degree of assurance. The CC approach identifies the separate concepts of assurance in a TOE at the end of the evaluation, and of maintenance of that assurance during the operational use of the TOE.

It is important to note that not all families and components from Part 3 are included in the EALs. This is not to say that these do not provide meaningful and desirable assurances. Instead, it is expected that these families and components will be considered for augmentation of an EAL in those PPs and STs for which they provide utility.

Evaluation assurance level (EAL) overview

The above table represents a summary of the EALs. The columns represent a hierarchically ordered set of EALs, while the rows represent assurance families. Each number in the resulting matrix identifies a specific assurance component where applicable.
As outlined in the next section, seven hierarchically ordered evaluation assurance levels are defined in the CC for the rating of a TOE's assurance. They are hierarchically ordered

inasmuch as each EAL represents more assurance than all lower EALs. The increase in assurance from EAL to EAL is accomplished by *substitution* of a hierarchically higher assurance component from the same assurance family (i. e. increasing rigour, scope, and/or depth) and from the *addition* of assurance components from other assurance families (i. e. adding new requirements).

These EALs consist of an appropriate combination of assurance components as described in chapter 2 of CC Part 3. More precisely, each EAL includes no more than one component of each assurance family and all assurance dependencies of every component are addressed.

While the EALs are defined in the CC, it is possible to represent other combinations of assurance. Specifically, the notion of "augmentation" allows the addition of assurance components (from assurance families not already included in the EAL) or the substitution of assurance components (with another hierarchically higher assurance component in the same assurance family) to an EAL. Of the assurance constructs defined in the CC, only EALs may be augmented. The notion of an "EAL minus a constituent assurance component" is not recognised by the CC as a valid claim. Augmentation carries with it the obligation on the part of the claimant to justify the utility and added value of the added assurance component to the EAL. An EAL may also be extended with explicitly stated assurance requirements.

Evaluation assurance level 1 (EAL1) - functionally tested

EAL1 is applicable where some confidence in correct operation is required, but the threats to security are not viewed as serious. It will be of value where independent assurance is required to support the contention that due care has been exercised with respect to the protection of personal or similar information.

EAL 1 requires only a limited security target. It is sufficient to simply state the SFRs that the TOE must meet, rather than deriving them from threats, OSPs and assumptions through security objectives.

EAL 1 provides an evaluation of the TOE as made available to the customer, including independent testing against a specification, and an examination of the guidance documentation provided. It is intended that an EAL 1 evaluation could be successfully conducted without assistance from the developer of the TOE, and for minimal outlay.

An evaluation at this level should provide evidence that the TOE functions in a manner consistent with its documentation.

Evaluation assurance level 2 (EAL2) - structurally tested

EAL2 requires the co-operation of the developer in terms of the delivery of design information and test results, but should not demand more effort on the part of the developer than is consistent with good commercial practice. As such it should not require a substantially increased investment of cost or time.

EAL2 is therefore applicable in those circumstances where developers or users require a low to moderate level of independently assured security in the absence of ready availability of the complete development record. Such a situation may arise when securing legacy systems, or where access to the developer may be limited.

Evaluation assurance level 3 (EAL3) - methodically tested and checked

EAL3 permits a conscientious developer to gain maximum assurance from positive security engineering at the design stage without substantial alteration of existing sound development practices.

EAL3 is applicable in those circumstances where developers or users require a moderate level of independently assured security, and require a thorough investigation of the TOE and its development without substantial re-engineering.

Evaluation assurance level 4 (EAL4) - methodically designed, tested, and reviewed

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

EAL4 is therefore applicable in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security-specific engineering costs.

Evaluation assurance level 5 (EAL5) - semiformally designed and tested

EAL5 permits a developer to gain maximum assurance from security engineering based upon rigorous commercial development practices supported by moderate application of specialist security engineering techniques. Such a TOE will probably be designed and developed with the intent of achieving EAL5 assurance. It is likely that the additional costs attributable to the EAL5 requirements, relative to rigorous development without the application of specialised techniques, will not be large.

EAL5 is therefore applicable in those circumstances where developers or users require a high level of independently assured security in a planned development and require a rigorous development approach without incurring unreasonable costs attributable to specialist security engineering techniques.

Evaluation assurance level 6 (EAL6) - semiformally verified design and tested

EAL6 permits developers to gain high assurance from application of security engineering techniques to a rigorous development environment in order to produce a premium TOE for protecting high value assets against significant risks.

EAL6 is therefore applicable to the development of security TOEs for application in high risk situations where the value of the protected assets justifies the additional costs.

Evaluation assurance level 7 (EAL7) - formally verified design and tested

EAL7 is applicable to the development of security TOEs for application in extremely high risk situations and/or where the high value of the assets justifies the higher costs. Practical application of EAL7 is currently limited to TOEs with tightly focused security functionality that is amenable to extensive formal analysis.

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Development	ADV_ARC		**1**	1	1	1	1	1
	ADV_FSP	**1**	**2**	**3**	**4**	**5**	5	**6**
	ADV_IMP				**1**	1	**2**	2
	ADV_INT					**2**	**3**	3
	ADV_SPM						**1**	1
	ADV_TDS		**1**	**2**	**3**	**4**	**5**	**6**
Guidance documents	AGD_OPE	**1**	1	1	1	1	1	3
	AGD_PRE	**1**	1	1	1	1	1	1
Live cycle support	ALC_CMC	**1**	**2**	**3**	**4**	4	**5**	5
	ALC_CMS	**1**	**2**	**3**	**4**	**5**	5	5
	ALC_DEL		**1**	1	1	1	1	1
	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD			**1**	1	1	1	**2**
	ALC_TAT				**1**	**2**	**3**	3
Security Target Evaluation	ASE_CCL	**1**	1	1	1	1	1	1
	ASE_ECD	**1**	1	1	1	1	1	1
	ASE_INT	**1**	1	1	1	1	1	1
	ASE_OBJ	**1**	**2**	2	2	2	2	2
	ASE_REQ	**1**	**2**	2	2	2	2	2
	ASE_SPD		**1**	1	1	1	1	1
	ASE_TSS	**1**	1	1	1	1	1	1
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**3**	3	4
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
Vulnerability Assessment	AVA_VAN	**1**	**2**	2	**3**	**4**	**5**	5

Evaluation assurance level summary

Class AVA: Vulnerability assessment

The AVA: Vulnerability assessment class addresses the possibility of exploitable vulnerabilities introduced in the development or the operation of the TOE.

Vulnerability analysis (AVA_VAN)

Vulnerability analysis is an assessment to determine whether potential vulnerabilities identified, during the evaluation of the development and anticipated operation of the TOE or by other methods (e. g. by flaw hypotheses or quantitative or statistical analysis of the security behaviour of the underlying security mechanisms), could allow attackers to violate the SFRs.

Vulnerability analysis deals with the threats that an attacker will be able to discover flaws that will allow unauthorised access to data and functionality, allow the ability to interfere with or alter the TSF, or interfere with the authorised capabilities of other users.

Part D

Evaluation Results regarding development and production environment

The IT product Smart-ID SecureZone Version version 11.5.23 has been evaluated at an approved evaluation facility using the Common Methodology for IT Security Evaluation (CEM) Version 3.1 extended by Scheme Interpretations, by advice of the Certification Body for components beyond EAL 5 and CC Supporting documents for conformance to the Common Criteria for IT Security Evaluation (CC), Version 3.1.

As a result of the TOE certification dated 2023-09-26 the following results regarding the development and production environment apply. ALC_CMS.4, ALC_DEL.1, ALC_DVS.1, ALC_LCD.1, ALC_TAT.1) are fulfilled for the development and production sites of the TOE listed below:

Name of site / Company name	Address	Type of site	Date of last audit	New audit / reused audit / n.r.
Tallinn, Estonia	SK ID Solutions AS Pärnu mnt 141, 11314 Tallinn, Estonia	TOE development (implementation and testing), TOE production and delivery initiation (TOE distribution), Development of CC evaluation evidence documentation.	2023-04-05 / 06	new audit
	Server room in data center Vae 14, Laagri Harju country, Estonia	Hosting all relevant development servers.	2023-04-05 / 06	new audit

For the site listed above, the requirements have been specifically applied in accordance with the Security Target [ST]. The evaluators verified, that the threats, security objectives and requirements for the TOE life cycle phases up to delivery as stated in the Security Target [ST] are fulfilled by the procedures of this site.

Part E

Security Target

Attached is the public version of the Security Target: “Security Target of the Smart-ID SecureZone, Author: SK ID Solutions AS
Date: 2023-08-28
Version: 3.0.8

Smart-ID SecureZone Security Target

Technical document
Version 3.0.8
August 28, 2023
121 pages

Contents

List of Figures

List of Tables

1 Introduction

1.1 Objectives and Scope of the Document

This document is the Security Target (ST) document for the Smart-ID SecureZone. The ST defines the Target of Evaluation and describes the security problem with the terms of Common Criteria.

1.2 Intended Audience

TOE users, developers, evaluators and certifiers.

1.3 Related Documents

1.3.1 Normative references

[1] *Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model*, Apr. 2017. [Online]. Available: https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf.

[2] *Common Criteria for Information Technology Security Evaluation. Part 2: Functional security components*, Apr. 2017. [Online]. Available: https://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf.

[3] *Common Criteria for Information Technology Security Evaluation. Part 3: Assurance security components*, Apr. 2017. [Online]. Available: https://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf.

[4] *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*, Aug. 2014. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.257.01.0073.01.ENG.

1.3.2 Other references

[5] A. Buldas, A. Kalu, P. Laud, and M. Oruaas, “Server-Supported RSA Signatures for Mobile Devices”, in *Computer Security – ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part I*, S. N. Foley, D. Gollmann, and E. Snekkenes, Eds. Cham: Springer International Publishing, 2017, pp. 315–333, ISBN: 978-3-319-66402-6. [Online]. Available: https://doi.org/10.1007/978-3-319-66402-6_19.

[6] *Trustworthy Systems Supporting Server Signing. Part 2: Protection Profile for QSCD for Server Signing*, EN 419 241-2:2019, Feb. 2019. [Online]. Available: https://standards.iteh.ai/catalog/standards/cen/6161a882-7bd0-4450-a2ca-bf20251d6382/en-419241-2-2019.

[7] “Smart-ID SecureZone Technical Architecture”, version 11.5, 2023.

[8] “Smart-ID Technical Architecture”, version 21.0, 2022.

[9] “Smart-ID Threshold Signature Engine Security Target”, version 3.1.0, 2023.

[10] “Administration Guide for SecureZone”, version 2.15, 2023.

[11] “Installation Guide for SecureZone”, version 2.32, 2023.

[12] “Smart-ID SecureZone monitoring guide”, version 1.6, 2023.

[13] “Signer User Guidance information for SecureZone and TSE library operators”, version 2.8, 2023.

[14] *PKCS #1: RSA Cryptography Specifications Version 2.2*, RFC 8017 (Informational), IETF, Nov. 2016. [Online]. Available: https://tools.ietf.org/html/rfc8017.

[15] *Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements*, ETSI EN 319 411-1, version 1.3.1, May 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_en/319400_319499/31941101/01.03.01_60/en_31941101v010301p.pdf.

[16] *Protection profiles for Secure Signature Creation Device — Part 2: Device with key generation*, EN 419 211:2-2013, Jul. 2013. [Online]. Available: https://standards.iteh.ai/catalog/standards/cen/b80a8253-afba-4d23-8e3e-ca1d1c8baeea/en-419211-2-2013.

[17] *Policy and security requirements for Trust Service Providers issuing certificates; Part 2: Requirements for trust service providers issuing EU qualified certificates*, ETSI EN 319 411-2, version 2.4.1, Nov. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_EN/319400_319499/31941102/02.04.01_60/en_31941102v020401p.pdf.

[18] ETSI, *Electronic Signatures and Infrastructures (ESI): Cryptographic Suites*, ETSI TS 119 312 V1.4.1, Aug. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/119300_119399/119312/01.04.01_60/ts_119312v010401p.pdf.

[19] S.-I. C. W. Group, *SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms*, Jan. 2020. [Online]. Available: https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf.

[20] *Diffie-Hellman Key Agreement Method*, RFC 2631 (Informational), IETF, Jun. 1999. [Online]. Available: https://tools.ietf.org/html/rfc2631.

[21] *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)*, RFC 3526 (Informational), IETF, May 2003. [Online]. Available: https://tools.ietf.org/html/rfc3526.

[22] *Recommendation for Key-Derivation Methods in Key-Establishment Schemes*, FIPS SP 800-56C Rev. 2, NIST, Aug. 2020. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-56c/rev-2/final.

[23] *Recommendation for Cryptographic Key Generation*, NIST Special Publication 800-133 Revision 2, NIST, Jun. 2020. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-133/rev-2/final.

[24] *Specification for the Advanced Encryption Standard (AES)*, FIPS PUB 197, NIST, Nov. 2001. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

[25] *The Keyed-Hash Message Authentication Code (HMAC)*, FIPS PUB 198-1, NIST, Jul. 2008. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf.

[26] *Secure Hash Standard (SHS)*, FIPS PUB 180-4, NIST, Aug. 2015. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

[27] *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, FIPS PUB 202, NIST, Aug. 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

1.4 Terms and Abbreviations

Notation	Description
CA	Certificate Authority – see also Certificate Service Provider (CSP).
CC	Common Criteria
CGA	Certificate Generation Application – service which allows the Signer to obtain a Qualified Certificate for Electronic Signature, which binds together the Validation Data and the Person Identification Data of the Signer, from a Qualified Trust Service Provider.
CSP	Certificate Service Provider – service, which issues the certificates binding together the SVD and identity of Signer. See also Certificate Authority (CA).
DTBS	Data To Be Signed – the document which the Signer wishes to sign. See also the asset D.DTBS.
DTBS/R	Data To Be Signed Representation – DTBS/R is generated from the Data To Be Signed (DTBS) with a hash algorithm. See also the asset D.DTBS/R.
HSM	Hardware Security Module – trusted hardware component, which provides the certified cryptographic functions.
HSM master key	Hardware Security Module master key – a root (or master) key is used to encrypt other keys, that are in turn used to encrypt the actual data you want to protect. The master key can decrypt all of the other keys, and therefore (indirectly) all of the data.
ICT	Information and Communications Technology
JakartaEE	JakartaEE – Jakarta EE, formerly Java Platform, Enterprise Edition (Java EE) and Java 2 Platform, Enterprise Edition (J2EE), is a set of specifications, extending Java SE with specifications for enterprise features such as distributed computing and web services.
JDK	Java SE Development Kit – software package, which includes the Java Virtual Machine (JVM) and the related libraries and utilities in order to run the Java applications.
JRE	Java Runtime Environment – the standard execution environment for the Java applications. See also JVM.
JVM	Java Virtual Machine – the standard execution environment for the Java applications. See also Java Runtime Environment (JRE).
JWE	JSON Web Encryption

Notation	Description
keyUUID	Key Universally Unique IDentifier – D.Signing_Key_Id is referred to as keyUUID in some places since this is the name of the attribute in the developer documents and sources. This is the unique identifier of the signing keys. It is also used to map the Signer to the signing keys.
KTK	Key Transport Key – the key which is used to encrypt cryptographic key material for transferring it from one Smart-ID system component to another component over an insecure communication channel. See also the asset D.KTK.
KWK	Key Wrapping Key – the key which is used to encrypt cryptographic key material for the purposes of secure storage of the key material. See also the asset D.KWK.
OCS	Operator card set – The smart-cards used by the nCipher Hardware Security Module (HSM), which are used to authenticate operators.
PKI	Public Key Infrastructure
QSCD	Qualified Signature Creation Device – device, which produces the qualified signatures according to the reg. (EU) 910/2014 [4].
RA	Registration Authority.
RNG	Random number generator
SAP	Signature Activation Protocol – Cryptographic protocol for activating the signing keys in the Server-Signing solutions.
SCA	Signature Creation Application
SCD	Signature Creation Data – the private key used for creating electronic signatures. See also asset D.SCD.
SCD/SVD	Cryptographic key pair with the Signature Creation Data (SCD) as the private key and Signature Verification Data (SVD) as the public key.
Signer	The natural person, who is the owner of the key pair (SCD and SVD) and who is creating the digital signatures with the key pair.
SSCD	Secure Signature Creation Device
ST	Security Target
SVD	Signature Verification Data – the public key corresponding to the SCD of a signature. SVD can be used to verify the signature. See also the asset D.SVD.
TEK	Transport Encryption Key – An AES-256 symmetric cryptographic key shared between the TOE and a specific instance of TSE. It is used to protect the communication between the TSE instance and SecureZone. TEK is created per key pair and has the same life cycle as the key pair SCD/SVD.
TOE	Target of Evaluation
TOE environment	The IT and computing environment in which the TOE is deployed and operated.
TSE	Threshold Signature Engine – Smart-ID App TSE is the software component, which works within the Signer's environment and helps and assists Signer to follow the Threshold Signature Scheme Protocol (TSSP) and to use the Smart-ID SecureZone services for the key enrolment and signature creation.
TSF	TOE Security Functions
TSFI	TOE Security Functionality Interface – The interface over which the TOE Security Functionality can be accessed and used or over which the data flows in either direction.
TSP	Trust Service Provider

Notation	Description
TSSP	Threshold Signature Scheme Protocol – cryptographic protocol and algorithms followed by the Signer and the TOE in order to generate the distributed key pair of the Signer and later use that key pair for producing signatures of the Signer. The TSSP is defined in the peer-review published article [5].
UML	Uniform Modelling Language
VAD	Verification Authentication Data – signer's VAD is the data, which is input by the Signer in order to authenticate himself. Usually this is the PIN code of the Signer.
VM	Virtual Machine

1.5 ST Reference Identification

Title: Smart-ID SecureZone Security Target
Version: 3.0.8
Publication date: August 28, 2023

1.6 TOE Reference Identification

TOE identification/version: Smart-ID SecureZone version 11.5.23

1.7 Document changelog

Version	Date	Summary of changes
1.0.0	15.05.2017	First submission to the evaluation process
1.0.1	24.05.2017	1. Update to the CC version 3.1, release 5 2. Addition of the A.PRIVILEGED_USER 3. Improved definitions of the SFRs in the section 6.1

Version	Date	Summary of changes
1.0.2	01.06.2017	1. Complemented Smart-ID system description in chapter 2 and fixed the product name spelling. 2. Added the description of the re-key functionality to the TOE with the corresponding SFPs in the section 6.1.2 and SF in the section 7.5.2. 3. Added the section 6.2 "Security Assurance Requirements" to the chapter 6. 4. Added the section 6.3 "SFR Dependencies Rationale" to the chapter 6. 5. Deleted the "Requirement Rationale" table from the chapter 6 because the table is no longer useful and the SFRs have the proper definitions in the section 6.1. 6. Clarified the user authentication and the roles in the TOE within the sections 6.1 and chapter 7.
1.0.3	15.06.2017	Fixed the problems outlined on the "Observation Report V1", observations 1 to 32. Detailed list of individual changes are listed in the response to the report.
1.0.4	05.07.2017	1. Fixed the problems outlined on the "Observation Report V1", observations 33 to 51. Detailed list of individual changes are listed in the response to the report. 2. Fixed the typos and problems outlined by SK.
2.0.0	19.01.2018	Rewrite of the document to be more similar with the concepts of PP 419 241-2 [6].
2.1.0	02.03.2018	Fixed the problems outlined on the "Observation Report V3". Detailed list of individual changes are listed in the response to the report. Improved the document according to TÜViT's feedback from the meeting on February 8th 2018.
2.2.0	16.03.2018	Fixed the problems outlined on the "Observation Report V4". Detailed list of individual changes are listed in the response to the report.
2.3.0	21.05.2018	Fixed the problems outlined in the Observation Reports V5 and V6. Detailed list of individual changes are listed in the response to the reports.

Version	Date	Summary of changes
2.4.0	22.06.2018	Fixed the problems outlined in the Observation Report V7. Detailed list of individual changes are listed in the response to the report. Updated the SecureZone reference version number to v10.3
2.5.0	30.07.2018	Fixed the problems outlined in the SZ ST Observation Report V8 and SZ AGD Observation Report V4. Detailed list of individual changes are listed in the response to the report. Removed the irrelevant reference to TSE in section 2.4.1.
2.6.0	19.09.2018	Fixes of the supported KWK, KTK and SHA-2 key sizes, according to ADV_IMP OR. Updated the database server version number in section 2.4.4 that was used during testing.
2.7.0	21.09.2018	Fixed the list of SFRs in section 8.1.1.2 SF.AccessControl, under point 2. Fixed the version number of the TOE as 10.3.5.
2.8.0	07.09.2021	Added nShield Connect XC HSM to list of tested HSMs, renamed Thales nShield to nCipher nShield, removed nonsensical compatibility sentence. Sections 2.4.5 and 2.5.4.
2.8.1	21.09.2021	Fixed name of nCipher nShield Connect XC HSM. Section 2.4.5.
2.8.2	28.09.2021	XC HSM idenfitied now as in CC certification report. Section 2.4.5.
2.8.3	12.11.2021	Updated Section 2.4.5 and 2.5.4
2.8.4	16.11.2021	Updated Section 2.4.5
2.8.5	25.01.2022	Updated 2.8.0 changelog, chapters 1.3.2, 1.6, 2.4.5
3.0.0	24.05.2022	Updated Chapters 4, 5, 7, and 8 for improved readability.
	25.05.2022	Re-organized Section 7.4 to a more readable format. Improved the explanations for fulfilling the security objectives.
	03.06.2022	Removed attribute "expiration_time" from the list of security attributes in the TOE (Section 7.1.2.2), since there is no such attribute in the TOE.
	07.06.2022	Corrected HSM model name in Section 2.5.4. Clarified non authenticated function in Section 2.5.3.1. Updated description on JakartaEE. Changed RSA key sizes from 2048 to 3072 where applicable. Sections 7.3.2.1.2, 7.3.2.3.1, 7.3.2.3.2.
	08.06.2022	Changed packaging of TOE from WAR to JAR. Section 2.5.1 Changed and simplified requirements for OS, hardware, Java, PostgreSQL. Section 2.4. Improved the description of the auditing functionality. Sections 2.2.3.5, 2.5.3.1, 8.1.1.3.

Version	Date	Summary of changes
		Fixed name of OE.DTBS_Intend and OE.TSE.DTBS_Intend in Section 5.5.2.1.8, was spelled OE.DTBS_Intended and OE.TSE.DTBS_Intended by mistake.
	09.06.2022	Fixed incorrect references to [15], clauses 6.3.3 and 6.3.1 (in Section 4.4.1.3).
	10.06.2022	In Section 4.1 Assets, removed references to security requirement "authenticity", since the notion of "integrity" already compasses "authenticity".
	15.06.2022	In Section 8.1.2.1, removed incorrect claim that Bouncy Castle module is used for generation of DEK AES key. Removed asset D.VC as it is outside the scope of the TOE along with other methods of tying sessions between SCA and TSE. Sections 2.3.3.2, 4.1, 4.4.5 and 5.5.2.1.8.
	16.06.2022	Modified FIA_AFL.1 (Section 7.3.4.1). Timelocking mechanism is now configurable by TOE administrator.
	17.06.2022	Changed definition of TSF_CONFIG_DATA (Section 4.1 Assets) to more accurately represent real configuration.
	21.06.2022	Added new TEK DH algorithm, Sections 7.3.2.1.3 and 8.1.2.1.
	21.06.2022	Added RSASSA-PSS, Sections 7.3.2.3.1 and 7.3.2.3.2.
	27.06.2022	Reviewed and updated references to published versions of EN 419 241-2:2019 and EN 419 211-2:2013.
	28.06.2022	Corrected explanation of threat T.Signature_Request_Disclosure mitigation (Section 5.5.2.1.14). Fixed Table 5 correspondingly. Unified the naming conventions for some security objectives and threats (in Section 5.4). Fixed definition of FMT_SMF.1.1.
	29.06.2022	Modified explanations for fulfulling security objectives OT.Sig_Secure, OT.SCD/SVD_Corresp, OT.TSSP_CloneDetection, OT.Privileged_User_Management, OT.Privileged_User_Authentication, and OT.Privileged_User_Protection (in Section 7.4). Modified mapping between security objectives and SFRs correspondingly (table 25).
	30.06.2022	Updated references to Smart-ID architecture documents and manuals.
	01.07.2022	Removed asset D.VC from the list of assets (see comment from 15.06.2022).
	15.07.2022	Added definitions for JWE and RNG. Renamed SFR FCS_COP.1/SHA-2 (Section 7.3.2.3.5) to FCS_COP.1/SHA and modified it to consist of two sub-items: FCS_COP.1.1/SHA-2 and FCS_COP.1.2/SHA-3. Changed the possible key sizes for RSA shares to have allowed values of 3k..8k (Section 7.3.2.3.1) and the corresponding RSA compound key to allow for 6k..16k (Section 7.3.2.1.1) Modified FCS_COP.1.1/AES (Section 7.3.2.3.3) to allow for key sizes that are larger than 128 bits. Added Section 2.3.3.3 Session tying methods.

Version	Date	Summary of changes
		Modified paragraph Fulfilling OT.TSSP_CloneDetection (Section 7.4.2) to employ SFRs FDP_ACC.1/Anonymous and FDP_ACF.1/Anonymous. Modified FMT_MTD.1 (Section 7.3.5.2.1) to be in correspondence with the SFR FMT_MTD.1 of PP 419 241-2 [6]. Added related security objective OT.System_Protection from PP 419 241-2 [6] to the current document (Section 5.1.11). Linked OT.System_Protection to mitigation of the threat T.Context_Alteration (Section 5.5.2.1.13).
3.0.1	16.09.2022	Fixed the issues outlined in the "Observation Report V1": In Table 27, the assurance class component AVA_VAN.5 was added; SFRs FCS_COP.1.1/SHA-2 and FCS_COP.1.1/SHA-3 are now listed correctly in the corresponding sections; SFRs FCS_CKM.1/DH_TEK and FCS_CKM.1/DH_TEK_EC are now listed correctly in the corresponding sections. For clarity, Application Note 1 was added to explain why assets R.SAD, R.Reference_Signer_Authentication_Data, and R.Authorisation_Data coincide in TSSP. Correspondingly, the three terms were merged to SAD. Changed maximum allowed key size for KTK to be 8k (Section 7.3.2.1.2).
3.0.2	10.01.2023	Added D.signatureParameters asset (for realization of RSA-PSS support) Removed OE.TSE.DTBS_Intend to prepare for planned changes in TSE ST document: 1) both SZ and TSE ST documents will contain the same SFR, OE.DTBS_Intend, with identical wordings; 2) OE.DTBS_Intend will cover both manual and automatic verification methods. Changed A.SIGNER_DEVICE to also refer to OE.DTBS_Intend (since OE.TSE.DTBS_Intend was removed). Removed SFR FCS_CKM.1/DH_TEK_EC, since the corresponding feature was not implemented. Modified SFR FCS_CKM.1/DH_TEK to not mention key sizes above 4k, since they were not implemented. Added information about methods for deletion of KTK, KWK, DEK and unused server shares. Updated FAU_GEN.1 to contain the list of the Audit log categories logged by the TOE. Section 7.1.2.2 has been updated to contain information that KWK has HMAC portion. Updated reference SP 800-56A to SP 800-56C to reflect the fact that concatKDF definition had moved to another specification.
3.0.3	23.01.2023	Expanded definition of D.SignatureParameters.

Version	Date	Summary of changes
		Updated FAU_GEN.1 to contain a missing audit log category (configuration initialization). Updated Section 2.4.5 to correctly reference the Connect 6000+ HSM.
3.0.4	15.02.2023	Unified descriptions of key pair generation and signing processes in SZ and TSE ST docs. Clarified definitions of some assets (D.DTBSR, D.SCD, D.Signing_Key_Id, D.SVD). Updated diagrams in Chapter 2 so that SZ and TSE have exactly the same diagrams for the process documentation. Modified SFR FCS_CKM.4 and SF.KeyZer to more accurately reflect the implementation.
3.0.5	10.03.2023	Unified descriptions of session tying methods in SZ and TSE ST docs.
3.0.6	18.05.2023	Removed [7] from physical scope, Section 2.5.1. Updated references.
3.0.7	17.08.2023	Fixed the version number of the TOE as 11.5.x. Updated referenced document versions.
3.0.8	25.08.2023	Fixed the version number of the TOE as 11.5.23. Updated the version number of the referenced AGD Installation guide.

2 System Overview

This chapter provides an informal overview of the digital signatures, Smart-ID Threshold Signature Scheme Protocol and the Smart-ID SecureZone as the TOE of this ST document. The formal Security Problem Definition using the CC terms, is given in the next chapters. However, where appropriate, references are made to the definitions in the following sections of the document.

2.1 Introduction to the Smart-ID system

The invention of the digital signatures and the Public Key Infrastructure (PKI) has enabled society to use convenient authentication and signature features. For example, when digital signature technology is combined with the smart-cards, the secure storage of the private keys can be implemented. Together with the PKI technology, the Secure Signature Creation Devices (SSCDs) have been developed by the Information and Communications Technology (ICT) industry. With such solution, the protection of the Signer's private key is handled by the Signer itself. However, as the features of the personal computing devices have been evolved, the usage of such special purpose devices has become more and more inconvenient. The ICT industry has been searching for alternative solutions. One of such solutions is the server-signing services, where the protection of the private key of the Signer is entrusted to the server-signing service provider.

The Smart-ID system has been developed to provide alternative solution for the digital signature creation, where the risk and responsibility to secure the private key is no longer placed to any single system participant, but is shared between multiple system components. With the application of the cryptographic threshold signature protocols, the private key can be generated in shares. In order to use the private key to create the digital signatures, the shares don't need to be combined in a single physical location. Instead, the individual shares are used to create the shares of the signature. Only when all shares of the signature are combined, the compound signature is achieved. With such kind of a protocol, the overall risks and technical threats can be greatly reduced.

The current document describes the Smart-ID TSSP and the Smart-ID SecureZone, which is the server-side implementation of this protocol and the Target of Evaluation (TOE) of this ST document.

2.2 Overview of the TOE

This section describes the TOE and explains its intended usage.

The TOE described in this Security Target (ST) is inspired by but is not strictly conformant to the PP 419 241-2 [6]. The reason for non-strict conformance is due to differences in the underlying technical solutions of the TOE and classical server signature solutions. The differences come from the TOE's usage of the Smart-ID TSSP ([5]). Otherwise, the same terminology and methodology as in the protection profile is used in the current ST document for describing the TOE. There are some informative references to the comparable assets and threats to the PP 419 241-2 [6] for the purpose of quicker grasp of the ST document and straightforward comparison.

2.2.1 TOE definition

The TOE is the computer software product "Smart-ID SecureZone". It is a Java application server package, which implements the server-side functions of the TSSP for the Signer and the management functions for the administrators. The Signer, who follows the client-side functions of the TSSP, can use the TOE services to enroll new key pairs, create digital signatures and to destroy the key pairs.

The important distinction here is that the TOE alone doesn't create the whole digital signature on behalf of the Signer, but they both participate in the cryptographic protocol. The TSSP is further explained in the section 2.3.

2.2.2 TOE type

The TOE is a software component, which implements the server-side functions of the Threshold Signature Scheme Protocol TSSP to activate a signature. It is deployed in a dedicated tamper protected environment, that is connected to the Hardware Security Module (HSM) via a trusted channel. It uses the Signature Activation Data (SAD) from the signer to complete the signature computation with the HSM.

Together, the following form a Qualified Signature Creation Device (QSCD): the mobile client, the TOE, and the HSM.

2.2.3 TOE usage and major security features

The TOE is intended to be used as a component of a QSCD system to conduct the following high-level functions:

1. Creation of Qualified Electronic Signatures, complying with eIDAS regulation reg. (EU) 910/2014 [4];
2. Enrolment and destruction of the Signer's key pair;
3. Security management and access control functions.

The high-level security features of the TOE are similar to the high-level security features of traditional QSCDs. The features are grouped into the following subsections, categorized according to the abovementioned main usage functions of the TOE.

2.2.3.1 Enrolment of the components of the new key pair of the Signer and other keys

1. Import of the server's part of the private key of the Signer. This is the asset D.serverPart.
2. Usage of the HSM to generate the server's share of the private key of the Signer. This is the asset D.serverShare.
3. Generation of the compound public key of the Signer. This is the asset D.SVD.
4. Generation of D.KTK RSA key (by HSM) for encrypting transferred keys between the TSE and TOE.
5. Generation of D.TEK (by TOE) to protect the communication between the TSE and TOE.
6. Generation of D.KWK AES key (by HSM) for wrapping key material in the TOE database.

7. Generation of D.DEK AES key (by TOE) to encrypt certain database fields.

More details can be found in the section 2.5.3.2 System Overview

2.2.3.2 Signature creation

1. Creation of the server's part of the signature of the Signer. This is the asset D.server SignaturePart.
2. Creation and validation of the applicationSignatureShare of the Signer from the D.applicationSignaturePart and the D.serverSignaturePart. This is the asset D.applicationSignatureShare.
3. Usage of the HSM to create the server's share of the signature of the Signer. This is the asset D.serverSignatureShare.
4. Creation and validation of the compound signature of the Signer. This is the asset D.signature.

2.2.3.3 Destroying the components of the key pair of the Signer and other keys

1. Destroying the shares of the private key of the Signer, the assets D.serverPart and D.serverShare.
2. Destroying a batch of unused D.serverShare assets.
3. Destroying D.KTK key.
4. Destroying D.KWK key.
5. Destroying D.DEK key.

2.2.3.4 Security management functions

The TOE also has the following management features:

1. Starting the TOE instance and securely connecting to the HSM to load the encryption keys for the TOE database.
2. Generation of D.KWK, D.KTK and D.DEK encryption keys.
3. Batch pre-generation of D.serverShare assets (for performance reasons).
4. Destruction of D.KWK, D.KTK and D.DEK encryption keys.
5. Destruction of unused D.serverShare assets.
6. Re-key process initiated by the CA that enables generating new key-pair and the corresponding certificate for an existing Signer (see also table 13). The TOE is involved in the process by:

   6.1 re-generating and rewriting the D.serverShare of the Signer's private key;

   6.2 re-generating and returning the new Signer's compound public key D.SVD.

2.2.3.5 Authentication, Access Control and Security audit generation functions

The TOE also has the following security features:

1. Authentication – This function provides different methods to authenticate users and protect the assets of the TOE.
2. Access control – Different users have access to their different assets and allowed operations.

3. Security audit generation – The audit records of the important system events are generated by the TOE and saved to its database to be exported to an external system.

More details about these connections can be found in the section 2.5.3.1 System Overview

2.2.3.6 Protecting communication with external components

The TOE uses an (encrypted) trusted path to communicate with the Smart-ID App TSE. On the other hand, the TOE communicates with HSM and Database via a vendor-specific secure channel.

More details about them can be found in the section 2.5.3.3 System Overview

2.2.3.7 Functions not present in the TOE

The TOE only provides the key pair related security functions and it doesn't have any features related to the identity proofing, Signer registration, certificate issuing and other features, which are commonly required by the full-scale PKI system. So, in order to establish the larger PKI system, the TOE will interface and work with the following external trusted IT systems:

1. Registration Authority (RA) is responsible for identity proofing of the Signer. The RA will use either existing digital identities of the Signer or will perform the identity proofing procedures to verify the government issued identity document in person. The RA will then forward this information to the CA, so that CA can issue the certificate for binding together the identity and the D.SVD of the Signer.
2. CA is responsible for issuing qualified certificates to the Signer. CA will receive the identity information from the RA and the D.SVD from the TOE.

The TOE places certain requirements to the security level of such functions that are provided by external trusted IT systems. For example, the TOE requires that the CA issues qualified certificates.

2.3 Threshold Signature Scheme Protocol (TSSP)

2.3.1 Introduction

The TSSP is the protocol to be followed by the Signer and the TOE, in order to generate the key pair of the Signer (the assets D.SCD and D.SVD), which is usable only when Signer, Smart-ID App TSE and the TOE are participating in the protocol. The private key of the key pair of the Signer (the asset D.SCD) is generated in shares. It is done in such a way, that multiple shares of the private key (the assets D.clientPart, D.serverPart, D.serverShare) are separately generated and they are independently protected by the Signer and TSE (the asset D.clientPart) and the TOE (the assets D.serverPart and D.serverShare).

In order to actually create the digital signature of the Signer, those individual shares of the private key have to be used by their respective holders to create the shares of the signature (the corresponding assets D.applicationSignaturePart, D.serverSignaturePart, D.serverSignatureShare). Those shares of the signature must then be combined and the resulting compound signature (the asset D.signature) is then verifiable with the public key of the Signer (the asset D.SVD).

The TSSP is fulfilling the same kind of purposes as the Signature Activation Protocol (SAP) from the PP 419 241-2 [6] and provides the same security capabilities and in some way, TSSP can be seen as an instance of the SAP. However, the TSSP also includes the key pair enrolment protocol and provides additional unique security capabilities to the Signer and the TOE. Therefore, we refer to the TSSP in this ST document.

The next sections give a high-level abstract overview how the TSSP works between the Signer, Smart-ID App TSE, and the TOE. Note that some technical details are omitted and simplified from these sections, in order to keep the description as short as possible. Please refer to the peer-reviewed paper [5] in order to get all the mathematical and cryptographical details along with the security proofs. Also, please refer to the architecture documents [8] and [7] in order to get all the implementation details of the TOE.

2.3.2 Key pair generation process

The high-level process for the key pair generation of the Signer is shown in the Figure 1 with the UML sequence diagram. In the following sections, the components and messages shown in the diagram are explained.

2.3.2.1 Actors and components

- Signer – This is the natural person, who is using the Smart-ID App Threshold Signature Engine (TSE) and the TOE services to generate, protect and use the key pair, which is split into multiple shares according to the TSSP. Signer keeps the knowledge-based secret asset D.PIN.
- Smart-ID App TSE – This is the software component, which is running on the personal mobile device of the Signer (phone, tablet or other smart-device). The mobile device is under the Signer's control and is helping Signer to generate the app's share of the key pair and to protect it. The Smart-ID App TSE implements the client-side functions of the TSSP. The security functions of the Smart-ID App TSE are evaluated according to the separate ST document [9].
- Smart-ID SecureZone (TOE) – This is the software component, which is the TOE of the current Security Target document. The TOE implements the server-side functions of the TSSP. The TOE allows Signer to generate, protect and use the key pair, which is split into multiple shares according to the TSSP.
- Smart-ID SecureZone database – This is the database, which is used by the TOE to store user data and TSF data. Sensitive security attributes are stored with HSM proprietary encryption or with TOE implemented encryption.
- Smart-ID SecureZone HSM – This is the trusted hardware component, which is providing the certified cryptographic functions to the TOE, such as key share generation and creation of the signature share.

2.3.2.2 Process steps

The high-level key pair generation process is shown in the Figure 1 with a UML sequence diagram.

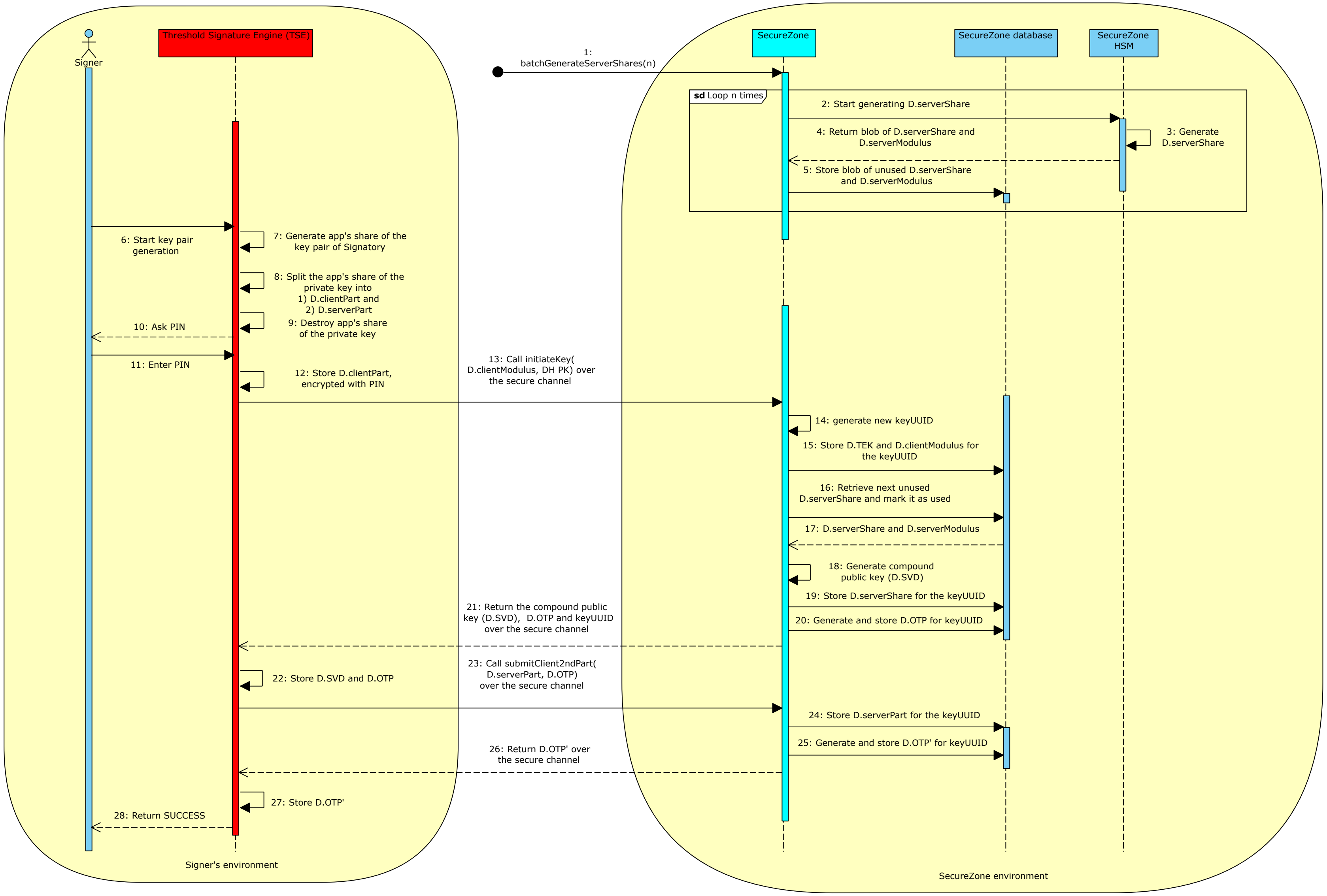

Signer
Threshold Signature Engine (TSE)
SecureZone
SecureZone database
SecureZone HSM
1: batchGenerateServerShares(n)
sd Loop n times
2: Start generating D.serverShare
3: Generate D.serverShare
4: Return blob of D.serverShare and D.serverModulus
5: Store blob of unused D.serverShare and D.serverModulus
6: Start key pair generation
7: Generate app's share of the key pair of Signatory
8: Split the app's share of the private key into 1) D.clientPart and 2) D.serverPart
9: Destroy app's share of the private key
10: Ask PIN
11: Enter PIN
12: Store D.clientPart, encrypted with PIN
13: Call initiateKey(D.clientModulus, DH PK) over the secure channel
14: generate new keyUUID
15: Store D.TEK and D.clientModulus for the keyUUID
16: Retrieve next unused D.serverShare and mark it as used
17: D.serverShare and D.serverModulus
18: Generate compound public key (D.SVD)
19: Store D.serverShare for the keyUUID
20: Generate and store D.OTP for keyUUID
21: Return the compound public key (D.SVD), D.OTP and keyUUID over the secure channel
22: Store D.SVD and D.OTP
23: Call submitClient2ndPart(D.serverPart, D.OTP) over the secure channel
24: Store D.serverPart for the keyUUID
25: Generate and store D.OTP' for keyUUID
26: Return D.OTP' over the secure channel
27: Store D.OTP'
28: Return SUCCESS
Signer's environment
SecureZone environment

Figure 1. Overview of the enrollment procedure in the TSSP.

TSSP key pair generation steps (the numbers correspond to the messages on the sequence diagram):

1. SZ operator asks SZ to pre-generate the server's shares, so that registration of new Signers is quicker.
2. SZ asks HSM to generate the new server's share of the key pair (D.serverShare).
3. HSM generates the new server's share of the key pair (D.serverShare and D.serverModulus).
4. SZ receives the encrypted blob of the private key (D.serverShare) and the public key of the key pair (D.serverModulus).
5. SZ stores the private key (D.serverShare) and the public key of the key pair (D.serverModulus) in the SZ database and marks them free to be used. The private key is stored and transferred encrypted.
6. Signer asks the Smart-ID App TSE to start generating the new key pair.
7. TSE generates the app's share of the key pair (D.clientShare/D.clientModulus). The key pair consists of the private key (D.clientShare) and the public key (the asset D.clientModulus). TSE generates an ephemeral Diffie-Hellman key pair for D.TEK establishment.
8. TSE mathematically splits the private key (D.clientShare) of the generated key pair into two parts (D.clientPart/D.serverPart), using an additive sharing method. The individual parts cannot be used to deduce information about the whole private key.

   8.1 D.clientPart is the part, which is stored and protected within the TSE

   8.2 D.serverPart is the part, which is to be transmitted to the SZ
9. TSE securely destroys the private key D.clientShare of the generated key pair.
10. TSE asks Signer to enter the D.PIN to derive the encryption key, which is used to encrypt the locally stored D.clientPart. The D.PIN is the knowledge-based factor, which is used to secure the TSSP.
11. Signer enters the D.PIN.
12. TSE uses the D.PIN to derive the encryption key and to encrypt the D.clientPart. The encryption is done in a way that no validation information about the cryptogram is stored. The D.PIN itself is not stored within the Smart-ID App TSE.
13. TSE initiates the initiateKey() operation (see also table 13) in the SZ, transmitting the D.clientModulus and the Diffie-Hellman public key (for establishing the D.TEK) to the SZ.
14. SZ receives the D.clientModulus and the client's Diffie-Hellman public key. SZ assigns a fresh unique D.Signing_Key_Id (also referred to as keyUUID) to the new key pair of the Signer, executes the server-side steps of Diffie-Hellmann key exchange and generates D.TEK.
15. SZ stores D.clientModulus and D.TEK in the database.
16. SZ marks the next unused D.serverShare and D.serverModulus as used and retrieves them from database.
17. SZ receives the D.serverShare and D.serverModulus from database.
18. SZ generates the compound public key (D.SVD) by mod-multiplying together D.clientModulus and D.serverModulus
19. SZ stores the D.SVD in the database.

20. SZ generates the one-time password (D.OTP) and stores it in the database.
21. SZ returns the D.SVD, D.OTP, D.Signing_Key_Id and Diffie-Hellmann key exchange material over the secure channel to the Smart-ID App TSE. The channel is secured by encrypting the data with the newly generated D.TEK.
22. TSE decrypts the response by using the D.TEK, verifies the Diffie-Hellmann key exchange material and stores the D.SVD and D.OTP.
23. TSE initiates the submitClient2ndPart() operation (see also table 13) in the SZ, by transmitting the D.serverPart and D.OTP over the secure channel to the SZ.
24. SZ stores the D.serverPart in the database.
25. SZ generates the new value of one-time password (D.OTP) (OTP') and stores it in the database.
26. SZ returns the new value of one-time password (D.OTP) (OTP') over the secure channel to the Smart-ID App TSE.
27. TSE decrypts the response by using the D.TEK and stores the new value of one-time password D.OTP.
28. TSE Smart-ID App returns the Signer the success message about the new key pair generation.

2.3.3 Signature generation process

The high-level signature creation process is shown in the Figure 2 with a UML sequence diagram. The process involves additional component, Signature Creation Application (Signature Creation Application (SCA)). The SCA is the general purpose trusted software application, which is used by the Signer in order to prepare and to create the digitally signed documents. Such features are not included in the Smart-ID App TSE or the TOE itself, in a similar way as the function for creation of digital documents are not included in the QSCDs.

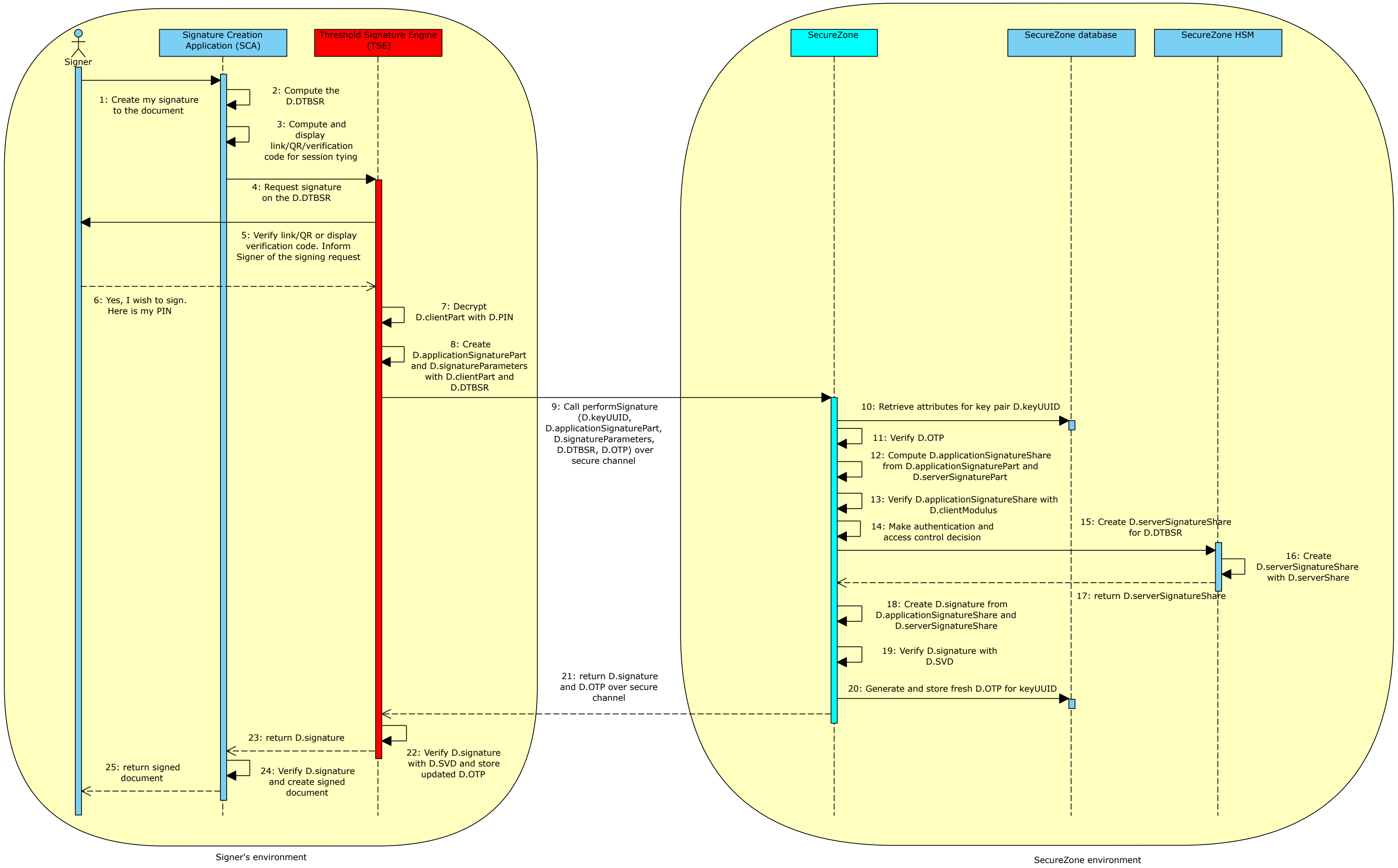

Signer
Signature Creation Application (SCA)
Threshold Signature Engine (TSE)
SecureZone
SecureZone database
SecureZone HSM
1: Create my signature to the document
2: Compute the D.DTBSR
3: Compute and display link/QR/verification code for session tying
4: Request signature on the D.DTBSR
5: Verify link/QR or display verification code. Inform Signer of the signing request
6: Yes, I wish to sign. Here is my PIN
7: Decrypt D.clientPart with D.PIN
8: Create D.applicationSignaturePart and D.signatureParameters with D.clientPart and D.DTBSR
9: Call performSignature (D.keyUUID, D.applicationSignaturePart, D.signatureParameters, D.DTBSR, D.OTP) over secure channel
10: Retrieve attributes for key pair D.keyUUID
11: Verify D.OTP
12: Compute D.applicationSignatureShare from D.applicationSignaturePart and D.serverSignaturePart
13: Verify D.applicationSignatureShare with D.clientModulus
14: Make authentication and access control decision
15: Create D.serverSignatureShare for D.DTBSR
16: Create D.serverSignatureShare with D.serverShare
17: return D.serverSignatureShare
18: Create D.signature from D.applicationSignatureShare and D.serverSignatureShare
19: Verify D.signature with D.SVD
20: Generate and store fresh D.OTP for keyUUID
21: return D.signature and D.OTP over secure channel
22: Verify D.signature with D.SVD and store updated D.OTP
23: return D.signature
24: Verify D.signature and create signed document
25: return signed document
Signer's environment
SecureZone environment

Figure 2. Overview of the signing procedure in the TSSP.

2.3.3.1 Actors and components

- Signer – This is the natural person, who is using the SCA, the Smart-ID App TSE, and the TOE services to digitally sign the document.
- Signature Creation Application – This is the general purpose trusted software application, which is handling the technical issues with creating the well-formatted and -encoded digital documents, computing the Data To Be Signed Representation (DTBS/R) and requesting the digital signature of the DTBS/R from the Signer.
- Smart-ID App TSE – This is the trusted software component, which is installed on the personal mobile device of the Signer (phone, tablet or other smart-device). The mobile device is under the Signer's control and is helping Signer to use the app's share of the key pair and to create the application's part of the signature. The Smart-ID App TSE implements the client-side functions of the TSSP. The security functions of the Smart-ID App TSE are evaluated according to the separate ST document [9].
- Smart-ID SecureZone (TOE) – This is the software component, which is the TOE of the current Security Target document. The TOE implements the server-side functions of the TSSP. The TOE allows Signer to use those parts/shares of the key pair which are stored in the TOE and thereby create the server's part of the signature and the compound signature.
- Smart-ID SecureZone database – This is the database, which is used by the TOE to store user data and TSF data. Sensitive security attributes are stored with HSM proprietary encryption or with TOE implemented encryption.
- Smart-ID SecureZone HSM – This is the trusted hardware component, which is providing the certified cryptographic functions to the TOE, such as key share generation and creation of the signature share.

2.3.3.2 Process steps

TSSP signature creation steps (the numbers correspond to the messages on the sequence diagram):

1. Signer asks the SCA to digitally sign the document.
2. SCA formats and encodes the document and computes the D.DTBS/R, which corresponds to the data to be signed.
3. SCA prepares a method of tying the signature creation session in the SCA application with the session in the Smart-ID App TSE (see Section 2.3.3.3 below for more details). This allows the Signer to be sure that he is agreeing with the correct signature request on the Smart-ID App TSE. SCA displays the session tying method to the Signer and asks to either click, scan or verify it as required.
4. At the same time, SCA requests the signature of the D.DTBS/R from the Smart-ID App TSE via the TOE.
5. Smart-ID App TSE informs the Signer of the new signing request and performs the necessary checks to verify that the signing session is the one intended by the Signer.
6. Signer verifies the displayed transaction details and agrees with the request. Signer enters the D.PIN to the TSE.
7. TSE uses the D.PIN to decrypt the D.clientPart. Note that the TSE does not verify if the entered D.PIN is correct or if the decrypted D.clientPart is valid or correct. There is no way for the TSE to validate the entered D.PIN locally, without contacting the TOE.

8. TSE uses the decrypted D.clientPart to create the signature share D.applicationSignature Part with the D.DTBS/R, and the D.signatureParameters required for creation of the signature share.
9. TSE initiates the performSignature() operation (see also table 13) in the TOE over the secure channel (by encrypting the data with D.TEK key), along with the following data: D.Signing_Key_Id, D.applicationSignaturePart, D.signatureParameters, D.DTBS/R, D.OTP.
10. TOE retrieves attributes from the database for the key pair identified by D.Signing_Key_Id.
11. TOE verifies that clone-detection token (D.OTP) for that particular D.Signing_Key_Id is valid. This gives us the possession-based authentication factor.
12. TOE uses the D.serverPart to create the signature share D.serverSignaturePart with the D.DTBS/R and D.signatureParameters and then uses the signature parts D.application SignaturePart and D.serverSignaturePart to create the signature share D.application SignatureShare.
13. TOE verifies if the signature share D.applicationSignatureShare is valid, with the D.clientModulus.
14. TOE makes the authentication and access control decision. If the signature share is not valid, the signature completion request is cancelled. This gives use to the knowledge-based authentication factor since the Signer had to use the correct D.PIN to decrypt the local D.clientPart.
15. TOE sends the D.DTBS/R to the HSM and asks for the creation of signature share with the D.serverShare.
16. HSM creates the signature share D.serverSignatureShare.
17. TOE receives the D.serverSignatureShare and verifies it with D.serverModulus and D.DTBS/R.
18. TOE creates the compound signature D.signature from the signature shares D.application SignatureShare and D.serverSignatureShare.
19. TOE verifies if the compound signature D.signature is valid and matches with the D.DTBS/R and D.SVD.
20. TOE generates a fresh clone-detection token (D.OTP) and stores it in the database.
21. TOE returns the compound signature D.signature and updated D.OTP to the TSE over the secure channel (by encrypting the data with the specific instance of D.TEK).
22. TSE decrypts the response by using the D.TEK key and receives D.signature and D.OTP. TSE verifies whether the D.signature is valid and matches with D.DTBS/R and D.SVD.
23. TSE returns the compound signature D.signature to the SCA and displays a notification to the Signer that the signature has been created successfully.
24. SCA receives the compound signature D.signature and verifies whether it is valid and matches with D.DTBS/R and D.SVD. It then creates the digitally signed document with the Signer's signature.
25. SCA returns the digitally signed document to the Signer.

2.3.3.3 Session tying methods

For security purposes, it is important to guarantee that the signature creation session in the SCA application matches with the session in Smart-ID App. There exist different methods for ensuring this, for example:

1. Calculation of a verification code from D.DTBS/R, which is displayed to the Signer both in the SCA and Smart-ID App.
2. Generation of a cryptographically secured token that is transported to the Smart-ID App via a URI link or a QR code.

Usage of a such method allows the Signer to be sure that the signature is given for the correct D.DTBS/R.

2.4 Required non-TOE hardware/software/firmware

This section lists the hardware and software components, which are required in order to successfully and securely run the TOE. In the previous sections, we have explained the major security functions of the TOE and described how the TSSP works. Note that the external components described on the sequence diagrams in the sections 2.3.2 and 2.3.3, such as the SCA, are not strictly required to operate the TOE and therefore they are not listed in this section.

2.4.1 Smart-ID App Threshold Signature Engine

The TOE must be used in conjunction with the Smart-ID Threshold Signature Engine TSE software library, or another application fulfilling the requirements set in the ST document [9], and evaluated to correspond with the EAL2 level, according to Common Criteria Part 3 [3].

2.4.2 Server hardware and operating system

The TOE is independent of hardware and operating system, the TOE security functions do not depend on the security functions of the underlying operating system.

The TOE has been tested with 64 bit Centos 7 Linux and is compatible with any contemporary Linux based operating environment, including Linux based container solutions.

The server hardware must be capable enough to run the operating system and the JVM along with the TOE software image.

2.4.3 Java Virtual Machine and related libraries

The TOE requires a JRE and a JRE compatible web application server to provide its functions.

The TOE has been tested with OpenJDK 17, and is compatible with any up-to-date version of JRE declaring compatibility with Java 17.

The TOE runs within the application server provided by Spring Boot Framework. When JVM process is started, then Spring Boot starts up a web application server and deploys TOE application into it. All network requests, which are accepted by the web application server, are routed directly into TOE. Spring Boot provides some application servers like Apache Tomcat, Eclipse Jetty and Undertow, which have their specific strengths. The TOE does not depend on any specific features of these application servers and may run within any server provided by Spring Boot project. The TOE has been tested with a default configuration, which uses Apache Tomcat embedded web application server.

The TOE uses SLF4J API (Simple Logging Facade for Java) for logging application messages (regular log messages, not audit log messages). SLF4J provides a solid and standard set of APIs, which are used to reduce the coupling between an application and any particular logging framework. The TOE depends only on the SLF4J API, which redirects all logged messages to some configured logging backend framework. SLF4J library finds available logging backend from the JVM classpath at application startup. Actual log writing is done by the logging backend framework, which is chosen and configured by the system operator depending on his infrastructural needs. The TOE does not depend on any specific features of the logging backend framework. The TOE has been tested with a `Logback` implementation, which is recommended logging backend for SLF4J and is part of library set defined in the Spring Boot project.

This configuration also assumes, that TOE is run by privileged administrator, who installs and configures only necessary for the TOE JVM libraries, and ensures that TOE is always the only application executing within the JVM process and classpath.

2.4.4 Database

The TOE is using general purpose database to store the operational data. The sensitive fields in the database are encrypted and they are protected with the integrity protection mechanisms. Therefore, the security features of the database are not relevant for the security of the TOE.

The TOE has been tested with PostgreSQL 15 and is compatible with older PostgreSQL versions as well. The TOE does not use any non-standard SQL queries or database functions and as such is very much version agnostic, so mostly likely also supports future versions. However, the version of PostgreSQL used should be supported by the database software vendor.

2.4.5 Hardware Security Module

The HSM supplies its own set of security functions and has to be certified to be compliant with the QSCD requirements according to eIDAS reg. (EU) 910/2014 [4]. The HSM is regarded as the trusted device and the TOE relies on the security functions of the HSM in order to fulfil subset of the security objectives of the TOE.

The TOE has been tested with the following HSMs:

1. Thales nShield Connect 6000+ HSM, with part number NH2068 and with the following version information:

- nShield HSM family version 11.72.02
- nCore firmware version 2.55.1
- nShield Connect firmware image version 0.9.9
- Hardserver version 2.92.1
- Client libraries: Generic stub version 3.30.5, NFKM and RQCard version 1.86.1, and PKCS#11 version 2.14.1
- Client utilities version 2.54.1

The TOE is also compatible with other HSM models from the same nShield Connect+ lineup when configured using the above Common Criteria certified software packages. The TOE is also compatible with updated software and firmware versions for these HSM models provided they have also been certified to the same requirements as outlined in the Common Criteria certification for the nShield HSM family release v11.72.02.

2. Entrust nShield Solo XC for nShield Connect XC, with part number NH2075 and with the following version information:

- Solo XC firmware version 12.60.15
- nShield Connect XC image version 12.70.8

The TOE is also compatible with other HSM models from the same lineup and software versions that fulfil the following requirements:

- Entrust nShield Solo XC firmware updates that are certified to the Common Criteria Protection Profile – Cryptographic Module for Trust Service Providers (EN 419221-5) with the same or higher Evaluation Assurance Level (EAL4+) as the tested Solo XC firmware version 12.60.15.
- Any up-to-date nShield Connect XC image version declaring compatibility with a compatible Solo XC firmware version as defined above.

2.5 Description of the TOE

2.5.1 Physical scope of the TOE

The following physical items make up the physical scope of the TOE:

1. Smart-ID SecureZone service software package, delivered as a Java archive (.jar) file.
2. Smart-ID SecureZone administrative command line interface (CLI) tool, delivered as a Java archive (.jar) file.
3. Installation and Administration Guides for SecureZone, consisting of:

   3.1 Administration Guide for SecureZone [10], delivered in pdf format

   3.2 Installation Guide for SecureZone [11], delivered in pdf format

   3.3 Smart-ID SecureZone monitoring guide [12], delivered in pdf format

   3.4 Signer User Guidance information for SecureZone and TSE library operators [13], delivered in pdf format

Each part of the TOE physical scope is delivered via a secure file transfer system. The secure delivery procedure of the items constituting the physical scope of the TOE must include verification of the checksums of all the delivered components and verification of the correspondence of version numbers in the TOE documentation and the .jar files.

2.5.2 Components outside of the physical scope of the TOE

The TOE is physically represented by the Smart-ID SecureZone software, written in Java and packaged into a Java archive file. The Java archive is installed and executed by the JVM process, see 2.4.3. Because the TOE does not rely on the security features of the JakartaEE (JakartaEE) Virtual Machine (VM), application server, application log framework, or the server operating system, those components are outside of the physical scope of the TOE.

The TOE exposes an API to the outside world, which can be used by external users to initiate communication to the TOE. This API is TOE Security Functionality Interface (TSFI). The TOE also uses the HSM and database APIs and because the information retrieved over those interfaces also influences TOE security functionality, they are considered to be TSFIs as well.

The TOE uses the HSM for the cryptographic operations. The HSM is required to be trusted and the security functions of the HSM are required to be certified to be compliant with the QSCD requirements according to eIDAS reg. (EU) 910/2014 [4]. Because the HSM is already certified, it is not included in the scope of the TOE and the security functions of the HSM are not evaluated according to the current ST.

The TOE uses an external database to store the cryptographic key material and to keep track of the key usage information. The TOE encrypts the sensitive data fields in the database and utilises the integration protection techniques, so that the external database component can be un-trusted and cannot influence the TOE Security Functions (TSF).

The overview of the physical scope of the TOE is given in the Figure 3.

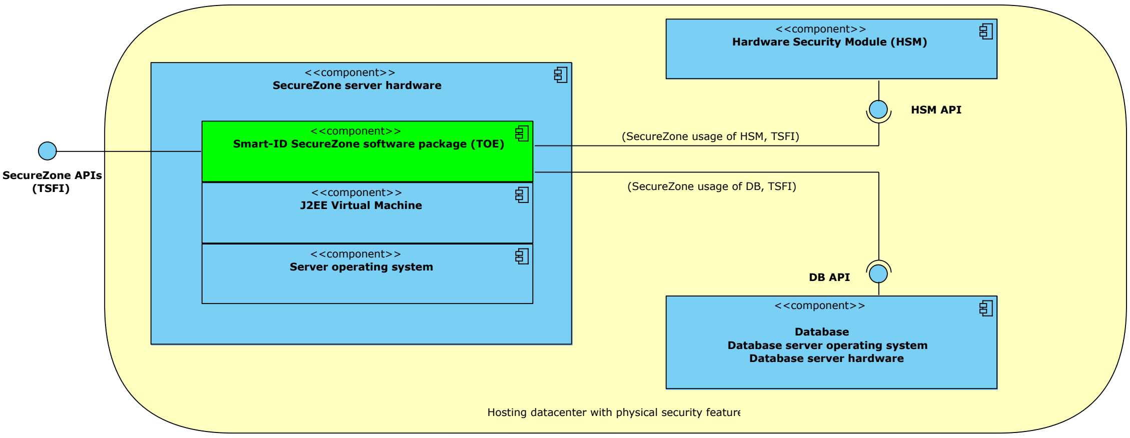

<<component>>
Hardware Security Module (HSM)
<<component>>
SecureZone server hardware
HSM API
<<component>>
Smart-ID SecureZone software package (TOE)
(SecureZone usage of HSM, TSFI)
SecureZone APIs
(TSFI)
<<component>>
J2EE Virtual Machine
(SecureZone usage of DB, TSFI)
<<component>>
Server operating system
DB API
<<component>>
Database
Database server operating system
Database server hardware
Hosting datacenter with physical security feature

Figure 3. TOE physical scope.

2.5.3 Logical scope of the TOE

This section describes the logical scope of the TOE.

2.5.3.1 TOE management and access control

1. Authentication – This function provides different methods to authenticate users and protect the assets of the TOE. Technical functions of the TOE, which do not require personalised user identification/authentication and strict access control, are not authenticated (cf. 7.2.5 or the monitoring interface). Nevertheless, even if they are not authenticated, they may still be restricted using IP address based whitelists.

   Other operations claim authentication of the users. The S.App and S.Admin authenticate with possession-based data (user-name and password). The S.Signer uses two-factor authentication based on knowledge (D.PIN) and possession of secret assets (D.OTP, D.clientPart). The authentication process is based on the TSSP, which is described in the section 2.3. Additionally, there exists an administrator-configurable lock mechanism to restrict the unsuccesful signer authentication by locking their key pair for a certain time period.

2. Access control – Different users have access to their various assets and allowed operations. Anonymous users are allowed to perform some operations, which do not require authentication and authorisation (for example, querying the status of a key pair or certain non-sensitive information).

   The access of Signer depends on his authentication method. In case of authentication with possession-based authentication factors, when the Smart-ID App is performing technical operations on behalf of the Signer and the App doesn't request authorisation with the entry of the D.PIN from the Signer, the TOE only allows to perform technical operations (creating a signature is not possible).

   The key pair owners (Signers) are allowed to perform the key pair operations on their own key pair. In case that the Signer is authenticated with possession-based and the knowledge-based authentication data, the TOE allows to complete the signature.

   Privileged users can perform key pair operations on any key pair, however, the list of operations is limited to only specific methods. Privileged users are not allowed to invoke signature completion at all.

All the rules are described in more detail within the section 7.2 Security Requirements (ASE_REQ). On the other hand, the section 8.1.1.2 TOE Summary Specification (ASE_TSS) desribes the details of the TOE Access Control mechanism.

3. Security audit generation – The audit records of the important system events are generated by the TOE and saved to its database to be exported to an external system.

2.5.3.2 Handling of cryptographic material and algorithms

1. Key generation – the TOE uses a Common Criteria certified HSM to perform most of the key generation operations. In case the HSM doesn't support generation and management of a particular key type, the TOE generates that by itself. The following keys are generated:

   - D.SVD (by TOE implementing the TSSP [5]) using modulus multiplication of D.clientModulus and D.serverModulus,
   - D.serverShare RSA key (by HSM),
   - D.KTK RSA key (by HSM),
   - D.TEK symmetric key (by TOE and TSE, using Diffie-Hellman for key exchange),
   - D.KWK AES key (by HSM) and
   - D.DEK AES key (by TOE).

   Further details can be found in 8.1.2.1 TOE Summary Specification (ASE_TSS) section.

2. Re-key process – the TOE allows the CA to use the reKey operation during which the new D.serverShare RSA key is generated (by HSM) and the new corresponding D.SVD is created (by the TOE) and associated with an existing D.Signing_Key_Id.

   See also section 7.2.1 Security Requirements (ASE_REQ), table 13. Further details about the generation of the mentioned keys can be found in 8.1.2.1 TOE Summary Specification (ASE_TSS) section.

3. Batch pre-generation of D.serverShare assets – For performance reasons the TOE requires Administrator to use the batchGenerateServerShares method to pre-generate a new batch of D.serverShare assets. At the time of creation, they are not associated with any existing D.Signing_Key_Id. They will be used during the new key-pair enrolments so that Signer enrolment can be done quicker.

   See more in section 7.2.1 Security Requirements (ASE_REQ), table 14.

4. Storing and protection of keys – The following cryptographic keys are stored in the TOE database, protected by the HSM master key: D.KTK, D.KWK and D.serverShare.

5. Cryptographic algorithms and operations – The following cryptographic algorithms are used in the TOE processes: computation of the signatures implementing the TSSP, creation and verification of RSA signatures, and encryption/decryption of JSON Web Encryption (JWE) messages for transmission and database storage.

6. Key destruction – The TOE destroys the following cryptographic keys after they are no longer used: D.serverPart, D.serverShare, D.DEK, D.TEK, D.KWK, D.KTK.

The section 8.1.2 TOE Summary Specification (ASE_TSS) describes the details of the mechanisms of cryptographic material and algorithms.

2.5.3.3 Protecting communication with external components

1. Trusted path with the user – the TOE uses JWE messages for communicating with the Smart-ID App TSE. JWE messages are encrypted with the D.TEK and they are integrity protected.
2. Secure channel with external components – the TOE uses vendor-specific proprietary communication channel when communicating with the HSM or the database, such as nShield impath and PostgreSQL connections. Those methods provide the cryptographic checksum validation of the integrity for the transmitted data. When the TOE detects the modifications and integrity errors with the transmitted data, it aborts the operation.

2.5.4 Features outside of the logical scope of the TOE

The TOE only provides the key pair related security functions and it doesn't have any features related to the identity proofing, Signer registration, certification issuing, and other features, which are commonly required by the full-scale PKI system.

Other features, which may be installed and configured on the SecureZone server hardware as well, are not included in the logical scope of the TOE. For example, the following features are not included in the logical scope of the TOE:

1. Software included in the operating system libraries and the applications, which are required to run and manage the SecureZone server.
2. HSM software packages, in case of the nShield HSMs, the nShield nCore API libraries, the nShield hardserver software and the file store for the nShield Security World.
3. Database software packages and libraries, which are required to connect to the external database server.

3 Conformance Claims (ASE_CCL)

3.1 CC Conformance

As defined by the references [1], [2] and [3], this TOE conforms to the requirements of Common Criteria version 3.1, revision 5.
Particularly: This Security Target claims to be Common Criteria Part 2 [2] and Common Criteria Part 3 [3] conformant.

3.2 Package conformance

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

3.3 PP Conformance

This ST does not claim conformance to any PP.

3.4 EU regulation conformance

This ST claims conformance to reg. (EU) 910/2014 [4] with fulfilling the following organisational policy requirements defined in section 4.5:

1. P.SCD_Confidential
2. P.SCD_Unique
3. P.Sig_unForgeable
4. P.SCD_userOnly
5. P.DTBS_Integrity
6. P.TSP_Qualified
7. P.SCD_Backup
8. P.DTBS/R_Unique
9. P.TSP_QCert

4 Security Problem Definition (ASE_SPD)

This section gives the list and definitions of the conceptual data assets, which are used to describe the threats and security objectives of the TOE. Not all of the data assets are managed or protected by the TOE itself. For more details, please refer to the list of user attributes and security attributes in the section 7.1.

4.1 Assets

Name	Description	Security
D.application SignaturePart	Share of the signature of D.DTBS/R, which is computed by the Signer with the D.clientPart. It is not possible to validate the D.applicationSignature Part with any public key. This is one part of the D.SAD since when combined with D.server SignaturePart, it will be the proof that the Signer used a correct PIN on the client side.	confidentiality, integrity
D.application Signature Share	Share of the signature of D.DTBS/R, which is created with the private key corresponding to the compound of D.clientPart and D.serverPart. Since that compound private key (D.clientShare) is destroyed after it has been split into the aforementioned parts, this signature share is instead created from the corresponding signature shares D.applicationSignaturePart and D.server SignaturePart. The D.applicationSignatureShare can be validated with D.clientModulus.	confidentiality, integrity
D.Audit_Data	Audit records generated by the TOE and stored and protected outside of the TOE.	confidentiality, integrity
D.clientModulus	Data, which can certify the integrity of D.application SignatureShare. This is the public part of the D.clientShare/D.clientModulus key pair.	integrity
D.clientPart	Part of the D.SCD. It is generated and protected by the Signer's PIN in the Smart-ID App sandbox in the Signer's mobile device. This also serves as one of the possession-based authentication factors used to authenticate the signer.	confidentiality, integrity

Name	Description	Security
D.clientShare	Part of the D.SCD. It is generated in the Smart-ID App sandbox in the Signer's mobile device, mathematically divided into D.clientPart and D.serverPart, and then deleted.	confidentiality, integrity
D.DEK	Symmetric cryptographic key, which is used by the TOE to encrypt and to integrity protect some database fields. D.DEK is generated and used by the TOE itself, and it is wrapped with the D.KWK.	confidentiality, integrity
D.DTBS	A set of data, which the Signer intends to sign in the SCA.	integrity
D.DTBS/R	A representation of a set of data, which the Signer intends to sign. This is the digest value that is generated from D.DTBS with the given hash algorithm.	integrity
D.KTK	Asymmetric encryption/decryption key pair, which is used to wrap the key material during the transmission from TSE to TOE. The TOE uses the HSM to generate and protect the key.	confidentiality, integrity
D.KWK	Symmetric encryption/decryption and integrity protection key, which is used to wrap the key material in the TOE database. The TOE uses the HSM to generate and protect the key	confidentiality, integrity
D.OTP	One-time password. Password token, which is updated and given to the TSE by the TOE for each subsequent key pair operation.	confidentiality, integrity
D.PIN	PIN is known by Signer and is entered to the TSE by Signer to authorise each signing operation. The D.PIN itself is never stored within TSE or TOE and never transmitted. Instead, the D.PIN is only used to derive the encryption/decryption key, which is used to protect the D.clientPart, when stored in the Signer's mobile device.	confidentiality
D.Privileged_ User	A set of data, that uniquely identifies a Privileged User within the TOE. In the TOE, there are two types of privileged users: 1. Administrator 2. CA	confidentiality, integrity
D.Random	Source of the random numbers, which are used to generate the encryption keys.	confidentiality, integrity

Name	Description	Security
D.Reference_ App_ Authentication_ Data	This data is used by the TOE to authenticate the Signer's mobile device where the Smart-ID App TSE has been installed, i.e. this is the data related with the Signer's possession-based authentication factor. It consists of: 1. D.OTP 2. D.Signing_Key_Id	confidentiality, integrity
D.Reference_ Privileged_ User_ Authentication_ Data	A set of data used by the TOE to authenticate the privileged user.	confidentiality, integrity
D.SAD	Signature Activation Data is a set of data involved in the signature activation protocol (SAP), which is used to authenticate and authorise the signature completion operation in the TOE. D.SAD consists of: 1. D.Reference_App_Authentication_Data 2. D.applicationSignaturePart 3. D.DTBS/R Since a part of the D.SAD (D.applicationSignature Part) is created on the TSE side using D.PIN, it is indirectly also a knowledge-based authentication factor. See also Application Note 1 for some clarifications about the nature of D.SAD in TSSP.	confidentiality, integrity
D.SCD	Signature Creation Data. In the conventional digital signature systems, this corresponds to the private key of the Signer's key pair. In the Smart-ID system, the D.SCD is never generated or combined in a single location. Instead, the three components of the D.SCD (D.clientPart, D.serverPart, D.serverShare) are generated and processed within distinct sub-systems.	(virtual asset)
D.server SignaturePart	Share of the signature of D.DTBS/R, which is computed by the TOE with the D.serverPart. It is not possible to validate the D.serverSignaturePart with any public key.	confidentiality, integrity
D.server Signature Share	Share of the signature of D.DTBS/R, which is created with the private key D.serverShare. The D.serverSignatureShare can be validated with the D.serverModulus.	confidentiality, integrity, non-repudiation

Name	Description	Security
D.serverModulus	Data, which can certify the integrity of D.server SignatureShare. This is the public part of the D.serverShare/D.serverModulus key pair.	integrity
D.serverPart	Part of the D.SCD of the Signer. Server part of the client's private key D.clientShare, which is generated in the TSE. It is transmitted to the TOE and protected by the TOE.	confidentiality, integrity
D.serverShare	Part of the D.SCD of the Signer. Server share of the private key, generated and protected by the HSM.	confidentiality, integrity
D.signature	Signature of the D.DTBS/R, which is created with the private key corresponding to the compound of D.clientPart, D.serverPart, and D.serverShare. As such private key does not actually exist, the signature is instead created from the signature shares D.applicationSignatureShare and D.server SignatureShare. The D.signature can be validated with the D.SVD.	integrity, non-repudiation
D.signature Parameters	Signature creation parameters set by the client and used in creation of D.applicationSignaturePart, D.serverSignaturePart and D.serverSignatureShare if the used signature scheme requires its usage. E.g., RSASSA-PSS requires the usage of a salt parameter (see RFC8017 [14]), which in the context of TSSP needs to be shared between the client and TOE.	confidentiality, integrity
D.Signer	Set of data, which represents the Signer and his/her identity. In the TOE, D.Signer is represented by D.Signing_Key_Id	integrity
D.Signing_ Key_Id	Unique identifier, which is used to connect all of the assets related to the signing key D.SCD: the assets D.clientPart, D.serverPart in the database and D.serverShare in the Cryptographic Module. It is also referred to as Key Universally Unique IDentifier (keyUUID) in some places since this is the name of this attribute in the developer documents and source code.	integrity
D.SVD	Signature verification data is the public part, associated with the signing key, for performing digital signature verification. In Smart-ID system, it is the compound modulus created by D.clientModulus and D.serverModulus. It is the data which is used to certify the integrity of the D.signature. The integrity of D.SVD is protected by the certificate issued by the CA.	integrity

Name	Description	Security
D.TEK	Symmetric cryptographic key shared between the TOE and a specific instance of TSE. D.TEK is established during the key pair enrolment with the Diffie-Hellman key exchange algorithm. It is used to protect the communication between the TSE instance and TOE.	confidentiality, integrity
D.TSF_ CONFIG_ DATA	It is the set of TOE configuration data used to operate the TOE. Among other parameters, it contains: • reference to the D.KWK keys used by the TOE; • reference to the D.DEK key used by the TOE; • list of valid cryptographical key sizes for the compound key D.SVD.	confidentiality, integrity

Application Note 1

The Protection Profile upon which this TOE is based on, PP 419 241-2 [6], contains the following three definitions:

1. R.Reference_Signer_Authentication_Data - the set of data used by TOE to authenticate the signer;
2. R.Authorisation_Data - data used by the TOE to activate a signing key in the Cryptographic Module;
3. R.SAD - set of data involved in the signature activation protocol, which activates the signature creation data to create a digital signature under the signer's sole control.

However, due to the specific nature of TSSP, there are no separate processes for the authentication of the Signer and for the signing key activation in the TOE. The Signer authentication and the signature completion happens in a single flow according to Section 2.3.3 – System Overview, during the process steps 9 to 13, and by using the same asset components.

Because of this, all of the three aforementioned assets are identical in the TOE. This single asset is referred to as D.SAD.

Remark: In an earlier version of the given Security Target these assets were separately defined (as D.Reference_Signer_Authentication_Data *and* D.Authorisation_Data*).*

4.2 Subjects

The TOE provides services and functions to the following external entities (natural persons and external IT systems) and uses the following list of subjects and roles in order to regulate access to the assets.

4.2.1 Natural Persons

1. U.User – Registered user of the Smart-ID services. U.User is using the TOE services to produce Qualified Electronic Signatures. U.User owns the mobile device with the Smart-ID App installed on it. Smart-ID App provides convenient user interface for the TOE services. Depending on the level of authentication (multi-factor or single-factor), the U.User is either bound to the subject S.Signer or to the subject S.App.
2. U.Admin – Administrator of the Smart-ID SecureZone, who installs, configures and maintains the TOE. Note that even though the TOE is installed, configured and administrated by the Administrator, the authentication of the Administrator is handled by the supporting IT environment, for example, by the operating system, which the TOE is running on top of, and the HSM module, by the use of the OCS password.

   Those TOE and TOE environment's administrative functions that involve installation and operation of the HSM module shall be conducted at least under dual control.

   The TOE allows Administrator to use the following functions:

   a. supply the OCS password to the TOE;
   b. generate the new batch of D.serverShare assets, which will be used during the new key-pair enrolments;
   c. generate D.KTK, D.KWK and D.DEK assets;
   d. destroy unused D.serverShare assets;
   e. destroy D.KTK, D.KWK and D.DEK assets.

4.2.2 External IT Systems

The following external IT systems use the services and functions of the TOE:

1. U.Monitoring – The IT component in the environment, which is quering the TOE status, health and monitoring information. This information is public and is provided to the monitoring component without authentication and access control. The U.Monitoring is bound to the subject S.Anonymous when processing the queries.
2. U.CA – The IT component "Smart-ID CA", which is managing the certificates. The U.CA is bound to the subject S.CA after the authentication of requests. U.CA executes the following functions:

   a. destroying of the key-pair after the revocation of a certificate,
   b. starting the re-key process of a key-pair.

4.2.3 Subjects

The TOE uses the following list of subjects when processing the requests and performing the access control decisions to the functions and assets.

1. S.Signer – Owner of the D.SCD, who is using the TOE functions to produce Qualified Electronic Signatures. U.User is bound to the S.Signer after the successful multi-factor authentication, which includes the possession-based information (from the mobile device) and the knowledge-based information, which only the U.User knows.
2. S.App - The Smart-ID App instance in the mobile device of the U.User. The Smart-ID App is using the technical TOE functions (such as updating D.OTP, performing the re-key operation) on behalf of the U.User. The S.App has limited access to the TOE objects. U.User is bound to the S.App after a successful single-factor authentication, which includes the possession-based information from the mobile device.
3. Privileged users:

   3.1 S.Admin – Subject S.Admin is used when administrators perform the management functions of the TOE and they authenticate themselves with the OCS password. Because the sensitive data fields in the TOE database are encrypted, administrators cannot modify them without supplying the valid OCS password. In that sense, the HSM is providing the authentication function for the administrators. Those TOE and TOE environment's administrative functions that involve installation and operation of the HSM module shall be conducted at least under dual control.

   3.2 S.CA – The IT component Smart-ID CA (U.CA) is bound to the S.CA after trusted channel-based authentication which is configured by S.Admin.

4. S.Anonymous – This subject is used, when the access control to the TOE services is handled with other environment measures, such as network firewalls and other measures, which do not provide personalised identification. For example, the TOE status, health and monitoring information, and some key pair status information is provided to other components within the larger PKI system, without personalised authentication and without the requesting user being fully known.

4.2.4 Roles

The TOE uses the following list of roles, when processing the request and deciding the access control:

1. R.Signer - The role R.Signer is used only when the U.User has been authenticated with multi-factor authentication.
2. R.App - The role R.App is used when the Smart-ID App is using the technical TOE functions (such as update D.OTP, perform re-key operation) on behalf of the U.User.
3. R.Admin – The role R.Admin is used when the subject S.Admin is authenticated with the HSM OCS password and the administrative function is performed. Note that those TOE and TOE environment's administrative functions that involve installation and operation of the HSM module must be conducted at least under dual control.
4. R.CA – The role R.CA is used when the IT component Smart-ID CA (U.CA) is authenticated and starts the key destroying or the re-key function.
5. R.Anonymous - The role R.Anonymous is used when the user is bound to the S.Anonymous.

The described roles are only the logical entities. The mapping between the subjects and the roles are hard-coded in the TOE configuration and source code. The TOE doesn't need to implement a dynamic administration module for role and permissions management.

4.3 Threat Agents

1. S.Attacker – A human or process acting on his behalf, located outside of the TOE. It is assumed that S.Attacker has complete knowledge about the components of the Smart-ID system, the structure of the TOE, algorithms, and API interfaces. However, he doesn't know any secret values, e.g. the key material. S.Attacker has high attack potential.

4.4 Threats

The following kind of threats are considered within this ST document. The main goal of the S.Attacker is to perform one of the following sub-attacks:

1. create one or more forged D.signatures of fresh D.DTBS/R under the name of Signer or
2. decrease the trust in the signatures created with the service Smart-ID Trust Service Provider (TSP) and in the security of the TOE.

ST document organises the individual threats in subsections, in order to present closely related threats next to each other.

4.4.1 Threats related to the key enrolment

Attacker may use the vulnerabilities of the key enrolment process to impersonate the Signer or to derive the D.SCD of the Signer or get the Signer's certificate issued for a different key pair. The following specific threats are considered within this ST document.

4.4.1.1 T.Enrolment_Signer_Authentication_Data_Disclosed

An attacker is able to obtain whole or part of D.SAD during enrolment. This can be during generation, storage or transfer of the data to the TOE or transfer between the signer and TOE. As an example it could happen by:

- eavesdropping during the TSSP key enrolment phase and retrieving the components of D.SAD, which are transmitted from the Signer to the TOE.

Such data disclosure may allow a potentially incorrect Signer authentication, leading to an unauthorised signature operation on behalf of the Signer (with the Signer's signature on a fresh D.DTBS/R without the Signer's consent).

4.4.1.2 T.Enrolment_Signer_Impersonation

Attacker impersonates signer during enrolment. As an example, it could be:

- performing a MITM attack during the TSSP key enrolment phase and modifying the value of the Signer's key pair, such as D.SCD components. Attacker may then use the modified values to forge the signatures of the Signer or to impersonate the Signer to the TOE, in order to create Signer's signature on the fresh D.DTBS/R without Signer's consent.

The asset D.SAD is threatened.

This is the same threat as T.ENROLMENT_SIGNER_IMPERSONATION in PP 419 241-2 [6].

4.4.1.3 T.SVD_Forgery

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

The asset D.SVD is threatened.

If the CA relies on the generation of the key pair controlled by the TOE as specified in ETSI 319 411-1 [15], requirement **GEN-6.3.3-06**, then an attacker can forge signatures masquerading as the signer.

This is the same threat as T.SVD_FORGERY in PP 419 241-2 [6].

> Application Note 2
>
> Issuing the certificate verifies the CSR – “proof of possession or control of the private key”, associated with the D.SVD, as specified in ETSI 319 411-1 [15], requirement **REG-6.3.1-01**. Therefore, this threat is countered without any specific measures within the TOE.

4.4.1.4 T.Random

Attacker guesses system secrets and is able to create or modify TOE objects or participate in communication with external systems.

D.Random is used to generate the D.SCD and other encryption/decryption keys. If attacker is able to guess random numbers, the attacker may be able to successfully derive the value of the D.SCD or other encryption/decryption keys and then impersonate the Signer to the TOE or create Signer’s signature on the fresh D.DTBS/R without Signer’s consent.

The asset D.Random is threatened.

This is the same threat as T.RANDOM in PP 419 241-2 [6].

4.4.2 Threats related to impersonation of the Signer within the signing process

Attacker may use the vulnerabilities in the signing process and try to impersonate the Signer to the TOE or use some other ways to get the TOE to create Signer’s signature without the Signer’s consent. The following specific threats are considered within this ST document.

This group of threats corresponds to a more general threat T.SigF_Misuse from PP 419 211-2 [16].

4.4.2.1 T.SAD_Forgery

Attacker forges or manipulates D.SAD during transfer in TSSP and is able to create a signature on the fresh D.DTBS/R without the Signer having authorised the operation.

The asset D.SAD is threatened.

4.4.2.2 T.SAP_ByPass

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

The asset D.SAD is threatened.

4.4.2.3 T.SAP_Replay

Attacker replays one or more steps of TSSP and is able to create a signature on the fresh D.DTBS/R without the signer having authorised the operation.

The asset D.SAD is threatened.

4.4.2.4 T.TSSP_Modification

Attacker modifies the user's data and/or security attributes within the TOE data storage and is able to submit the query to the TOE's signing function so that TOE outputs the Signer's signature on the fresh D.DTBS/R without the Signer's consent.

The assets D.SAD and D.Signing_Key_Id are threatened.

This threat corresponds to the threats T.MAINTENANCE_AUTHENTICATION_DISCLOSE and T.SIGNER_AUTHENTICATION_DATA_MODIFIED as in PP 419 241-2 [6].

4.4.2.5 T.TSSP_Duplication

Attacker gets hold of the D.clientPart, D.OTP and D.TEK and impersonates Signer to the TOE's signing function. Attacker is able to submit the valid D.applicationSignatureShare with the fresh D.DTBS/R so that TOE outputs the Signer's signature on the fresh D.DTBS/R without the Signer's consent.

The assets D.clientPart, D.OTP, D.TEK and D.applicationSignatureShare are threatened.

This threat corresponds to the T.AUTHENTICATION_SIGNER_IMPERSONATION in the PP 419 241-2 [6].

4.4.3 Threats related to signature forgery

Attacker may use the vulnerabilities in the cryptographic algorithm and the signature scheme itself or the hashing function itself and try to claim that Signer has signed such documents, which he has not intended. The following specific threats are considered within this ST document.

4.4.3.1 T.Signature_Forgery

Attacker uses a vulnerability in the cryptographic signature algorithm and forges (without having a copy of the D.SCD) the value of the new signature for a fresh D.DTBS/R, which can be successfully validated with D.SVD.

The asset D.signature is threatened.

4.4.3.2 T.DTBSR_Forgery

Attacker modifies the D.DTBS/R, before it is submitted to the Signer from the SCA (Signature Creation Application) or within the TOE, during the execution of the TOE's signing function. Attacker can then get the signature on a different kind of D.DTBS/R than was intended to be signed by the Signer.

The asset D.DTBS/R is threatened.

This threat corresponds to the T.DTBSR_FORGERY in the PP 419 241-2 [6].

4.4.4 Other threats

Attacker may use other attacks on the TOE to create the signatures and he may also try to attack the audit log of the TOE in order to claim that Signer has signed some documents, which he has not intended. The following specific threats are considered within this ST document.

4.4.4.1 T.Admin_Impersonation

Attacker impersonates a Privileged User and updates D.SAD, D.Signing_Key_Id, and/or D.SVD. Such data modification may allow a potential incorrect signer authentication leading to unauthorised signature operation on behalf of Signer.

The assets D.SAD, D.Signing_Key_Id, and D.SVD are threatened.

This threat corresponds to the threats T.ADMIN_IMPERSONATION, T.AUTHORISATION_DATA_UPDATE and T.AUTHORISATION_DATA_DISCLOSE as in PP 419 241-2 [6].

4.4.4.2 T.Privileged_User_Insertion

Attacker is able to create D.Privileged_User including D.Reference_Privileged_User_Authentication_Data and is able to log on to the TOE as a Privileged User.

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

4.4.4.3 T.Reference_Privileged_User_Authentication_Data_Modification

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

The asset D.Reference_Privileged_User_Authentication_Data is threatened.

4.4.4.4 T.Audit_Alteration

Attacker modifies system audit and is able hide trace of TOE modification or usage in the following ways:

- Attacker attacks the audit function of the TOE or the audit log storage outside of the TOE and deletes the existing log entries, modifies the existing log entries or creates new log entries. Attacker is then able to hide his own actions and attack attempts or he is able to claim that the Signer has signed a different kind of D.DTBS/R than intended by the Signer, even though the corresponding D.signature may not even exist.

The asset D.Audit_Data is threatened.

4.4.4.5 T.Context_Alteration

Attacker modifies the system configuration D.TSF_CONFIG_DATA to perform an unauthorised operation in the following way:

- Attacker gets root-level or physical access to the TOE or underlying IT components and is able modify the user's data, security attributes, and/or program code of the TOE. Attacker is then able to produce a D.signature for a fresh D.DTBS/R, which the Signer did not intend to sign.

The assets D.Signing_Key_Id, D.SAD, D.SVD, and D.TSF_CONFIG_DATA are threatened.

4.4.4.6 T.Signature_Request_Disclosure

Attacker obtains knowledge of D.DTBS/R or D.SAD during transfer to the TOE.

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

4.4.5 Relations between threats and assets

Table 4. Compiled overview of relations between threats and assets

Asset	Security Requirement	Threats
D.Signing_Key_Id	Integrity	T.TSSP_Modification, T.Admin_Impersonation, T.Context_Alteration
D.serverShare	Integrity, Confidentiality	
D.serverPart	Integrity, Confidentiality	
D.SCD	(virtual asset)	
D.PIN	Confidentiality	
D.clientPart	Integrity, Confidentiality	T.TSSP_Duplication
D.DTBS	Integrity	
D.DTBS/R	Integrity	T.DTBSR_Forgery, T.Signature_Request_Disclosure
D.SAD	Integrity, Confidentiality	T.SAD_Forgery, T.SAP_Bypass, T.SAP_Replay, T.Context_Alteration, T.Signature_Request_Disclosure, T.Enrolment_Signer_Authentication-_Data_Disclosed, T.Enrolment_Signer_Impersonation, T.TSSP_Modification, T.Context_Alteration
D.Reference_App_ Authentication_Data	Integrity, Confidentiality	T.Context_Alteration
D.applicationSignaturePart	Integrity, Confidentiality	
D.serverSignaturePart	Integrity, Confidentiality	
D.signatureParameters	Integrity, Confidentiality	
D.applicationSignature Share	Integrity, Confidentiality, Non-repudiation	T.TSSP_Duplication
D.serverSignatureShare	Integrity, Confidentiality, Non-repudiation	

Table 4. Compiled overview of relations between threats and assets

Asset	**Security Requirement**	**Threats**
D.signature	Integrity, Non-repudiation	T.Signature_Forgery
D.SVD	Integrity	T.SVD_Forgery, T.Admin_Impersonation, T.Context_Alteration
D.clientModulus	Integrity	
D.serverModulus	Integrity	
D.Audit_Data	Integrity, Confidentiality	T.Audit_Alteration
D.Signer	Integrity, Confidentiality	
D.TSF_CONFIG_DATA	Integrity, Confidentiality	T.Context_Alteration
D.Privileged_User	Integrity, Confidentiality	T.Privileged_User_Insertion
D.Reference_Privileged_User_Authentication_Data	Integrity, Confidentiality	T.Privileged_User_Insertion, T.Reference_Privileged_User_Authentication_Data_Modification
D.Random	Integrity, Confidentiality	T.Random
D.DEK	Integrity, Confidentiality	
D.TEK	Integrity, Confidentiality	T.TSSP_Duplication
D.KWK	Integrity, Confidentiality	
D.KTK	Integrity, Confidentiality	
D.OTP	Integrity, Confidentiality	T.TSSP_Duplication

4.5 Organization Security Policies

4.5.1 P.SCD_Confidential

The confidentiality of D.SCD must be reasonably assured (from reg. (EU) 910/2014 [4], Annex II, point 1.(a)).

4.5.2 P.SCD_Unique

Any given instance of a D.SCD shall occur only once (from reg. (EU) 910/2014 [4], Annex II, point 1.(b)).

4.5.3 P.Sig_unForgeable

An electronic signature shall be reliably protected against forgery using currently available technology. It shall not be possible, with reasonable assurance, to derive an electronic signature from data other than the D.SCD (from reg. (EU) 910/2014 [4], Annex II, point 1.(c)).

4.5.4 P.SCD_userOnly

D.SCD of a legitimate Signer shall be reliably protected against use by others (from reg. (EU) 910/2014 [4], Annex II, point 1.(d) and Article 26, point (c)).

4.5.5 P.DTBS_Integrity

The TOE and its environment shall not alter D.DTBS nor D.DTBS/R. The TOE and its environment shall not prevent such data from being presented to the Signer prior to signing (from reg. (EU) 910/2014 [4], Annex II, point 2).

4.5.6 P.TSP_Qualified

Generating or managing D.SCD on behalf of the Signer may only be done by a qualified trust service provider (from reg. (EU) 910/2014 [4], Annex II, point 3).

4.5.7 P.SCD_Backup

The TSP may duplicate the D.SCD only for back-up purposes provided the 1) security of the duplicated datasets must be at the same level as for the original datasets and 2) number of duplicated datasets shall not exceed the minimum needed to ensure continuity of the service (from reg. (EU) 910/2014 [4], Annex II, point 4).

4.5.8 P.TSP_QCert

The TSP must use a trustworthy Certificate Generation Application (CGA) to generate a qualified certificate for the SVD generated by the TOE. The TSP must ensure that the advanced electronic signature is uniquely linked to the Signer and the Signer can be identified through the related certificate (from reg. (EU) 910/2014 [4], Article 26, point (a) and (b)).

4.5.9 P.DTBS/R_Unique

The electronic signature must be linked to D.DTBS in such a way that any subsequent change in data is detectable – for example, any subsequent change in data shall result in a different D.DTBS/R generated for this data (from reg. (EU) 910/2014 [4], Article 26, point (d)).

4.5.10 P.Reliable_Audit

The TSP shall keep reliable audit records about the signing events.

4.6 Assumptions

4.6.1 A.CA

It is assumed that the qualified TSP that issues qualified certificates is compliant with the relevant requirements for qualified TSP's as defined in reg. (EU) 910/2014 [4]. The CGA protects the authenticity of the Signer's name and the SVD in the qualified certificate by an advanced electronic signature of the CSP.

4.6.2 A.ACCESS_PROTECTED

The TOE environment limits physical and logical access to the components in the TOE environment to an authorized U.Admin. The TOE software and hardware environment are maintained by U.Admin in a secure state, including protection against unauthorized software and configuration changes. The TOE environment provides reasonable protection against denial of service attacks.

It is assumed that copies of the data protected by the TOE are managed outside of the TOE, and appropriate protection is provided for that data to the level required by the application context and the risks in the deployment environment.

Informative: based on Application note 21 of the PP 419 241-2 [6], the following data are managed outside of the TOE:

- D.clientPart
- D.DTBS/R
- D.applicationSignaturePart
- D.Audit_Data
- D.serverShare

4.6.3 A.PRIVILEGED_USER

The U.Admin, who has unrestricted physical and logical access to the TOE and the TOE environment, is well-trained, trusted, and performs his duties competently. Those TOE and TOE environment's administrative functions that involve installation and operation of the HSM module shall be conducted at least under dual control.

4.6.4 A.SIGNER_ENROLMENT

The signer shall be enrolled and certificates managed in conformance with the regulations given in reg. (EU) 910/2014 [4]. Guidance for how to implement an enrolment and certificate management system in conformance with eIDAS reg. (EU) 910/2014 [4] are given in e.g. ETSI 319 411-1 [15] or for qualified certificate in e.g. ETSI 319 411-2 [17].

4.6.5 A.SIGNER_AUTHENTICATION_DATA_PROTECTION

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

4.6.6 A.SIGNER_DEVICE

The Signer has the trusted TSE component in his environment to help him to complete the TSSP steps for the key generation and signing operations. The TSE component is evaluated with the EAL2 level, according to the ST document [9], and it fulfills the security objectives OE.TSE.* and OE.DTBS_Intend.

4.6.7 A.TSP_AUDITED

It is assumed that the TSP deploying the TOE is a qualified TSP and is audited to be compliant with the requirements for TSPs, as described in reg. (EU) 910/2014 [4].

4.6.8 A.CSPRNG

It is assumed that the HSM provides the secure random number generator, which can be used by the TOE to generate the cryptographic keys and random nonces.

4.6.9 A.CRYPTO

It is assumed that cryptographic algorithms, algorithm parameters, and key lengths, which are in use by the TOE, are endorsed by recognized authorities as appropriate for the use of TSPs. This includes the algorithms for generation of random numbers and signing key pairs, the algorithms for creating signatures, and also the algorithms for protecting integrity and confidentiality of TOE assets.

> Application Note 3
>
> The TOE supports the standard cryptographic algorithms and recommended key sizes according to ETSI TS 119 312 [18] and [19].

4.6.10 A.JVM

It is assumed that the TOE is the only application which is running on the JVM.

5 Security Objectives (ASE_OBJ)

This chapter identifies and defines the security objectives for the TOE and its environment. Objectives counter the identified threats and comply with the organizational security policies and assumptions.

5.1 Security Objectives for the TOE

5.1.1 OT.SCD_Confidential

The TOE shall keep the D.serverPart components of the D.SCD confidential.

5.1.2 OT.Sig_Secure

The TOE shall generate electronic signatures, that cannot be forged without knowledge of the D.SCD, through robust cryptographic techniques. The TOE shall not allow the D.SCD to be reconstructed from the digital signatures.

5.1.3 OT.SCD/SVD_Corresp

The TOE shall guarantee the correspondence between the D.SVD and the D.SCD. This includes unambiguous reference of a created SVD/SCD pair for export of the D.SVD and in creating a digital signature with the D.SCD.

5.1.4 OT.TSSP_End2End

The TOE shall protect the confidentiality and integrity of the communications between the TOE and Signer. The TOE shall not allow the attacker to eavesdrop and modify the information transmitted between the TOE and the Signer.

5.1.5 OT.SAP_Replay_Protection

The TOE shall protect the communications between the TOE and Signer against replay attacks.

5.1.6 OT.TSSP_Require_clientSignatureShare

The TOE shall protect the signature creation function of the TOE by following the TSSP and requiring the valid D.applicationSignatureShare in order to create the D.signature.

5.1.7 OT.TSSP_Validate_clientSignatureShare

The TOE shall protect the signature creation function of the TOE by following the TSSP and validating the D.applicationSignatureShare in order to make sure that the correct D.clientPart has been used to create the D.applicationSignatureShare (validated with the D.clientModulus).

5.1.8 OT.TSSP_CloneDetection

The TOE shall protect the signature creation function of the TOE by following the TSSP and detecting the usage of incorrect D.OTP in signature creation requests with valid D.applicationSignatureShare. This situation indicates that Signer's local environment has been cloned. The valid D.clientPart has leaked, but only one of the clients has been issued the correct D.OTP for the subsequent key pair operation. The TOE shall initiate revocation of the Signer's certificate and destroy the respective key pair after detecting such situation.

5.1.9 OT.TSSP_TimeDelay_Locks

The TOE shall protect the signature creation function of the TOE by following the TSSP and after submission of incorrect D.applicationSignatureShare (which most likely indicates that the Signer has entered the wrong D.PIN to the TSE), the TOE shall prevent the immediate re-try of the signature creation request with new D.applicationSignatureShare for the same D.DTBS/R. The TOE shall apply time-delay between accepting the new requests and shall initiate revocation of the Signer's certificate and destroy the respective key pair after the limit of incorrect D.applicationSignatureShare submissions has been reached.

5.1.10 OT.DTBS/R_Protect

The TOE shall protect the D.DTBS/R from substitution and modification. The protection shall be also applied when the D.DTBS/R is transmitted from/to another IT component in the TOE environment.

5.1.11 OT.System_Protection

The TOE shall ensure that modification of D.TSF_CONFIG_DATA is authorized by D.Privileged_User.

5.1.12 OT.Audit_Events

The TOE shall create audit records about the important system events.

5.1.13 OT.Privileged_User_Management

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

5.1.14 OT.Privileged_User_Authentication

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

> Application Note 4
>
> The exception to this objective is when the initial (set of) Privileged Users are created as part of the system initialisation.

5.1.15 OT.Privileged_User_Protection

The TOE shall ensure that data associated with D.Privileged_User are protected in integrity and if needed, in confidentiality.

5.2 Security Objectives for the Environment fulfilled by HSM

The HSM inside the TOE environment is CC evaluated and conforming to the QSCD requirements. This means that the HSM fulfils several Security Objectives by design. Because HSM processes the components of the D.SCD and provides important security functions to the TOE, it is useful to show which security objectives for the environment are fulfilled by the HSM itself.

5.2.1 OE.HSM.SCD_Confidential

The HSM shall protect the confidentiality of the components of the D.SCD.

5.2.2 OE.HSM.SCD_Unique

The HSM shall ensure cryptographic quality of generated keys. HSM shall generate the D.serverShare (component of the D.SCD) and the corresponding D.serverModulus (component of the D.SCD) securely. It shall not be possible to derive D.serverShare from D.serverModulus and the probability of obtaining equal D.serverShare assets shall be negligible.

5.2.3 OE.HSM.Sig_Secure

The HSM shall generate electronic signatures (D.serverSignatureShare) that cannot be forged without knowledge of the private key (D.serverShare), through robust cryptographic techniques. The D.serverShare cannot be reconstructed from the digital signatures.

5.2.4 OE.HSM.Tamper_Resistance

The HSM shall prevent or resist physical tampering with HSM device and components.

5.2.5 OE.HSM.Sigy_SigF

The HSM shall provide the share of the signature (D.serverSignatureShare) creation function for the TOE only and protects the D.serverShare against attempts by other users to create a digital signature using it.

5.2.6 OE.HSM.DTBS/R_Integrity

The HSM shall ensure that the D.DTBS/R cannot be altered when processed by the HSM.

5.3 Security Objectives for the Environment fulfilled by TSE

The Threshold Signature Engine (TSE) inside the Signer environment is CC evaluated and it fulfils security objectives by design. Because the TSE processes the components of the D.SCD and provides important security functions to the Signer, it is useful to show which security objectives for the environment are fulfilled by the TSE itself.

5.3.1 OE.TSE.Sig_Secure

The TSE shall generate D.applicationSignaturePart, that cannot be forged without knowledge of the D.clientPart, through robust cryptographic techniques. The TSE shall not allow the private key to be reconstructed from the digital signatures.

5.3.2 OE.TSE.SCD_Unique

The TSE shall ensure the cryptographic quality of the generated keys. The TSE shall generate the D.clientShare (component of the D.SCD) and the corresponding D.clientModulus (component of the D.SVD) securely. It shall not be possible to derive D.clientShare from D.clientModulus and probability of equal D.clientShares shall be negligible.

5.3.3 OE.TSE.SCD_Confidential

The TSE shall protect the confidentiality of the components of the D.SCD.

5.3.4 OE.TSE.TSSP_End2End

The TSE shall protect the confidentiality and integrity of the communications between the TSE and TOE.

5.3.5 OE.TSE.App_Sandbox

The TSE shall be run in an isolated mobile app process, protected from other apps.

5.4 Security Objectives for the Environment fulfilled by other components

5.4.1 OE.CA_Request_Certificate

The operational environment shall ensure that the qualified TSP that issues qualified certificates is compliant with the relevant requirements for qualified TSPs as defined in reg. (EU) 910/2014 [4].

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

5.4.2 OE.Env

The TSP deploying the TOE is a qualified TSP and audited to be compliant with the requirements for TSPs given by reg. (EU) 910/2014 [4]. The audit of the qualified TSP shall cover the security objectives for the operational environment specified in this clause.

The TOE environment shall provide a System Overview which ensures that the TOE is the only application deployed in a container of the System Overview.

The TOE environment shall limit physical and logical access to the components in the TOE environment to an authorised U.Admin. The TOE software, hardware environment, and backup datasets shall be maintained by U.Admin in a secure state, including protection against unauthorised software and configuration changes.

5.4.3 OE.Trusted_Timestamps

The TOE environment shall provide trusted timestamps.

5.4.4 OE.Trusted_Admin

The U.Admin, who has unrestricted physical and logical access to the TOE and the TOE environment, shall be well-trained and trusted and shall perform his duties.

5.4.5 OE.SVD_Authenticity

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

The TOE environment shall ensure the integrity of the D.SVD exported by the TOE to the CGA. The CGA shall verify the correspondence between the D.SCD of the Signer and the D.SVD in the input provided to the certificate generation function of the CGA.

5.4.6 OE.DTBS_Intend

The Signature Creation Application (SCA) generates the D.DTBS/R of the data that has been presented as D.DTBS, which the Signer intends to sign. The TOE environment shall allow for either manual (by the Signer) or automatic verification of the integrity of the D.DTBS/R, so that the Signer can be sure he is signing the same document that he intends to sign (see Section 2.3.3.3 for details).

5.4.7 OE.DTBS/R_Protect

The TOE environment shall ensure that the D.DTBS/R cannot be altered in transit between physically separated components of the TOE environment.

5.4.8 OE.DTBS/R_Unique

The TOE environment shall ensure that D.DTBS may practically have only one unique representation as D.DTBS/R. The TOE environment shall ensure that the probability for existence of two different D.DTBS-s having identical D.DTBS/R is negligible.

5.4.9 OE.CGA_QCert

The CGA shall generate qualified certificate and thus confirm that the D.SCD, corresponding to the certified D.SVD, is under the control of Signer. The CGA shall include identifying information of the Signer in the certificate and therefore enable to identify the Signer by the signature.

5.4.10 OE.Protected_AuditLog

The TOE environment shall protect the integrity of the audit log and protect the audit log from unauthorized deletion.

5.4.11 OE.CSPRNG

The HSM must provide the cryptographically secure random number generator for the TOE. The TOE will use the Random number generator (RNG) provided to generate D.OTP, D.TEK and D.DEK.

Application Note 5

The environment objective OE.CSPRNG has been defined to accurately reflect the implementation, where the SZ is using the HSM-provided random number generation service and it is not implementing the random number generation on its own.

5.4.12 OE.Signer_Authentication_Data

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

5.5 Security Objectives Rationale

5.5.1 Mapping between SPD and Security Objectives

The mapping between Security Problem Definition (SPD) and security objectives has been divided into multiple tables for size considerations, according to the type of the security objectives:

1. mapping to TOE security objectives is shown in the table 5 on page 61,
2. mapping to HSM security objectives is shown in the table 6 on page 62,
3. mapping to TSE security objectives is shown in the table 7 on page 63,
4. mapping to general environment security objectives is shown in the table 8 on page 64.

Table 5. Mapping between Security Problem Definition (SPD) and TOE security objectives

	OT.SCD_Confidential	OT.Sig_Secure	OT.TSSP_End2End	OT.SAP_Replay_Protection	OT.TSSP_Require_clientSignatureShare	OT.TSSP_Validate_clientSignatureShare	OT.TSSP_CloneDetection	OT.TSSP_TimeDelay_Locks	OT.DTBS/R_Protect	OT.SCD/SVD_Corresp	OT.System_Protection	OT.Audit_Events	OT.Privileged_User_Management	OT.Privileged_User_Authentication	OT.Privileged_User_Protection
T.Enrolment_Signer_Authentication_Data_Disclosed			X												
T.Enrolment_Signer_Impersonation			X												
T.SAD_Forgery						X		X							
T.SAP_ByPass					X										
T.SAP_Replay				X											
T.TSSP_Modification										X					
T.TSSP_Duplication							X								
T.Signature_Forgery		X													
T.DTBSR_Forgery				X					X						

Table 5. Mapping between Security Problem Definition (SPD) and TOE security objectives

	OT.SCD_Confidential	OT.Sig_Secure	OT.TSSP_End2End	OT.SAP_Replay_Protection	OT.TSSP_Require_clientSignatureShare	OT.TSSP_Validate_clientSignatureShare	OT.TSSP_CloneDetection	OT.TSSP_TimeDelay_Locks	OT.DTBS/R_Protect	OT.SCD/SVD_Corresp	OT.System_Protection	OT.Audit_Events	OT.Privileged_User_Management	OT.Privileged_User_Authentication	OT.Privileged_User_Protection
T.Admin_Impersonation					X										
T.Context_Alteration					X						X				
T.Signature_Request_Disclosure			X	X	X	X									
T.Privileged_User_Insertion													X	X	
T.Reference_Privileged_User_Authentication_Data_Modification													X	X	X
P.SCD_Confidential	X		X												
P.Sig_unForgeable		X													
P.SCD_userOnly	X		X	X	X	X	X	X							
P.DTBS_Integrity			X						X						
P.Reliable_Audit												X			

Table 6. Mapping between Security Problem Definition (SPD) and HSM security objectives

	OE.HSM.SCD_Confidential	OE.HSM.SCD_Unique	OE.HSM.Sig_Secure	OE.HSM.Tamper_Resistance	OE.HSM.Sigy_SigF	OE.HSM.DTBS/R_Integrity
T.Random		X		X		
T.Signature_Forgery			X			
T.DTBSR_Forgery						X
T.Context_Alteration				X		
P.DTBS_Integrity						X
P.SCD_Confidential	X			X		
P.SCD_Unique		X				
P.Sig_unForgeable			X			
P.SCD_userOnly	X	X		X	X	
A.Crypto		X	X			

Table 7. Mapping between Security Problem Definition (SPD) and TSE security objectives

	OE.TSE.Sig_Secure	OE.TSE.SCD_Unique	OE.TSE.TSSP_End2End	OE.TSE.SCD_Confidential	OE.TSE.App_Sandbox
T.Enrolment_Signer_Authentication_Data_Disclosed			X		
T.Enrolment_Signer_Impersonation			X		
T.Random		X			
T.Signature_Forgery	X				
T.DTBSR_Forgery					
T.Signature_Request_Disclosure			X		
P.SCD_Unique		X			
P.SCD_Confidential			X	X	X
P.SCD_userOnly		X	X	X	X

Table 7. Mapping between Security Problem Definition (SPD) and TSE security objectives

	OE.TSE.Sig_Secure	OE.TSE.SCD_Unique	OE.TSE.TSSP_End2End	OE.TSE.SCD_Confidential	OE.TSE.App_Sandbox
P.DTBS_Integrity					
P.Sig_unForgeable	X				
A.SIGNER_DEVICE	X	X	X	X	X

Table 8. Mapping between Security Problem Definition (SPD) and environment security objectives

	OE.Env	OE.SVD_Authenticity	OE.DTBS_Intend	OE.DTBSR/R_Protect	OE.DTBS/R_Unique	OE.CGA_QCert	OE.Trusted_Admin	OE.Trusted_Timestamps	OE.Protected_AuditLog	OE.CSPRNG	OE.CA_Request_Certificate	OE.Signer_Authentication_Data
T.TSSP_Modification						X						
T.Context_Alteration	X											
T.SVD_Forgery		X				X					X	
T.DTBSR_Forgery			X	X								
T.Audit_Alteration									X			
P.DTBS_Integrity			X	X								
P.DTBS/R_Unique					X							
P.TSP_QCert						X						
P.SCD_Backup	X											
P.TSP_Qualified	X											
P.Reliable_Audit								X	X			
A.ACCESS_PROTECTED	X											
A.CA		X				X					X	
A.PRIVILEGED_USER							X					
A.SIGNER_ENROLMENT	X											

Table 8. Mapping between Security Problem Definition (SPD) and environment security objectives

	OE.Env	OE.SVD_Authenticity	OE.DTBS_Intend	OE.DTBSR/R_Protect	OE.DTBS/R_Unique	OE.CGA_QCert	OE.Trusted_Admin	OE.Trusted_Timestamps	OE.Protected_AuditLog	OE.CSPRNG	OE.CA_Request_Certificate	OE.Signer_Authentication_Data
A.SIGNER_AUTHENTICATION_DATA_PROTECTION												X
A.SIGNER_DEVICE			X									
A.TSP_AUDITED	X											
A.CSPRNG										X		
A.CRYPTO										X		
A.JVM	X											

5.5.2 Security Objectives Rationale

5.5.2.1 Rationale for mitigating threats

5.5.2.1.1 Mitigating T.Enrolment_Signer_Authentication_Data_Disclosed and T.Enrolment_Signer_Impersonation

T.Enrolment_Signer_Authentication_Data_Disclosed (Attacker eavesdrops on the TSSP key enrolment run and retrieves the D.SCD components, which are transmitted from the Signer to the TOE) and T.Enrolment_Signer_Impersonation (Attacker performs the MITM attack on the TSSP key enrolment run and modifies the value of the Signer's key pair, such as D.SCD components) are mitigated by OT.TSSP_End2End and OE.TSE.TSSP_End2End, which in combination, give the following assurances:

1. Signer authenticates the TOE by the known public key.
2. The TOE authenticates the instance of the Signer by the Signer's Diffie-Hellman public key and the shared symmetric encryption key.
3. Signer and the TOE use the Diffie-Hellman key exchange protocol to create the shared symmetric encryption key to protect the confidentiality and integrity of the communication channel.

5.5.2.1.2 Mitigating T.SAD_Forgery

T.SAD_Forgery (Attacker submits forged value of D.applicationSignatureShare to the TOE's signing function and is able to get the Signer's signature on the fresh D.DTBS/R) is mitigated by the OT.TSSP_Validate_clientSignatureShare and OT.TSSP_TimeDelay_Locks.

First, OT.TSSP_Validate_clientSignatureShare ensures that the TOE computes the D.serverSignaturePart and combines it with the submitted D.applicationSignaturePart and creates the D.applicationSignatureShare. The validity of the D.applicationSignatureShare is verified with the D.clientModulus. Because only Signer has the correct D.clientPart, which was required to create the D.applicationSignaturePart, the TOE shall prevent the Signer impersonation and shall provide the signature creation function for the legitimate Signer only.

Secondly, OT.TSSP_TimeDelay_Locks limits the number of attempts the attacker has to guess the correct D.PIN or D.clientPart. The TOE security objective OT.TSSP_TimeDelay_Locks ensures that the TOE doesn't immediately accept a new signature creation attempt for the same D.DTBS/R. It also ensures that the TOE will destroy the key pair and initiate the revocation of the respective certificate after the limit of incorrect signature creation attempts has been reached.

5.5.2.1.3 Mitigating T.SAP_ByPass

T.SAP_ByPass (Attacker bypasses the access control part of the TOE's signing function and is able to get the Signer's signature on the fresh D.DTBS/R without providing the valid D.applicationSignatureShare) is mitigated by the OT.TSSP_Require_clientSignatureShare.

This security objective ensures that TOE implements the TSSP correctly and computes the compound signature D.signature only when a valid D.applicationSignatureShare is available. In fact, the cryptographic properties of the TSSP ensure that the computed D.signature is valid only in the case where all the signature shares, which are used (D.applicationSignatureShare and D.serverSignatureShare), are valid as well.

5.5.2.1.4 Mitigating T.SAP_Replay

T.SAP_Replay (Attacker eavesdrops the data, which is submitted to the TOE's signing function by the Signer and is able to modify the data and replay it, so that the TOE outputs the Signer's signature on the fresh D.DTBS/R without the Signer's consent) is mitigated by OT.SAP_Replay_Protection.

This security objective ensures that the TOE implements the TSSP correctly and computes the D.serverSignaturePart on the submitted D.DTBS/R and combines it with the submitted D.applicationSignaturePart and in turn, creates the D.applicationSignatureShare on the submitted D.DTBS/R. In fact, the D.applicationSignatureShare is the RSA signature and it has the cryptographic properties that in case the signed message has been changed, the signature is not valid anymore. Therefore, it is not possible to change the D.DTBS/R, after the Signer created the D.applicationSignaturePart for the particular D.DTBS/R.

5.5.2.1.5 Mitigating T.TSSP_Modification

T.TSSP_Modification (Attacker modifies the user's data and/or security attributes within the TOE data storage and is able to submit the query to the TOE's signing function so that the TOE outputs the Signer's signature on the fresh D.DTBS/R) is mitigated by OT.SCD/SVD_Corresp and OE.CGA_QCert.

First, the TOE security objective OT.SCD/SVD_Corresp ensures that D.SCD and D.SVD correspond to each other cryptographically. So, in the case where the attacker would be able to modify some components of the D.SCD, the assets D.SCD and the D.SVD would no longer match with each other.

Secondly, the environment security objective OE.CGA_QCert ensures that the authentic value of the D.SVD is recorded in the certificate issued by the CA. In case where the attacker would be able to modify the D.SCD, the authentic value of the public key (D.SVD from the certificate) would no longer correspond to the modified private key.

5.5.2.1.6 Mitigating T.TSSP_Duplication

T.TSSP_Duplication (attacker gets hold of the D.clientPart, D.OTP and D.TEK and impersonates Signer to the TOE's signing function) is mitigated by OT.TSSP_CloneDetection.

The TOE security objective OT.TSSP_CloneDetection ensures that the TOE detects situations where a valid D.applicationSignatureShare is submitted to the signature creation function along with an old or incorrect D.OTP. This indicates that multiple clients have been operating and only one of the clients has been issued the correct D.OTP for the subsequent key pair operation. In this case, the key pair is destroyed and the respective certificate's revocation is initiated by the TOE.

5.5.2.1.7 Mitigating T.Signature_Forgery

T.Signature_Forgery (attacker uses a vulnerability in the cryptographic signature algorithm and, without having a copy of the D.SCD, crafts the value of a new signature for a fresh D.DTBS/R) is mitigated by OT.Sig_Secure, OE.HSM.Sig_Secure and OE.TSE.Sig_Secure.

First, the TSE security objective OE.TSE.Sig_Secure ensures that it is not possible to generate the D.applicationSignaturePart without access to the private key D.clientPart, by ensuring that the TSE performs the signature computation according to the RSA signature algorithm and with using the specified key sizes.

Secondly, the HSM security objective OE.HSM.Sig_Secure ensures that it is not possible to generate the D.serverSignatureShare without access to the private key D.serverShare, by ensuring that the HSM performs the signature computation according to the RSA signature algorithm and with using the specified key sizes.

Finally, the TOE security objective OT.Sig_Secure ensures that it is not possible to generate the D.serverSignaturePart without access to the private key D.serverPart and finally, that it is not possible to generate the compound signature D.signature without having access to the components of the D.SCD (D.clientPart, D.serverPart and D.serverShare). This is ensured by the TOE by performing the signature computation according to the RSA signature algorithm and TSSP and with using the specified key sizes.

Therefore, signature forgery without having access to the D.SCD is not possible.

5.5.2.1.8 Mitigating T.DTBSR_Forgery

T.DTBSR_Forgery (attacker modifies the D.DTBS/R before or during the signing process) is mitigated by the following security objectives.

1. OT.SAP_Replay_Protection – Attacker cannot submit the eavesdropped signature creation request with a modified D.DTBS/R, because the D.applicationSignaturePart depends on the D.DTBS/R and the modified request would not pass validation.
2. OT.DTBS/R_Protect – D.DTBS/R is protected, when the TOE is processing the signature creation request or transmitting the D.DTBS/R to another IT component.
3. OE.DTBS/R_Protect – D.DTBS/R is protected by the environment, when the signature creation request is submitted from the SCA.
4. OE.HSM.DTBS/R_Integrity – D.DTBS/R is protected, when HSM is processing the request to create the D.serverSignatureShare.
5. OE.DTBS_Intend – SCA utilizes a method of tying the signature creation session in the SCA application with the session in the Smart-ID App TSE (see Section 2.3.3.3 for details). For example, this could be a cryptographically secured token that is transported to the Smart-ID App TSE via a URI link or a QR code; or a verification code that is

displayed to the Signer both in the SCA and Smart-ID App TSE. This guarantees that a substituted session (containing a modified D.DTBS/R) will be detected.

Therefore, the combination of the security objectives prevents the substitution of D.DTBS/R.

5.5.2.1.9 Mitigating T.Admin_Impersonation

T.Admin_Impersonation (attacker personates the privileged user of the TOE and executes the TOE's signing function for the Signer) is mitigated by OT.TSSP_Require_clientSignatureShare.

Fulfilling the security objective OT.TSSP_Require_clientSignatureShare means that even when the attacker manages to execute the TOE internal functions directly, he cannot create the D.signature for the fresh D.DTBS/R without the corresponding D.applicationSignaturePart, which can only be created with the D.clientPart, which is under the control of the Signer.

5.5.2.1.10 Mitigating T.Privileged_User_Insertion

T.Privileged_User_Insertion (Attacker is able to create D.Privileged_User including D.Reference_Privileged_User_Authentication_Data and is able to log on to the TOE as a Privileged User) is covered by the following security objectives.

1. OT.Privileged_User_Management – only a Privileged User can create a new R.Privileged_User.
2. OT.Privileged_User_Authentication – a Privileged User must be authenticated.

5.5.2.1.11 Mitigating T.Reference_Privileged_User_Authentication_Data_Modification

T.Reference_Privileged_User_Authentication_Data_Modification (an attacker modifies D.Reference_Privileged_User_Authentication_Data and is able to log on to the TOE as the Privileged User) is covered by the following security objectives.

1. OT.Privileged_User_Management – only a Privileged User can modify R.Privileged_User
2. OT.Privileged_User_Authentication – a Privileged User must be authenticated.
3. OT.Privileged_User_Protection – the data associated with a Privileged User must be protected in integrity.

5.5.2.1.12 Mitigating T.Audit_Alteration

T.Audit_Alteration (attacker attacks the audit function of the TOE or the audit log storage outside of the TOE and deletes the existing log entries, modifies the existing log entries or creates new log entires) is mitigated by OE.Protected_Auditlog, which ensures that the TOE environment protects the audit records.

5.5.2.1.13 Mitigating T.Context_Alteration

T.Context_Alteration (attacker gets the root-level or physical access to the TOE or underlying IT components and is able to modify the user's data, security attributes, and/or program code) is mitigated by the following security objectives.

1. OE.Env – The environment of the TOE provides the first-level protection against physical attacks.

2. OE.HSM.Tamper_Resistance – The HSM security objective provides protection against physical attacks and provides resistance to the tampering with the security attributes and program code protected by HSM. Because D.serverSignatureShare can only be created by HSM and D.serverSignatureShare is required to create the D.signature, the tamper resistance is extended to the D.signature as well.
3. OT.System_Protection – The TOE security objective ensures that modification of D.TSF_CONFIG_DATA is authorized by D.Privileged_User.
4. OT.TSSP_Require_clientSignatureShare – The TOE security objective means that even in the case that the attacker has managed to break other security features, the D.applicationSignaturePart is still required according to the TSSP. The D.application SignaturePart can only be created with the D.clientPart, which is under the control of the Signer.

Therefore, the combination of the abovementioned security objectives prevents physical attacks.

5.5.2.1.14 Mitigating T.Signature_Request_Disclosure

T.Signature_Request_Disclosure (Attacker obtains knowledge of D.DTBS/R or D.SAD during transfer to the TOE) is mitigated by the following security objectives.

1. OT.TSSP_End2End and OE.TSE.TSSP_End2End – The TOE and TSE shall protect the confidentiality and integrity of the communications between the TOE and Signer.
2. OT.SAP_Replay_Protection – The TOE shall protect the communications between the TOE and Signer against replay attacks.
3. OT.TSSP_Require_clientSignatureShare and OT.TSSP_Validate_clientSignatureShare – These TOE security objectives imply that even if the attacker has managed to obtain D.SAD by intercepting communications between the TOE and Signer, the knowledge-based factor D.PIN is still required for decrypting D.clientPart and giving subsequent signatures.

Therefore, the combination of the abovementioned security objectives ensures the protection of the assets of the signature creation function.

5.5.2.1.15 Mitigating T.Random

Threat T.Random (attacker guesses the random values, which are used to generate the D.SCD and is able to successfully derive the value of the D.SCD) is mitigated by following security objectives.

1. OE.TSE.SCD_Unique - The TSE security objective ensures the cryptographic quality of generated keys. This includes the cryptographic quality of the random number generator.
2. OE.HSM.SCD_Unique - The HSM security objective ensures the cryptographic quality of generated keys. This includes the cryptographic quality of the random number generator.
3. OE.HSM.Tamper_Resistance - The HSM security objective ensures that the internal random number generator of HSM cannot be influenced by attacker.

The combination of the security objectives ensures that the attacker cannot guess random values.

5.5.2.1.16 Mitigating T.SVD_Forgery

Threat T.SVD_Forgery (attacker modifies the D.SVD value, which is created by the TOE and presented to the CA for the certification of the Signer's key pair) is mitigated by OE.SVD_Authenticity, OE.CA_Request_Certificate, and OE.CGA_QCert.

The environment objective OE.SVD_Authenticity and OE.CA_Request_Certificate ensure the integrity of the SVD exported by the TOE to the CGA. The environment objective OE.CGA_QCert ensures that the CA verifies that the Signer has control over the D.SCD corresponding to the D.SVD presented for the certification.

5.5.2.2 Rationale for fulfilling organisational policy requirements

5.5.2.2.1 Fulfilling P.SCD_Confidential

P.SCD_Confidential (The confidentiality of SCD must be reasonably assured) is addressed by the following objectives:

1. OT.SCD_Confidential
2. OT.TSSP_End2End
3. OE.TSE.App_Sandbox
4. OE.TSE.TSSP_End2End
5. OE.TSE.SCD_Confidential
6. OE.HSM.SCD_Confidential
7. OE.HSM.Tamper_Resistance

In the Smart-ID system, the D.SCD consists of the three shares residing in physically separated components. In order to export D.SCD outside of the TOE environment, an attacker needs to be able to export and successfully decrypt all of the three shares together. The confidentiality of the corresponding shares of the D.SCD is assured as shown in the table 9, by securing them in transit, at rest and when in use.

Table 9. Protection of the components of the D.SCD

Component	Protection assurances		
	Data in transit	Data at rest	Data in use
D.clientPart	D.clientPart is not transmitted anywhere	D.clientPart is stored inside the mobile app sandbox, encrypted with the key derived from VAD. The OE.TSE.SCD_Confidential is defined in [9].	D.clientPart is generated and used securely inside a mobile app process that is isolated from the other apps. The OE.TSE.App_Sandbox is defined in [9].

Table 9. Protection of the components of the D.SCD

Component	Protection assurances		
	Data in transit	Data at rest	Data in use
D.serverPart	Transmitted over protected communication channel into the TOE. When in transmission, the D.serverPart is encrypted with the D.TEK. Refer to OE.TSE.TSSP_End2End and OT.TSSP_End2End.	Stored in the TOE database, wrapped with D.KWK. The OT.SCD_Confidential assures the confidentiality of this operation.	D.serverPart is generated securely inside a mobile app process that is isolated from the other apps. D.serverPart is used securely within the TOE. The OT.SCD_Confidential assures the confidentiality of this operation.
D.serverShare	Not transmitted anywhere.	Stored in the TOE database, wrapped with HSM encryption (with HSM master key). The OE.HSM.SCD_Confidential and OE.HSM.Tamper_Resistance assure the confidentiality of this operation.	D.serverShare is generated and processed in clear only in the HSM. The OE.HSM.SCD_Confidential and OE.HSM.Tamper_Resistance assure the confidentiality of this operation.

5.5.2.2.2 Fulfilling P.Sig_unForgeable

P.Sig_unForgeable (electronic signature shall be reliably protected against forgery and it shall not be possible, to derive an electronic signature from data other than the D.SCD) is addressed by OT.Sig_Secure, OE.HSM.Sig_Secure and OE.TSE.Sig_Secure, in a same way as the threat T.Signature_Forgery (attacker uses the vulnerability in the cryptographic signature algorithm and without having the copy of the D.SCD, crafts the value of the new signature for the fresh D.DTBS/R) is mitigated. Refer to the corresponding section about mitigating T.Signature_Forgery.

5.5.2.2.3 Fulfilling P.SCD_userOnly

P.SCD_userOnly (D.SCD shall be reliably protected against use by others) is addressed by the mitigation of the following threats:

1. T.Enrolment_Signer_Authentication_Data_Disclosed
2. T.Enrolment_Signer_Impersonation
3. T.SAD_Forgery
4. T.SAP_ByPass
5. T.SAP_Replay
6. T.TSSP_Modification
7. T.TSSP_Duplication

8. T.Admin_Impersonation
9. T.Context_Alteration
10. T.Random

All those threats impact the policy requirement that the D.SCD shall be reliably protected against use by others than the legitimate Signer. Refer to the individual sections about the mitigation of those threats. In summary, they are mitigated by the following security objectives:

1. OT.SCD_Confidential
2. OT.TSSP_End2End
3. OT.SAP_Replay_Protection
4. OT.TSSP_Require_clientSignatureShare
5. OT.TSSP_Validate_clientSignatureShare
6. OT.TSSP_CloneDetection
7. OT.TSSP_TimeDelay_Locks
8. OE.TSE.TSSP_End2End
9. OE.TSE.SCD_Unique
10. OE.TSE.SCD_Confidential
11. OE.TSE.App_Sandbox
12. OE.HSM.SCD_Unique
13. OE.HSM.SCD_Confidential
14. OE.HSM.Tamper_Resistance
15. OE.HSM.Sigy_SigF

5.5.2.2.4 Fulfilling P.DTBS_Integrity

P.DTBS_Integrity (the TOE and its environment shall not alter DTBS nor DTBS/R and not prevent such data from being presented to the Signer prior to signing) is addressed by the following security objectives:

1. OT.TSSP_End2End ensures that when D.DTBS/R is transmitted from TSE to TOE, the transmission is encrypted and cannot be changed.
2. OT.DTBS/R_Protect ensures that when D.DTBS/R is processed in the TOE or transmitted to another IT components D.DTBS/R is protected from substitution and modification.
3. OE.HSM.DTBS/R_Integrity ensures that D.DTBS/R is protected when processed by HSM.
4. OE.DTBS_Intend ensures that the integrity of the D.DTBS/R is verified either manually (by the Signer) or automatically and that the Signer can be sure that he is signing the correct DTBS (see Section 2.3.3.3 for details).
5. OE.DTBS/R_Protect ensures that D.DTBS/R is protected when transmitted in the TOE environment.

5.5.2.2.5 Fulfilling P.SCD_Unique

P.SCD_Unique (any given instance of a SCD shall occur only once) is addressed by OE.HSM.SCD_Unique and OE.TSE.SCD_Unique. In the Smart-ID system, the D.SCD consists of the three shares residing in physically separated components.

The D.clientPart and D.serverPart are generated in the TSE. The OE.TSE.SCD_Unique ensures that the TSE generates the key pair D.clientShare/D.clientModulus with sufficient cryptographic quality and that the probability of encountering equal values for D.clientShare is negligible.

The D.serverShare is generated in the HSM. The OE.HSM.SCD_Unique ensures that HSM generates the key pair D.serverShare/D.serverModulus with sufficient cryptographic quality and that the probability of encountering equal values for D.serverShare is negligible.

5.5.2.2.6 Fulfilling P.DTBS/R_Unique

P.DTBS/R_Unique (the electronic signature must be linked to D.DTBS in such a way that any subsequent change in the data is detectable) is addressed by OE.DTBS/R_Unique which, by the use of appropriate cryptographic techniques, ensures that it is infeasible to generate data which would correspond to a given D.DTBS/R, thus ensuring that the signed data and the electronic signature are securely linked together. Any subsequent change in the data will result in a different D.DTBS/R and is therefore detectable.

5.5.2.2.7 Fulfilling P.TSP_Qualified

P.TSP_Qualified (generating or managing the SCD may only be done by a qualified trust service provider) is addressed by OE.Env, which ensures that the TSP is audited.

5.5.2.2.8 Fulfilling P.TSP_QCert

P.TSP_QCert (the TSP must use a trustworthy CGA to generate a qualified certificate for the SVD generated by Smart-ID) is addressed by OE.CGA_QCert which ensures that the CGA generates a qualified certificate and thus confirms with the generated certificate that the SCD, corresponding to the certified SVD, is under the control of U.Signer. Signatures created by the U.Signer are uniquely linked to the U.Signer and it is possible to identify the U.Signer by the signature.

5.5.2.2.9 Fulfilling P.Reliable_Audit

P.Reliable_Audit (the TOE shall keep reliable audit records about events in the TOE) is addressed by combination of the following objectives:

1. OT.Audit_Events - ensures that audit records will be generated about the important system events.
2. OE.Protected_AuditLog - ensures that audit records are reliably timestamped and protected from modifications.
3. OE.Trusted_Timestamps - ensures that the TOE can use the operating system provided trusted timestamps.

5.5.2.2.10 Fulfilling P.SCD_Backup

P.SCD_Backup (the security of backups must be at the same level as for the original datasets) is addressed by OE.Env, which ensures that TSP secures the backups and keeps the datasets at minimum.

5.5.2.3 Rationale for fulfilling assumptions

5.5.2.3.1 Fulfilling A.CA

A.CA (the CGA protects the authenticity of the Signer's name and the SVD in the qualified certificate by an advanced electronic signature of the TSP) is addressed by OE.SVD_Authenticity, OE.CA_Request_Certificate and OE.CGA_QCert. The OE.SVD_Authenticity ensures integrity of the SVD exported by the TOE to the CGA. OE.CA_Request_Certificate ensures that the integrity of the request of the certificate including D.SVD and signer information is protected. OE.CGA_QCert ensures that the CGA generates a qualified certificate and thus confirms with the generated certificate that the SCD, corresponding to the certified SVD, is under the control of U.Signer. Signatures created by the U.Signer are uniquely linked to the U.Signer and it is possible to identify the U.Signer by the signature.

5.5.2.3.2 Fulfilling A.ACCESS_PROTECTED

A.ACCESS_PROTECTED (the TOE environment limits physical and logical access to the components in the TOE environment) is addressed by OE.Env which ensures that the TOE environment is protected and limits the exposure to physical attacks.

5.5.2.3.3 Fulfilling A.PRIVILEGED_USER

A.PRIVILEGED_USER (the U.Admin is trusted) addressed by OE.Trusted_Admin, which ensures that the U.Admin is well trained and trusted to perform his duties.

5.5.2.3.4 Fulfilling A.SIGNER_ENROLLMENT

A.SIGNER_ENROLLMENT (the signer enrolment is conformant with reg. (EU) 910/2014 [4]) addressed by OE.Env, which ensures that the TSP is audited.

5.5.2.3.5 Fulfilling A.SIGNER_AUTHENTICATION_DATA_PROTECTION

A.SIGNER_AUTHENTICATION_DATA_PROTECTION (the signer will not disclose his authentication factors) addressed by OE.Signer_Authentication_Data, which ensures that signer's management of authentication factors data outside the TOE is carried out in a secure manner.

5.5.2.3.6 Fulfilling A.SIGNER_DEVICE

A.SIGNER_DEVICE (Signer has the trusted and evaluated TSE component in his environment to help him to complete the TSSP steps for key generation and signing operations) is addressed by environment objectives marked OE.TSE.* and OE.DTBS_Intend.

5.5.2.3.7 Fulfilling A.TSP_AUDITED

A.TSP_AUDITED (TSP deploying the TOE is a qualified TSP) is addressed by OE.Env which ensures that the TOE operator is a qualified TSP.

5.5.2.3.8 Fulfilling A.CSPRNG

A.CSPRNG (HSM provides the secure random number generator) is fulfilled by OE.CSPRNG which provides a cryptographically secure random number generator.

5.5.2.3.9 Fulfilling A.CRYPTO

A.CRYPTO (endorsed algorithms, algorithm parameters and key lengths) is fulfilled by OE.CSPRNG, which provides a cryptographically secure random number generator and by OE.HSM.SCD_Unique and OE.HSM.Sig_Secure.

5.5.2.3.10 Fulfilling A.JVM

A.JVM (the TOE is the only application running on the JVM) is fulfilled by OE.Env, which ensures that the TOE is the only application deployed in the container included in System Overview.

6 Extended components definition (ASE_ECD)

There are no extended components used in SZ.

7 Security Requirements (ASE_REQ)

7.1 Data in the TOE: user data and TSF data

This section classifies the assets defined in the ASE_SPD and the security attributes used in the SFR definitions.

7.1.1 User data

Those attributes are considered 'user data' as per the definition of the CC Part 2, page 21, paragraph 36. These are the attributes to which the TOE places no special meaning and which the TOE does not use for any security related functions.

The protection of user data is handled by the access control policies defined in SFRs FDP_ACC.1 and FDP_ACF.1.

Table 10. User data attributes in the TOE

Attribute name	Corresponding asset	Storage location	Notes
DTBSR	D.DTBS/R	in memory only	The digest for the signing. Submitted by the client during the performSignature() operation (see also table 13).

7.1.2 TSF data

Rest of the data handled by the TOE is classified as 'TSF data' as per the definition of the CC Part 2, page 21, paragraph 36.

7.1.2.1 Authentication data

Following attributes in the table 11 are considered 'authentication data' as per the definition of the CC Part 2, page 21, paragraph 40. Authentication data is used to verify the claimed identity of a user requesting services from the TOE. Authentication data is used by the authentication mechanisms defined in SFRs FIA_UAU.3 and FIA_UAU.5. The authentication data itself is protected with the SFRs from the family FPT and FMT.

Table 11. Authentication data attributes in the TOE

Attribute name	Corresponding asset	Storage location	Notes
client_share_2nd_part	D.serverPart	database, wrapped	This is the other half of the D.clientShare. It is used to complete the signature share D.applicationSignatureShare.
client_modulus	D.clientModulus	database	This is the public key of the D.clientShare key pair. It is used to verify the signature share D.applicationSignatureShare.
server_modulus	D.serverModulus	database	Generated by the HSM and stored in the TOE database
composite_modulus	D.SVD	database	Computed by the TOE and stored in the TOE database
current_one_time_password	D.OTP	database	This is the next one-time password, which is expected to be sent by the TSE for the next key pair operation. D.OTP is not wrapped but is rather stored in hashed form. This value is only used for comparison.
sz_keypair_uuid	D.Signing_Key_Id	database	This is the identifier for the key pair.

7.1.2.2 Security data

Following attributes are considered 'security attributes' as per the definition of the CC Part 2, page 21, paragraph 35. Security attributes are used by TSF in order to make decisions as required by the SFRs. Security attributes are protected with the SFRs from the family FPT and FMT.

Table 12. Security attributes in the TOE

Attribute name	Corresponding asset	Storage location	Notes
DEK_symmetric_key	D.DEK	database, wrapped	This is used by the TOE to encrypt and to integrity protect some database fields. D.DEK is generated and used by the TOE itself, and it is wrapped with the D.KWK.
server_privatekey	D.serverShare	database, wrapped	Generated by the HSM and stored in the TOE database, wrapped with the HSM master key.

Table 12. Security attributes in the TOE

Attribute name	Corresponding asset	Storage location	Notes
TEK_symmetric_key	D.TEK	database, wrapped	This is generated by the TOE and TSE during the Diffie-Hellman key exchange and is afterwards used to encrypt/decrypt the messages transmitted between the TSE and TOE. HMAC portion of the key is used to provide and verify the authenticity and integrity of the messages. For storage, it is wrapped with the D.DEK.
KTK_wrapper_key	D.KTK	database, wrapped	This is generated by the admin and is used by the TOE to sign the replies to the initiateKey() operation and to allow the TSE to authenticate the TOE.
KWK_wrapper_key	D.KWK	database, wrapped	This is generated by the admin and is used by the TOE to wrap the key material in the TOE database. HMAC portion of the key is used to provide and verify the authenticity and integrity of the stored key material.
sz_keypair_state		database	The status of the key pair, for example 'IN_PREPARATION', 'READY', 'TIMELOCKED'.
locked_until_time		database	Timestamp until the key is not usable.
pin_attempts		database	Number of times the FIA_UAU.5.2/Signer authentication method has failed in a row.
DH_keyPair		ephemeral, in memory	Temporary DH key pair, which is used to generate the D.TEK. After the D.TEK is established and stored, the DH_keyPair is destroyed.

7.2 Security Function Policies (SFP)

This section defines the rules and policies for the access control decisions performed by the TSF. These policies are referenced in the SFR definitions.

The SFPs are defined in tabular form. The tables are processed from up to down. If the request parameters match with the attributes of a row in the table, the corresponding access control decision is taken. For the case where none of the previous rows matched the request, the last line is usually the wildcard match with the access control decision to deny the request.

7.2.1 Operations

First of all, an overview is provided of the possible operations which can be requested by the users and admins. Those operations correspond to the TOE API methods and further information, including the detailed list of method arguments and error conditions, can be found in the architecture documents. The table 13 gives the short summary about the user operations and table 14 lists the admin operations.

Table 13. List of operations, which can be requested by TOE users

Operation name	Description
initiateKey	This is the first method that is called by the TSE in order to enrol a new key pair with the TOE. The method establishes the D.TEK and also D.OTP, which is used in the subsequent methods.
submitClient2ndPart	This is the second method to be called by the TSE during the new key pair enrolment.
performSignature	This is the main method for creating signatures with the enrolled key pair. The TSE submits the digest to be signed, the signature part computed in the Signer's environment, signature creation parameters and other multi-factor authentication data. The TSE receives the completed signature.
reKey	This is the method to complete the re-key process, which is initiated by the CA in order to initiate the generation of a new server share of the private key and the corresponding new compound public key for the Signer.
refreshCloneDetection	This is the technical method used by the TSE to request the fresh D.OTP without creating any signature.
getKeyState	This is the technical method used by the TSE to get the status information about the key pair, for example, the remaining time until the key pair is un-locked.
getFreshnessToken	This is a technical method for ensuring that the key pair operations are performed in sequence on different cluster nodes and that they do not conflict with each other.
revokeKey	This method is used by the TSE and the CA to destroy the key pair in the TOE so that it cannot be used anymore.

Table 14. List of operations, which can be requested by TOE admins

Operation name	Description
hsmPasswordEntry	This method is used by the admin after starting the TOE, in order to load the HSM password. The HSM password is not stored in the configuration file and must be entered on each boot manually.
generateKTKKey	This method is used by the admin to generate D.KTK.
generateKWKKey	This method is used by the admin to generate D.KWK.
generateDEKKey	This method is used by the admin to generate D.DEK.
deleteKTKKey	This method is used by the admin to destroy D.KTK.
deleteKWKKey	This method is used by the admin to destroy D.KWK.
deleteDEKKey	This method is used by the admin to destroy D.DEK.
batchGenerateServerShares	This method is used by the admin to pre-generate D.serverShare assets, so that Signer enrolment can be done quicker.
deleteServerShares	This method is used by the admin to destroy a specified subset of unused D.serverShare assets.

7.2.2 SFP/Init

Table 15. Security Function Policy, which specifies the default values for the new attributes and objects created by the TOE.

Object or attribute	Operation	Default value
sz_keypair_state	initiateKey	'IN_PREPARATION'
pin_attempts	initiateKey	0

7.2.3 SFP/Signer

The SFP/Signer is regulating the access to the signature generation function of the TOE. Only Signer should have access to this function, after he has authenticated himself with knowledge-based and possession-based authentication factors.

In the following table, "objects related to authenticated D.Signing_Key_Id" refers to all of the database fields which are associated with the same identifier D.Signing_Key_Id (these fields all belong to a single Signer). Essentially, it means "objects owned by the authenticated Signer".

Table 16. Security Function Policy, which specifies when the U.User is allowed to perform the operation performSignature.

User	Subject	Role	Objects	Operation	Rule
U.User	S.Signer	R.Signer	objects related to authenticated D.Signing_Key_Id	perform-Signature	allow

Table 16. Security Function Policy, which specifies when the U.User is allowed to perform the operation performSignature.

User	Subject	Role	Objects	Operation	Rule
U.User	S.Signer	R.Signer	*	performSignature	deny
U.User	S.Signer	R.Signer	*	*	deny

In the table below, it is further specified which TSF data attributes the authenticated R.Signer can manage in the course of the allowed performSignature operation.

Table 17. TSF data attributes managed by the R.Signer.

Operation	change_default	query	modify	delete
performSignature	-	D.serverShare, D.serverModulus, D.SVD	-	-

7.2.4 SFP/App

The SFP/App is regulating access to technical functions of the TOE. The TSE uses those functions on behalf of the Signer and uses only possession-based authentication factors to authenticate himself.

Table 18. Security Function Policy, which specifies what are the access rights of the S.App.

User	Subject	Role	Objects	Operation	Rule
U.User	S.App	R.App	objects related to authenticated D.Signing_Key_Id	submitClient2ndPart	allow
U.User	S.App	R.App	objects related to authenticated D.Signing_Key_Id	reKey	allow
U.User	S.App	R.App	objects related to authenticated D.Signing_Key_Id	refreshCloneDetection	allow
U.User	S.App	R.App	other objects	submitClient2ndPart, reKey, refreshCloneDetection	deny
U.User	S.App	R.App	*	*	deny

In the table below, it is further specified which TSF data attributes the authenticated R.App can manage in the course of the allowed operations.

Table 19. TSF data attributes managed by the R.App.

Operation	change_default	query	modify	delete
submitClient-2ndPart	-	D.Signing_Key_Id, D.OTP	D.serverPart, D.OTP	-
reKey	-	D.Signing_Key_Id, D.OTP	D.serverShare, D.serverModulus, D.SVD, D.OTP	D.serverShare, D.serverModulus, D.OTP
refreshClone-Detection	-	D.Signing_Key_Id, D.OTP	D.OTP	-

7.2.5 SFP/Anonymous

The SFP/Anonymous is regulating access to technical functions of the TOE, which do not require personalised user identification/authentication and strict access control. For example, all users are permitted to enrol a new key pair and all users are permitted to query status of the key pair and get freshness tokens. Also, destruction of the key pair does not need authentication, because user may not have control of the authentication factors anymore.

This doesn't mean that the access to those methods is wide open without any security. The other components of the Smart-ID system and network devices are configured to perform the preliminary access control and the channel-based authentication is still performed by those components and devices.

The "New object with fresh D.Signing_Key_Id" means that a new keyUUID is generated, which is different from all existing keyUUIDs. Essentially, "the new object, which will be owned by the new Signer, who made the request".

Table 20. Security Function Policy, which specifies what are the access rights of the un-authenticated users.

User	Subject	Role	Objects	Operation	Rule
N/A	S.Anonymous	R.Anonymous	new object with fresh D.Signing_Key_Id	initiateKey	allow
N/A	S.Anonymous	R.Anonymous	attributes 'lockDurationSec', 'pinAttemptsLeft', 'wrongAttempts', 'status' of the object of the requested D.Signing_Key_Id	getKeyState	allow
N/A	S.Anonymous	R.Anonymous	attribute 'freshnessToken' of the object of the requested D.Signing_Key_Id	getFreshness-Token	allow

Table 20. Security Function Policy, which specifies what are the access rights of the un-authenticated users.

User	Subject	Role	Objects	Operation	Rule
N/A	S.Anonymous	R.Anonymous	attribute 'status' of the object of requested D.Signing_Key_Id	revokeKey	allow
N/A	S.Anonymous	R.Anonymous	*	*	deny

7.2.6 SFP/Admin

The SFP/Admin is regulating access to the admins.

The "new object D.serverShare not associated with any existing D.Signing_Key_Id" means that administrator can only request the generation of new and fresh D.serverShare values and cannot access any D.serverShare values which are already "in use" by some existing key pair.

Table 21. Security Function Policy, which specifies what are the access rights of the admins.

User	Subject	Role	Objects	Operation	Rule
U.Admin	S.Admin	R.Admin	in-memory OCS password	hsmPasswordEntry	allow
U.Admin	S.Admin	R.Admin	new object D.serverShare not associated with any existing D.Signing_Key_Id	batchGenerateServerShares	allow
U.Admin	S.Admin	R.Admin	new object D.KTK	generateKTKKey	allow
U.Admin	S.Admin	R.Admin	new object D.KWK	generateKWKKey	allow
U.Admin	S.Admin	R.Admin	new object D.DEK	generateDEKKey	allow
U.Admin	S.Admin	R.Admin	D.serverShare objects not associated with any existing D.Signing_Key_Id	deleteServerShares	allow
U.Admin	S.Admin	R.Admin	existing D.KTK	deleteKTKKey	allow
U.Admin	S.Admin	R.Admin	existing D.KWK	deleteKWKKey	allow
U.Admin	S.Admin	R.Admin	existing D.DEK	deleteDEKKey	allow
U.Admin	S.Admin	R.Admin	*	*	deny

In the table below, it is further specified which TSF data attributes the authenticated R.Admin can manage in the course of the allowed operations.

Table 22. TSF data attributes managed by the R.Admin.

Operation	change_default	query	modify	delete
hsmPassword-Entry	-	-	-	-
batchGenerate-ServerShares	-	-	D.serverShare	-
generateKTK-Key	-	-	D.KTK	-
generateKWK-Key	-	-	D.KWK	-
generateDEK-Key	-	-	D.DEK	-
deleteServer-Shares	-	-	-	D.serverShare
deleteKTKKey	-	-	-	D.KTK
deleteKWKKey	-	-	-	D.KWK
deleteDEKKey	-	-	-	D.DEK

7.2.7 SFP/CA

The SFP/CA is regulating access to the administrative functions, which are required by the CA. CA can call the prepareReKey and revokeKey operations on any existing key pairs.

Table 23. Security Function Policy, which specifies what are the access rights of the CA.

User	Subject	Role	Objects	Operation	Rule
U.CA	S.CA	R.CA	requested D.Signing_Key_Id	prepare-ReKey	allow
U.CA	S.CA	R.CA	requested D.Signing_Key_Id	revokeKey	allow
U.CA	S.CA	R.CA	*	*	deny

In the table below, it is further specified which TSF data attributes the authenticated R.CA can manage in the course of the allowed operations.

Table 24. TSF data attributes managed by the R.CA.

Operation	change_default	query	modify	delete
prepareReKey	-	D.Signing_Key_Id	-	-
revokeKey	-	D.Signing_Key_Id	-	D.serverShare, D.serverModulus, D.SVD, D.OTP

7.3 Security Functional Requirements

This document uses the following typograhic conventions, as suggested in the https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_41_BSI_PP_ST_Guide_pdf.pdf?__blob=publicationFile:

- Iterations of the SFRs are denoted by a slash "/" and the iteration indicator after the component, for example FCS_CKM.1/RSA_SVD.
- Refinements of security requirements made by the ST author are denoted in such a way that added words are in **bold, highlighted text** and removed words are ~~strikethrough~~.
- Selections having been made by the ST author are denoted as *italic, highlighted text* and in addition a footnote will show the original text from [2].
- Assignments having been made by the ST author are denoted in the same way as selections.

7.3.1 Security Audit (FAU)

7.3.1.1 Security audit generation (FAU_GEN.1)

7.3.1.1.1 FAU_GEN.1 – Security audit generation

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

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

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

c) *Other specifically defined auditable events:*[b]

1) *Privileged User authentication;*
2) *Signer management;*
3) *Signer authentication;*
4) *Signing key generation;*
5) *Signing key destruction;*
6) *Signing key management;*
7) *Signing key activation and usage, including the D.DTBS/R and the hash of D.signature;*
8) *Configuration initialization;*
9) *TOE administration.*

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

[b] assignment: other specifically defined auditable events

Application Note 6

The PP 419 241-2 [6] includes the "Privileged User management", which is not relevant for the TOE, because privileged users and corresponding roles are hard-coded in the static TOE configuration file.
The PP 419 241-2 [6] includes the "Change of TOE configuration", which is not relevant for the TOE, because the TOE configuration is a static text file and TOE management functions do not change the configuration.

FAU_GEN.1.2

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST:

   1) type of action performed (success or failure),
   2) identity of the role which performs the operation. [a]

[a] assignment: other audit relevant information

7.3.2 Cryptographic support (FCS)

7.3.2.1 Cryptographic key generation (FCS_CKM.1)

The FCS_CKM.1 is iterated for different types of generated keys.

7.3.2.1.1 FCS_CKM.1/RSA_SVD – Cryptographic key generation

The TOE generates the D.SVD from the shares of public key (D.clientModulus and D.serverModulus).

FCS_CKM.1.1/RSA_SVD

The TSF shall generate **D.SVD** ~~cryptographic keys~~ in accordance with a specified cryptographic key generation algorithm *TSSP compound public key generation from shares of the public key*[a] and specified cryptographic key sizes *6142, 6143, 6144, 8190, 8191, 8192, 12286, 12287, 12288, 16382, 16383, and 16384 bits*[b] that meet the following: *standard RFC8017 [14] (section 3.1) and article [5]*[c]

[a] assignment: cryptographic key generation algorithm [b] assignment: cryptographic key sizes [c] assignment: list of standards

7.3.2.1.2 FCS_CKM.1/RSA_KTK – Cryptographic key generation

The D.KTK is an RSA key pair, which is used to authenticate the TOE to the TSE, when initiating the secure channel between the TSE and TOE. TOE uses the HSM to generate and protect the key pair.

FCS_CKM.1.1/RSA_KTK	The TSF shall generate **RSA key pair D.KTK** ~~cryptographic keys in accordance~~ with a ~~specified cryptographic key generation algorithm~~ *Common Criteria certified HSM*[a] and specified cryptographic key sizes *3072 bits up to 8192 bits*[b] that meet the following: *standard RFC8017 [14]*[c]

[a] assignment: cryptographic key generation algorithm [b] assignment: cryptographic key sizes [c] assignment: list of standards

Application Note 7

The TOE is expected to use a Common Criteria certified HSM, see also OE.CSPRNG, OE.HSM.SCD_Unique and OE.HSM.Sig_Secure for key generation. Although the TSF may not generate keys itself, this SFR expresses the requirement for the TSF to invoke the HSM with the appropriate parameters whenever key generation is required.

7.3.2.1.3 FCS_CKM.1/DH_TEK – Cryptographic key generation

The D.TEK is a symmetric encryption/decryption and integrity protection key, which is used to create the secure communication channel between the TSE and TOE. D.TEK is generated with a variant of Diffie-Hellman key agreement protocols:

FCS_CKM.1.1/DH_TEK	The TSF shall generate **D.TEK** ~~cryptographic keys~~ with a specified cryptographic key generation algorithm *Diffie-Hellman station-to-station protocol and concatKDF*[a] and specified cryptographic key sizes *2048 bits up to 4096 bits*[b] that meet the following: *standards RFC2631 [20], RFC3526 [21] and SP 800-56C Rev. 2 [22] (section 4.1)*[c].

[a] assignment: cryptographic key generation algorithm [b] assignment: cryptographic key sizes [c] assignment: list of standards

7.3.2.1.4 FCS_CKM.1/AES_KWK – Cryptographic key generation

The D.KWK is a symmetric encryption/decryption and integrity protection key, which is used to wrap the key material in the TOE database. The TOE uses the HSM to generate and protect the key, therefore the reference to the Common Criteria certified HSM has been included.

FCS_CKM.1.1/AES_KWK

The TSF shall generate **D.KWK** ~~cryptographic keys in accordance~~ with a ~~specified cryptographic key generation algorithm~~ *Common Criteria certified HSM*[a] and specified cryptographic key sizes *128 bits*[b] that meet the following: *standard SP 800-133r2 [23]*[c]

[a] assignment: cryptographic key generation algorithm [b] assignment: cryptographic key sizes [c] assignment: list of standards

Application Note 8

The TOE is expected to use a Common Criteria certified HSM, see also OE.CSPRNG, OE.HSM.SCD_Unique and OE.HSM.Sig_Secure for key generation. Although the TSF may not generate keys itself, this SFR expresses the requirement for the TSF to invoke the HSM with the appropriate parameters whenever key generation is required.

7.3.2.1.5 FCS_CKM.1/AES_DEK – Cryptographic key generation

The D.DEK is symmetric encryption/decryption and integrity protection key, which is used to wrap the sensitive and confidential attributes in the TOE database. The TOE generates the D.DEK by itself, but uses the D.KWK to wrap the key for storage.

FCS_CKM.1.1/AES_DEK

The TSF shall generate **D.DEK** ~~cryptographic keys~~ in accordance with a specified cryptographic key generation algorithm *SP 800-133r2 [23] (section 5)*[a] and specified cryptographic key sizes *128 bits*[b] that meet the following: *standard SP 800-133r2 [23]*[c]

[a] assignment: cryptographic key generation algorithm [b] assignment: cryptographic key sizes [c] assignment: list of standards

7.3.2.2 Cryptographic key destruction (FCS_CKM.4)

The TOE uses same key destruction method for all kind of keys, regardless whether they are stored only in the memory of the TOE, in the database or they are encrypted by the HSM:

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a ~~specified~~ cryptographic key destruction method: *deletion with platform standard tools*[a] ~~that meets the following: [assignment: list of standards]~~.

[a] assignment: cryptographic key destruction method

Application Note 9

According to Application Note 34 in PP 419 241-2 [6], it is sufficient to describe the action taken to destroy the keys instead of referencing an external standard in FCS_CKM.4.1. Note that deletion with platform standard tools is sufficient, since all cryptographic material is already stored in encrypted form before deletion and ultimately HSM is necessary for its decryption.

7.3.2.3 Cryptographic operation (FCS_COP.1)

The FCS_COP.1 is iterated for different types of cryptographic operations. The TOE uses cryptography in multiple areas as follows.

7.3.2.3.1 FCS_COP.1/RSA_SCD – Cryptographic operation

The RSA signature generation and verification algorithm is used in two cases (the given SFR and the following SFR below). To generate the compound signature of the Signer (D.signature), the TOE uses the RSA signature computation algorithm as defined in TSSP description:

FCS_COP.1.1/RSA_SCD	The TSF shall perform *RSA signature generation*[a] in accordance with a specified cryptographic algorithm *TSSP compound signature generation from the signature shares*[b] and cryptographic key sizes *3071, 3072, 4095, 4096, 6143, 6144, 8191, and 8192 bits*[c] that meet the following: *standard RFC8017 [14] (methods RSASSA-PSS and RSASSA-PKCS1-v1_5) and article [5]*[d].

[a] assignment: list of cryptographic operations [b] assignment: cryptographic algorithm [c] assignment: cryptographic key sizes [d] assignment: list of standards

7.3.2.3.2 FCS_COP.1/RSA_Other – Cryptographic operation

In addition to Signer's signatures, the TOE also uses the RSA algorithm to perform message decryption and encryption and generation and verification of signatures, when securing the communication between the TOE and the Signer. The TOE uses the algorithms in RFC8017 [14] for that.

FCS_COP.1.1/RSA_Other	The TSF shall perform *RSA decryption, encryption, signature generation and verification*[a] in accordance with a specified cryptographic algorithm *RSASSA-PSS, RSASSA-PKCS1-v1_5 or RSAES-OAEP*[b] and cryptographic key sizes *3072 bits up to 16384 bits*[c] that meet the following: *standard RFC8017 [14] (methods RSASSA-PSS, RSASSA-PKCS1-v1_5 and RSAES-OAEP)*[d].

[a] assignment: list of cryptographic operations [b] assignment: cryptographic algorithm [c] assignment: cryptographic key sizes [d] assignment: list of standards

7.3.2.3.3 FCS_COP.1/AES – Cryptographic operation

Encryption and decryption is performed with the AES algorithm:

FCS_COP.1.1/AES	The TSF shall perform *encryption and decryption*[a] in accordance with a specified cryptographic algorithm *AES*[b] and cryptographic key sizes *128 bits or longer*[c] that meet the following: *standard FIPS 197 [24]*[d].

[a] assignment: list of cryptographic operations [b] assignment: cryptographic algorithm [c] assignment: cryptographic key sizes [d] assignment: list of standards

7.3.2.3.4 FCS_COP.1/HMAC – Cryptographic operation

Integrity protection and verification is performed with the keyed HMAC algorithm:

FCS_COP.1.1/HMAC	The TSF shall perform *integrity protection and verification*[a] in accordance with a specified cryptographic algorithm *HMAC*[b] and cryptographic key sizes *128 bits*[c] that meet the following: *standard FIPS 198-1 [25]*[d].

[a] assignment: list of cryptographic operations [b] assignment: cryptographic algorithm [c] assignment: cryptographic key sizes [d] assignment: list of standards

7.3.2.3.5 FCS_COP.1/SHA-2 – Cryptographic operation

Digest computation is performed either with the SHA-2 family of algorithms (this section) or with the SHA-3 family of algorithms (next section).

FCS_COP.1.1/SHA-2	The TSF shall perform *digest computation*[a] in accordance with a specified cryptographic algorithm *SHA-2*[b] and cryptographic key sizes *256 bits, 384 bits, 512 bits*[c] that meet the following: *standard FIPS 180-4 [26]*[d].

[a] assignment: list of cryptographic operations [b] assignment: cryptographic algorithm [c] assignment: cryptographic key sizes [d] assignment: list of standards

7.3.2.3.6 FCS_COP.1/SHA-3 – Cryptographic operation

FCS_COP.1.1/SHA-3	The TSF shall perform *digest computation*[a] in accordance with a specified cryptographic algorithm *SHA-3*[b] and cryptographic key sizes *256 bits, 384 bits, 512 bits*[c] that meet the following: *standard FIPS 202 [27]*[d].

[a] assignment: list of cryptographic operations [b] assignment: cryptographic algorithm [c] assignment: cryptographic key sizes [d] assignment: list of standards

7.3.3 User data protection (FDP)

7.3.3.1 Access control policy and rules (FDP_ACC.1 and FDP_ACF.1)

7.3.3.1.1 FDP_ACC.1/Signer – Subset access control

FDP_ACC.1.1/Signer	The TSF shall enforce the *SFP/Signer*[a] on *list of subjects, objects and operations as specified in the table 16 in section 7.2.3 - Security Requirements (ASE_REQ)*[b].

[a] assignment: access control SFP [b] assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP

7.3.3.1.2 FDP_ACF.1/Signer – Security attribute based access control

FDP_ACF.1.1/Signer	The TSF shall enforce the *SFP/Signer*[a] to objects based on the following: *list of rules as specified in the table 16 in section 7.2.3 - Security Requirements (ASE_REQ)*[b].

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

FDP_ACF.1.2/Signer	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *list of rules as specified in the table 16 in section 7.2.3 - Security Requirements (ASE_REQ)*[a].

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

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

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

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

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

7.3.3.1.3 FDP_ACC.1/App – Subset access control

FDP_ACC.1.1/App	The TSF shall enforce the *SFP/App*[a] on *list of subjects, objects and operations as specified in the table 18 in section 7.2.4 - Security Requirements (ASE_REQ)*[b].

[a] assignment: access control SFP

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

7.3.3.1.4 FDP_ACF.1/App – Security attribute based access control

FDP_ACF.1.1/App	The TSF shall enforce the *SFP/App*[a] to objects based on the following: *list of rules as specified in the table 18 in section 7.2.4 - Security Requirements (ASE_REQ)*[b].

[a] assignment: access control SFP

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

FDP_ACF.1.2/App	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *list of rules as specified in the table 18 in section 7.2.4 - Security Requirements (ASE_REQ)*[a].

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

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

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

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

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

7.3.3.1.5 FDP_ACC.1/Anonymous – Subset access control

FDP_ACC.1.1/Anonymous	The TSF shall enforce the *SFP/Anonymous*[a] on *list of subjects, objects and operations as specified in the table 20 in section 7.2.5 - Security Requirements (ASE_REQ)*[b].

[a] assignment: access control SFP

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

7.3.3.1.6 FDP_ACF.1/Anonymous – Security attribute based access control

FDP_ACF.1.1/Anonymous	The TSF shall enforce the *SFP/Anonymous*[a] to objects based on the following: *list of rules as specified in the table 20 in section 7.2.5 - Security Requirements (ASE_REQ)*[b].

[a] assignment: access control SFP

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

FDP_ACF.1.2/Anonymous	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *list of rules as specified in the table 20 in section 7.2.5 - Security Requirements (ASE_REQ)*[a].

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

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

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

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

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

7.3.3.1.7 FDP_ACC.1/Admin – Subset access control

FDP_ACC.1.1/Admin The TSF shall enforce the *SFP/Admin* [a] on *list of subjects, objects and operations as specified in the table 21 in section 7.2.6 - Security Requirements (ASE_REQ)* [b].

[a] assignment: access control SFP [b] assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP

7.3.3.1.8 FDP_ACF.1/Admin – Security attribute based access control

FDP_ACF.1.1/Admin The TSF shall enforce the *SFP/Admin* [a] to objects based on the following: *list of rules as specified in the table 21 in section 7.2.6 - Security Requirements (ASE_REQ)* [b].

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

FDP_ACF.1.2/Admin The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *list of rules as specified in the table 21 in section 7.2.6 - Security Requirements (ASE_REQ)* [a].

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

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

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

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

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

7.3.3.1.9 FDP_ACC.1/CA – Subset access control

FDP_ACC.1.1/CA The TSF shall enforce the *SFP/CA*[a] on *list of subjects, objects and operations as specified in the table 23 in section 7.2.7 - Security Requirements (ASE_REQ)*[b].

[a] assignment: access control SFP [b] assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP

7.3.3.1.10 FDP_ACF.1/CA – Security attribute based access control

FDP_ACF.1.1/CA The TSF shall enforce the *SFP/CA*[a] to objects based on the following: *list of rules as specified in the table 23 in section 7.2.7 - Security Requirements (ASE_REQ)*[b].

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

FDP_ACF.1.2/CA The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *list of rules as specified in the table 23 in section 7.2.7 - Security Requirements (ASE_REQ)*[a].

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

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

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

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

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

7.3.4 Identification and authentication (FIA)

7.3.4.1 Authentication failure handling (FIA_AFL)

7.3.4.1.1 FIA_AFL.1 – Authentication failure handling

Authentication failure handling is defined for the following authentication events:

FIA_AFL.1.1	The TSF shall detect when *a TOE administrator configurable number of (within range 3..18)*[a] unsuccessful authentication attempts occur related to *Signer authentication with knowledge-based authentication factor*[b].

[a] selection: [assignment: positive integer number, a TOE administrator configurable positive integer within [assignment: range of acceptable values] [b] assignment: list of authentication events

FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been *surpassed*[a], the TSF shall *disable the user account*[b]. **Additionally, the TOE administrator can configure the user account to become locked for a specific number of hours after a certain number of unsuccessful authentication attempts have occured.**

[a] selection: met, surpassed [b] assignment: list of actions

7.3.4.2 Timing of identification and authentication (FIA_UID.1 and FIA_UAU.1)

Some TOE functions can be accessed without identification and authentication, as shown in the following sections:

7.3.4.2.1 FIA_UID.1 – Timing of identification

FIA_UID.1.1	The TSF shall allow *operations 'initiateKey', 'getKeyState', 'getFreshnessToken', 'revokeKey'*[a] on behalf of the user to be performed before the user is identified.

[a] assignment: list of TSF-mediated actions

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

7.3.4.2.2 FIA_UAU.1 – Timing of authentication

FIA_UAU.1.1	The TSF shall allow *operations 'initiateKey', 'getKeyState', 'getFreshnessToken', 'revokeKey'*[a] on behalf of the user to be performed before the user is authenticated.

[a] assignment: list of TSF-mediated actions

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

Application Note 10

FIA_UAU.1 requires all users to be authenticated including U.Admin and U.CA as well.

7.3.4.3 Multifactor unforgeable authentication (FIA_UAU.3 and FIA_UAU.4)

7.3.4.3.1 FIA_UAU.3 - Unforgeable authentication

FIA_UAU.3.1	The TSF shall *prevent*[a] use of authentication data that has been forged by any user of the TSF.

[a] selection: detect, prevent

FIA_UAU.3.2	The TSF shall *detect*[a] use of authentication data that has been copied from any other user of the TSF.

[a] selection: detect, prevent

Application Note 11

Note that the SFR FIA_UAU.3 has been traditionally used with biometric authentication in the context where the TSF shall be able to detect the forged biometric data. In our case, the TSF is able to detect copied one-time passwords and forged digital signatures.

7.3.4.3.2 FIA_UAU.4/Signer - Single-use authentication mechanisms

FIA_UAU.4.1/Signer	The TSF shall prevent reuse of authentication data related to *Signer authentication*[a].

[a] assignment: identified authentication mechanism(s)

7.3.4.3.3 FIA_UAU.4/App - Single-use authentication mechanisms

FIA_UAU.4.1/App	The TSF shall prevent reuse of authentication data related to *App authentication*[a].

[a] assignment: identified authentication mechanism(s)

Application Note 12

The authentication methods, which are used to authenticate Signers and Apps, use one-time passwords and the TSF can prevent the re-use of the old passwords.

7.3.4.3.4 FIA_UAU.5/Signer - Multiple authentication mechanisms

FIA_UAU.5.1/Signer	The TSF shall provide *knowledge-based and possession-based authentication mechanism*[a] to support **U.User** ~~user~~ authentication.

[a] assignment: list of multiple authentication mechanisms

FIA_UAU.5.2/Signer

The TSF shall authenticate U.User's ~~any user's~~ claimed identity according to the *following input information and algorithm:*[a]

1. claimed D.Signing_Key_Id value (to identify the user)
2. transmitted D.OTP (possession-based factor)
3. JWE envelope encrypted with correct D.TEK (possession-based factor)
4. transmitted D.applicationSignaturePart computed on D.DTBS/R with D.clientPart, decrypted with the correct D.PIN (knowledge-based factor)

authentication algorithm works as follows:

1. The TOE receives the operation performSignature() request and parses the JWE envelope.
2. The TOE takes the claimed D.Signing_Key_Id value from the JWE header and retrieves the D.OTP and D.TEK of the corresponding D.Signing_Key_Id object from the database.
3. The TOE verifies that the JWE envelope is encrypted with the same D.TEK and decrypts the envelope contents.
4. The TOE verifies that the D.OTP inside the envelope and the D.OTP from the database match.
5. The TOE retrieves the D.serverPart and D.clientModulus of the corresponding D.Signing_Key_Id from the database, computes the D.serverSignaturePart on the presented D.DTBS/R. The TOE then combines D.applicationSignaturePart and D.serverSignaturePart to the D.applicationSignatureShare and verifies it with D.clientModulus.

In case the steps 3, 4 and 5 give a positive match, the authentication result is positive, the TOE binds U.User with subject S.Signer and role R.Signer. S.Signer is identified with the value of D.Signing_Key_Id.

[a] assignment: rules describing how the multiple authentication mechanisms provide authentication

7.3.4.3.5 FIA_UAU.5/App - Multiple authentication mechanisms

FIA_UAU.5.1/App

The TSF shall provide *possession-based encryption key and one-time password authentication mechanism*[a] to support U.User ~~user~~ authentication.

[a] assignment: list of multiple authentication mechanisms

FIA_UAU.5.2/App

The TSF shall authenticate **U.User’s** ~~any user’s~~ claimed identity according to the *following input information and algorithm:*[a]

1. **claimed D.Signing_Key_Id value**
2. **transmitted D.OTP (possession-based factor)**
3. **JWE envelope encrypted with correct D.TEK (possession-based factor)**

authentication algorithm works as follows:

1. **The TOE receives the operation performSignature() request and parses the JWE envelope.**
2. **The TOE takes the claimed D.Signing_Key_Id value from the JWE header and retrieves the D.OTP and D.TEK of the corresponding D.Signing_Key_Id object from the database.**
3. **The TOE verifies that the JWE envelope is encrypted with the same D.TEK and decrypts the envelope contents.**
4. **The TOE verifies that the D.OTP inside the envelope and the D.OTP from the database match.**

In case the steps 3 and 4 give a positive match, the authentication result is positive, the TOE binds U.User with subject S.App and role R.App. S.App is identified with the value of D.Signing_Key_Id.

[a] assignment: rules describing how the multiple authentication mechanisms provide authentication

7.3.5 Security Management (FMT)

7.3.5.1 Management of security attributes (FMT_MSA)

7.3.5.1.1 FMT_MSA.1/Signer – Management of security attributes

FMT_MSA.1.1/Signer

The TSF shall enforce the *SFP/Signer*[a] to restrict the ability to *query*[b] the security attributes *listed in the section 7.2.3 – Security Requirements (ASE_REQ) , in Table 17*[c] to *role R.Signer*[d].

[a] assignment: access control SFP(s), information flow control SFP(s) [b] selection: change_default, query, modify, delete, [assignment: other operations] [c] assignment: list of security attributes [d] assignment: the authorised identified roles

7.3.5.1.2 FMT_MSA.1/App – Management of security attributes

FMT_MSA.1.1/App	The TSF shall enforce the *SFP/App*[a] to restrict the ability to *query, modify, delete*[b] the security attributes *listed in the section 7.2.4 – Security Requirements (ASE_REQ) , in Table 19*[c] to *role R.App*[d].

[a] assignment: access control SFP(s), information flow control SFP(s) [b] selection: change_default, query, modify, delete, [assignment: other operations] [c] assignment: list of security attributes [d] assignment: the authorised identified roles

7.3.5.1.3 FMT_MSA.1/Admin – Management of security attributes

FMT_MSA.1.1/Admin	The TSF shall enforce the *SFP/Admin*[a] to restrict the ability to *modify*[b] the security attributes *listed in the section 7.2.6 – Security Requirements (ASE_REQ) , in Table 22*[c] to *role R.Admin*[d].

[a] assignment: access control SFP(s), information flow control SFP(s) [b] selection: change_default, query, modify, delete, [assignment: other operations] [c] assignment: list of security attributes [d] assignment: the authorised identified roles

7.3.5.1.4 FMT_MSA.1/CA – Management of security attributes

FMT_MSA.1.1/CA	The TSF shall enforce the *SFP/CA*[a] to restrict the ability to *query, delete*[b] the security attributes *listed in the section 7.2.7 – Security Requirements (ASE_REQ) , in Table 24*[c] to *role R.CA*[d].

[a] assignment: access control SFP(s), information flow control SFP(s) [b] selection: change_default, query, modify, delete, [assignment: other operations] [c] assignment: list of security attributes [d] assignment: the authorised identified roles

7.3.5.1.5 FMT_MSA.2 – Secure security attributes

FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for *attributes listed in the section 7.1.2 – Security Requirements (ASE_REQ)*[a].

[a] assignment: list of security attributes

7.3.5.1.6 FMT_MSA.3 – Static attribute initialisation

FMT_MSA.3.1	The TSF shall enforce the *SFP/Init*[a] to provide *restrictive*[b] default values for security attributes that are used to enforce the SFP.

[a] assignment: access control SFP(s), information flow control SFP(s) [b] selection, choose one of: restrictive, permissive, [assignment: other property]

FMT_MSA.3.2	The TSF shall allow *no role*[a] to specify alternative initial values to override the default values when an object or information is created.

[a] assignment: the authorised identified roles

7.3.5.2 Management of TSF config data (FMT_MTD)

7.3.5.2.1 FMT_MTD.1 – Management of TSF data

FMT_MTD.1.1	The TSF shall restrict the ability to *modify*[a] the *D.TSF_CONFIG_DATA*[b] to *R.Admin*[c]

[a] selection: change_default, query, modify, delete, clear, [assignment: other operations] [b] assignment: list of TSF data [c] assignment: the authorised identified roles

7.3.5.3 Specification of management functions (FMT_SMF)

7.3.5.3.1 FMT_SMF.1 – Specification of Management Functions

FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: *listed operations in the section 7.2.1 – Security Requirements (ASE_REQ), table 14*[a].

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

7.3.5.4 Security management roles (FMT_SMR)

7.3.5.4.1 FMT_SMR.1 – Security roles

FMT_SMR.1.1	The TSF shall maintain the roles *R.Signer, R.App, R.Admin, R.CA, R.Anonymous*[a].

[a] assignment: the authorised identified roles

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

7.3.6 Protection of the TSF (FPT)

7.3.6.1 Confidentiality and integrity of transmitted TSF data (FPT_ITC and FPT_ITI)

7.3.6.1.1 FPT_ITC.1 – Inter-TSF confidentiality during transmission

FPT_ITC.1.1	The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorised disclosure during transmission.

7.3.6.1.2 FPT_ITI.1 – Inter-TSF detection of modification

FPT_ITI.1.1	The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric: *HMAC integrity protection*[a].

[a] assignment: a defined modification metric

FPT_ITI.1.2	The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform *operation abortion*[a] if modifications are detected.

[a] assignment: action to be taken

7.3.6.2 Replay detection (FPT_RPL)

7.3.6.2.1 FPT_RPL.1 – Replay detection

FPT_RPL.1.1	The TSF shall detect replay for the following entities: *Signer*[a].

[a] assignment: list of identified entities

FPT_RPL.1.2	The TSF shall perform *key pair destroying*[a] when replay is detected.

[a] assignment: list of specific actions

7.3.7 Trusted path (FTP)

7.3.7.1 Confidentiality and integrity of transmitted TSF data (FTP_ITC)

7.3.7.1.1 FTP_ITC.1 – Inter-TSF trusted channel

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

FTP_ITC.1.2	The TSF shall permit *the TSF*[a] to initiate communication via the trusted channel.

[a] selection: the TSF, another trusted IT product

FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for *operations with database and operations with HSM*[a].

[a] assignment: list of functions for which a trusted channel is required

7.3.7.2 Confidentiality and integrity of communication with users (FTP_TRP)

7.3.7.2.1 FTP_TRP.1 – Trusted path

FTP_TRP.1.1	The TSF shall provide a communication path between itself and **Signer** *remote*[a] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification, disclosure, replay attack*[b].

[a] selection: remote, local [b] selection: modification, disclosure, [assignment: other types of integrity or confidentiality violation]

FTP_TRP.1.2	The TSF shall permit **Signer** *remote users*[a] to initiate communication via the trusted path.

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

FTP_TRP.1.3	The TSF shall require the use of the trusted path for *all operations requested by users*[a].

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

7.4 Security Requirements Rationale

7.4.1 Mapping between SFRs and TOE Security Objectives

The mapping of TOE Security Objectives to SFRs is shown in the table 25.

Table 25. Mapping between TOE security objectives and SFRs

	OT.SCD_Confidential	OT.Sig_Secure	OT.SCD/SVD_Corresp	OT.TSSP_End2End	OT.SAP_Replay_Protection	OT.TSSP_Require_clientSignatureShare	OT.TSSP_Validate_clientSignatureShare	OT.TSSP_CloneDetection	OT.TSSP_TimeDelay_Locks	OT.DTBS/R_Protect	OT.System_Protection	OT.Audit_Events	OT.Privileged_User_Management	OT.Privileged_User_Authentication	OT.Privileged_User_Protection
FAU_GEN.1												X			
FCS_CKM.1/RSA_SVD			X												
FCS_CKM.1/RSA_KTK				X											
FCS_CKM.1/DH_TEK				X											
FCS_CKM.1/AES_KWK	X														
FCS_CKM.1/AES_DEK	X							X	X						
FCS_CKM.4	X			X											
FCS_COP.1/RSA_SCD		X	X			X									
FCS_COP.1/RSA_Other				X											
FCS_COP.1/AES	X			X	X										
FCS_COP.1/HMAC	X			X	X										
FCS_COP.1/SHA-2	X	X		X	X										

Table 25. Mapping between TOE security objectives and SFRs

	OT.SCD_Confidential	OT.Sig_Secure	OT.SCD/SVD_Corresp	OT.TSSP_End2End	OT.SAP_Replay_Protection	OT.TSSP_Require_clientSignatureShare	OT.TSSP_Validate_clientSignatureShare	OT.TSSP_CloneDetection	OT.TSSP_TimeDelay_Locks	OT.DTBS/R_Protect	OT.System_Protection	OT.Audit_Events	OT.Privileged_User_Management	OT.Privileged_User_Authentication	OT.Privileged_User_Protection
FCS_COP.1/SHA-3	X	X		X	X										
FDP_ACC.1/Signer FDP_ACF.1/Signer						X	X								
FDP_ACC.1/App FDP_ACF.1/App							X	X							
FDP_ACC.1/Anonymous FDP_ACF.1/Anonymous								X							
FDP_ACC.1/Admin FDP_ACF.1/Admin													X	X	X
FDP_ACC.1/CA FDP_ACF.1/CA		X													
FIA_AFL.1									X						
FIA_UID.1 FIA_UAU.1				X	X	X	X	X	X						
FIA_UAU.3							X								
FIA_UAU.4/Signer					X			X							
FIA_UAU.4/App					X			X							
FIA_UAU.5/Signer				X		X	X	X	X						
FIA_UAU.5/App								X							
FMT_MSA.1/Signer				X	X			X	X						
FMT_MSA.1/App				X	X			X	X						
FMT_MSA.1/Admin													X		X
FMT_MSA.1/CA													X		X
FMT_MSA.2	X	X		X	X										
FMT_MSA.3	X			X					X						
FMT_MTD.1											X				
FMT_SMF.1													X		
FMT_SMR.1													X		X
FPT_RPL.1					X			X							
FPT_ITC.1 FPT_ITI.1 FTP_ITC.1	X	X		X	X					X					

Table 25. Mapping between TOE security objectives and SFRs

	OT.SCD_Confidential	OT.Sig_Secure	OT.SCD/SVD_Corresp	OT.TSSP_End2End	OT.SAP_Replay_Protection	OT.TSSP_Require_clientSignatureShare	OT.TSSP_Validate_clientSignatureShare	OT.TSSP_CloneDetection	OT.TSSP_TimeDelay_Locks	OT.DTBS/R_Protect	OT.System_Protection	OT.Audit_Events	OT.Privileged_User_Management	OT.Privileged_User_Authentication	OT.Privileged_User_Protection
FTP_TRP.1				X	X					X					

7.4.2 SFR Rationale

Here we present the rationale about the satisfaction of security objectives for the TOE by TOE SFRs.

Fulfilling OT.SCD_Confidential

OT.SCD_Confidential (the TOE shall keep the D.serverPart components of the D.SCD confidential) is addressed by the following SFRs.

- FCS_CKM.1/AES_KWK ensures that all keys stored in the database are protected in integrity.
- FCS_CKM.1/AES_DEK ensures that all confidential data that is stored in the database is protected in integrity.
- FCS_CKM.4 ensures that all the keys used for securing the data are destroyed securely.
- FCS_COP.1/AES ensures that encryption and decryption of confidential data is performed with the AES algorithm.
- FCS_COP.1/HMAC ensures that integrity protection and verification is performed with the HMAC algorithm.
- FCS_COP.1/SHA-2 and FCS_COP.1/SHA-3 ensure that digest computation is performed with either the SHA-2 or SHA-3 family of algorithms.
- FMT_MSA.2 and FMT_MSA.3 ensure that only secure values are accepted for the security attributes.
- FTP_ITC.1, FPT_ITI.1, and FPT_ITC.1 ensure integrity and confidentiality protection during transmission of data.

Fulfilling OT.Sig_Secure

OT.Sig_Secure (electronic signatures generated by the TOE must not be forgeable without D.SCD and it must not be possible to reconstruct D.SCD from digital signatures) is addressed by the following SFRs.

- FCS_COP.1/RSA_SCD ensures that the TOE generates the compound signature of the Signer (D.signature) with sufficient cryptographic strength. The TOE uses the RSA signature computation algorithm.
- FCS_COP.1/SHA-2 and FCS_COP.1/SHA-3 ensure that digest computation is performed with either the SHA-2 or SHA-3 family of algorithms.
- FDP_ACC.1/CA and FDP_ACF.1/CA ensure that CA can revoke key in the case when it's needed.
- FMT_MSA.2 ensures that only secure values are accepted for the security attributes.
- FTP_ITC.1, FPT_ITI.1, and FPT_ITC.1 ensure integrity and confidentiality protection during transmission of data.

Fulfilling OT.SCD/SVD_Corresp
OT.SCD/SVD_Corresp (the TOE shall guarantee the correspondence between the D.SVD and the D.SCD) is addressed by the following SFR.

- FCS_CKM.1/RSA_SVD and FCS_COP.1/RSA_SCD ensure that the TOE generates the D.SVD from the shares of public key (D.clientModulus and D.serverModulus) and performs signature generation using an algorithm (the TSSP protocol) that meets the standard RFC8017 [14] (see [5] for details).

Fulfilling OT.TSSP_End2End
OT.TSSP_End2End (the TOE shall protect the confidentiality and integrity of the communications between the TOE and Signer) is addressed by the following SFRs.

- FCS_CKM.1/RSA_KTK ensures the authentication of the TOE to the TSE library.
- FCS_CKM.1/DH_TEK ensures the existence of a secure communication channel between the TOE and the TSE library.
- FCS_CKM.4 ensures that all the keys used for securing the data are destroyed securely.
- FCS_COP.1/RSA_Other ensures that the TOE produces technical signatures to secure the communication between the TOE and Signer
- FCS_COP.1/AES ensures that encryption and decryption of confidential data is performed with the AES algorithm.
- FCS_COP.1/HMAC ensures that integrity protection and verification is performed with the HMAC algorithm.
- FCS_COP.1/SHA-2 and FCS_COP.1/SHA-3 ensure that digest computation is performed with either the SHA-2 or SHA-3 family of algorithms.
- FIA_UID.1 and FIA_UAU.1 ensure that the user is identified and/or authenticated before each operation if needed.
- FIA_UAU.5/Signer ensures that the signer authenticates itself.
- FMT_MSA.1/Signer ensures that only an authenticated Signer can use the TSF data belonging to him.
- FMT_MSA.1/App ensures that the App can only manage specific TSF data belonging to the given User.
- FMT_MSA.2 and FMT_MSA.3 ensure that only secure values are accepted for the security attributes.
- FTP_ITC.1, FPT_ITI.1, and FPT_ITC.1 ensure integrity and confidentiality protection during transmission of data.

- FTP_TRP.1 ensures that a trusted path is used for communication.

Fulfilling OT.SAP_Replay_Protection
OT.SAP_Replay_Protection (the TOE shall protect the communications between the TOE and Signer against replay attacks) is addressed by the following SFRs.

- FCS_COP.1/AES ensures that encryption and decryption of confidential data are performed with the AES algorithm.
- FCS_COP.1/HMAC ensures that integrity protection and verification is performed with the HMAC algorithm.
- FCS_COP.1/SHA-2 and FCS_COP.1/SHA-3 ensure that digest computation is performed with either the SHA-2 or SHA-3 family of algorithms.
- FIA_UID.1 and FIA_UAU.1 ensure that the user is identified and/or authenticated before each operation if needed.
- FIA_UAU.4/Signer ensures that there is no reuse of the signer authentication data.
- FIA_UAU.4/App ensures that there is no reuse of the app authentication data.
- FMT_MSA.1/Signer ensures that only an authenticated Signer can use the TSF data belonging to him.
- FMT_MSA.1/App ensures that the App can only manage specific TSF data belonging to the given User.
- FMT_MSA.2 ensures that only secure values are accepted for the security attributes.
- FPT_RPL.1 ensures there is no replay of D.OTP.
- FTP_ITC.1, FPT_ITI.1, and FPT_ITC.1 ensure integrity and confidentiality protection during transmission of data.
- FTP_TRP.1 ensures that a trusted path is used for communication.

Fulfilling OT.TSSP_Require_clientSignatureShare
OT.TSSP_Require_clientSignatureShare (the TOE shall protect the signature creation function of the TOE by following the TSSP and requiring the valid D.applicationSignatureShare in order to create the D.signature) is addressed by the following SFRs.

- FCS_COP.1/RSA_SCD ensures that the TOE generates the compound signature of the Signer (D.signature) as defined in the TSSP and with sufficient cryptographic strength.
- FDP_ACC.1/Signer and FDP_ACF.1/Signer ensure that the Signer must provide a valid D.applicationSignatureShare in order to authorize and complete the signature creation operation.
- FIA_UID.1 and FIA_UAU.1 ensure that the user is identified and/or authenticated before each operation if needed.
- FIA_UAU.5/Signer ensures that the Signer is authenticated.

Fulfilling OT.TSSP_Validate_clientSignatureShare
OT.TSSP_Validate_clientSignatureShare (the TOE shall protect the signature creation function of the TOE by following the TSSP and validating the D.applicationSignatureShare) is addressed by the following SFRs.

- FDP_ACC.1/Signer and FDP_ACF.1/Signer ensure that the Signer must provide a valid D.applicationSignatureShare in order to authorize and complete the signature creation operation.

- FDP_ACC.1/App and FDP_ACF.1/App ensure that the App can upload the D.clientModulus to SZ.
- FIA_UID.1 and FIA_UAU.1 ensure that the user is identified and/or authenticated before each operation if needed.
- FIA_UAU.3 ensures that the authentication data was not forged.
- FIA_UAU.5/Signer ensures that the Signer is authenticated.

Fulfilling OT.TSSP_CloneDetection
OT.TSSP_CloneDetection (the TOE shall protect the signature creation function by detecting the usage of incorrect D.OTP in signature creation requests with valid D.applicationSignature Share) is addressed by the following SFRs.

- FCS_CKM.1/AES_DEK ensures that all confidential data that is stored in the database (including D.OTP) is protected in integrity.
- FDP_ACC.1/App and FDP_ACF.1/App ensure that the App can upload D.OTP to SZ and obtain a fresh D.OTP.
- FIA_UID.1 and FIA_UAU.1 ensure that the user is identified and/or authenticated before each operation if needed.
- FIA_UAU.4/Signer ensures there is no reuse of the signer authentication data.
- FIA_UAU.4/App ensures there is no reuse of the app authentication data.
- FIA_UAU.5/Signer ensures that the Signer is authenticated.
- FIA_UAU.5/App ensures that the App is authenticated.
- FMT_MSA.1/Signer ensures that only an authenticated Signer can use the TSF data belonging to him.
- FMT_MSA.1/App ensures that the App can only manage specific TSF data belonging to the given User.
- FPT_RPL.1 ensures there is no replay of D.OTP.
- FDP_ACC.1/Anonymous and FDP_ACF.1/Anonymous ensure that a new freshness token can be queried.

Fulfilling OT.TSSP_TimeDelay_Locks
OT.TSSP_TimeDelay_Locks (the TOE shall apply time-delay between accepting new requests after submission of incorrect D.applicationSignatureShare and initiate revocation of the Signer's certificate after the limit of incorrect D.applicationSignatureShare submissions has been reached) is addressed by the following SFRs.

- FCS_CKM.1/AES_DEK ensures that all confidential data that is stored in the database is protected in integrity.
- FIA_AFL.1 ensures that the necessary action is taken (time-locking / disabling the account) after a defined number of unsuccessful Signer authentication attempts has been reached.
- FIA_UID.1 and FIA_UAU.1 ensure that the user is identified and/or authenticated before each operation if needed.
- FIA_UAU.5/Signer ensures that incorrect login attempts by the Signer are detected.
- FMT_MSA.1/Signer ensures that only an authenticated Signer can use the TSF data belonging to him.

- FMT_MSA.1/App ensures that the App can only manage specific TSF data belonging to the given User.
- FMT_MSA.3 ensures that only secure values are accepted for security attributes.

Fulfilling OT.DTBS/R_Protect
OT.DTBS/R_Protect (the TOE shall protect the D.DTBS/R from substitution and modification) is addressed by the following SFRs.

- FTP_ITC.1, FPT_ITI.1, and FPT_ITC.1 ensure the integrity and confidentiality of the D.DTBS/R when transmitted to/from external IT components.
- FTP_TRP.1 ensures the confidentiality of the data by providing a trusted communication path between the TOE and remote users.

Fulfilling OT.System_Protection
FMT_MTD.1 ensures that D.TSF_CONFIG_DATA can only be modified by R.Admin.

Fulfilling OT.Audit_Events
FAU_GEN.1 ensures that the TOE creates audit records about the important system events.

Fulfilling OT.Privileged_User_Management
OT.Privileged_User_Management (the TOE shall ensure that any modification to D.Privileged_User and D.Reference_Privileged_User_Authentication_Data are performed under the control of a Privileged User) is addressed by the following SFRs.

- FDP_ACC.1/Admin and FDP_ACF.1/Admin enforce the access control policy for U.Admin as described by SFP/Admin.
- FMT_MSA.1/Admin ensures that the security attributes listed in SFP/Admin can only be modified by the role R.Admin.
- FMT_MSA.1/CA ensures that the security attributes listed in SFP/CA can only be deleted by the role R.CA.
- FMT_SMF.1 ensures that the Privileged users app can execute their associated operations.
- FMT_SMR.1 ensures the maintenance of Privileged roles and associates the Privileged users with roles.

Fulfilling OT.Privileged_User_Authentication
OT.Privileged_User_Authentication (the TOE shall ensure that an administrator as a Privileged User is authenticated before any action on the TOE is performed) is addressed by the following SFRs.

- FDP_ACC.1/Admin and FDP_ACF.1/Admin enforce the access control policy for U.Admin as described by SFP/Admin.

Fulfilling OT.Privileged_User_Protection
OT.Privileged_User_Protection (the TOE shall ensure that data associated with D.Privileged_User are protected in integrity and if needed, in confidentiality) is addressed by the following SFRs.

- FDP_ACC.1/Admin and FDP_ACF.1/Admin enforce the access control policy for U.Admin as described by SFP/Admin.

- FMT_MSA.1/Admin ensures that the security attributes listed in SFP/Admin can only be modified by the role R.Admin.
- FMT_MSA.1/CA ensures that the security attributes listed in SFP/CA can only be deleted by the role R.CA.
- FMT_SMR.1 ensures the maintenance of Privileged roles and associates the Privileged users with roles.

7.4.3 SFR Dependencies Analysis

Table 26 shows how the dependencies of the SFRs are fulfilled.

Meaning of the wildcards in the SFR iteration are the followings:

- FCS_CKM.1/* = (FCS_CKM.1/RSA_SVD, FCS_CKM.1/RSA_KTK, FCS_CKM.1/DH_TEK, FCS_CKM.1/AES_KWK, FCS_CKM.1/AES_DEK)
- FCS_COP.1/* = (FCS_COP.1/RSA_SCD, FCS_COP.1/RSA_Other, FCS_COP.1/AES, FCS_COP.1/HMAC, FCS_COP.1/SHA-2, FCS_COP.1/SHA-3)
- FDP_ACC.1/* = (FDP_ACC.1/Signer, FDP_ACC.1/App, FDP_ACC.1/Anonymous, FDP_ACC.1/Admin, FDP_ACC.1/CA)
- FDP_ACF.1/* = (FDP_ACF.1/Signer, FDP_ACF.1/App, FDP_ACF.1/Anonymous, FDP_ACF.1/Admin, FDP_ACF.1/CA)
- FIA_UAU.4/* = (FIA_UAU.4/Signer, FIA_UAU.4/App)
- FIA_UAU.5/* = (FIA_UAU.5/Signer, FIA_UAU.5/App)
- FMT_MSA.1/* = (FMT_MSA.1/Signer, FMT_MSA.1/App, FMT_MSA.1/Admin, FMT_MSA.1/CA)

Table 26. Analysis of fulfillment of SFR dependencies

SFR	Dependecies	Fulfilled by
FAU_GEN.1	FPT_STM.1	(See application note 13)
FCS_CKM.1/*	FCS_CKM.2 or FCS_COP.1	FCS_COP.1/*
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1/*
FCS_COP.1/*	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1/*
	FCS_CKM.4	FCS_CKM.4
FDP_ACC.1/*	FDP_ACF.1	FDP_ACF.1/*
FDP_ACF.1/*	FDP_ACC.1/*	FDP_ACC.1/*
	FMT_MSA.3	FMT_MSA.3
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.3	none	

Table 26. Analysis of fulfillment of SFR dependencies

SFR	Dependecies	Fulfilled by
FIA_UAU.4/*	none	
FIA_UAU.5/*	none	
FIA_UID.1	none	
FMT_MSA.1/*	FDP_ACC.1 or FDP_IFC.1	FDP_ACC.1/*
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.2	FDP_ACC.1 or FDP_IFC.1	FDP_ACC.1/*
	FMT_MSA.1	FMT_MSA.1/*
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3	FMT_MSA.1	FMT_MSA.1/*
	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_SMF.1	none	
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_ITC.1	none	
FPT_ITI.1	none	
FPT_RPL.1	none	
FTP_ITC.1	none	
FTP_TRP.1	none	

Application Note 13

The FAU_GEN.1 dependency on the FPT_STM.1 is not fulfilled, because the TSF relies on the operating system to provide trusted timestamps. The environment objective OE.Trusted_Timestamps is ensuring that the operating system is configured to synchronise the clock to the trusted time source.

7.5 Security Assurance Requirements

7.5.1 Rationale for selecting the SARs

The assurance level for this ST is chosen to be the EAL4 augmented. EAL4 permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices, without the need for highly specialised processes and practices. EAL4 is considered to be the highest level that could be applied to the product without undue expense and complexity. As such, EAL4 is appropriate for the commercial products that require moderate to high security functions. The TOE described in this ST (TOE type QSCD) is just such a product.

The EAL4 is augmented by AVA_VAN.5 (Advanced methodical vulnerability analysis). This is chosen because the TOEs of type QSCD must be highly resistant to the penetration attacks to meet the security objectives of the P.SCD_Confidential, P.Sig_unForgeable, P.SCD_userOnly.

7.5.2 Security assurance components

The assurance components are identified in the table 27.

Table 27. Security Assurance Components used in the ST

Assurance Class	Assurance Components
Security Target (ASE)	ST introduction (ASE_INT.1) Conformance claims (ASE_CCL.1) Security problem definition (ASE_SPD.1) Security objectives (ASE_OBJ.2) Extended components definition (ASE_ECD.1) Derived security requirements (ASE_REQ.2) TOE summary specification (ASE_TSS.1)
Development (ADV)	Security architecture description (ADV_ARC.1) Complete functional specification (ADV_FSP.4) Basic modular design (ADV_TDS.3) Implementation representation of the TSF (ADV_IMP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1) Preparative measures (AGD_PRE.1)
Life-cycle support (ALC)	Production support, acceptance procedures and automation (ALC_CMC.4) Problem tracking CM coverage (ALC_CMS.4) Delivery procedures (ALC_DEL.1) Identification of security measures (ALC_DVS.1) Developer defined life-cycle model (ALC_LCD.1) Well-defined development tools (ALC_TAT.1)
Tests (ATE)	Functional testing (ATE_FUN.1) Analysis of coverage (ATE_COV.2) Testing: basic design (ATE_DPT.1) Independent testing - sample (ATE_IND.2)
Vulnerability assessment (AVA)	Advanced methodical vulnerability analysis (AVA_VAN.5)

7.5.3 SAR dependencies analysis

The assurance level of this ST is EAL4 augmented by AVA_VAN.5 (advanced methodical vulnerability analysis). The component AVA_VAN.5 has the following dependencies:

- ADV_ARC.1 Architectural design with domain separation and non-bypassability
- ADV_FSP.4 Complete functional specification
- ADV_TDS.3 Basic modular design
- ADV_IMP.1 Implementation representation of the TSF

- AGD_OPE.1 Operational user guidance
- AGD_PRE.1 Preparative procedures
- ATE_DPT.1 Testing: basic design

All of these dependencies are met in the EAL4 assurance package.

8 TOE Summary Specification (ASE_TSS)

This section provides the summary information of the Security Functions of the TOE and describes how the TOE satisfies all the SFRs described in the section 7.3 – Security Requirements (ASE_REQ). It is meant as a high-level overview of the TOE. For more detailed information, please refer to the technical architecture documents of the SecureZone [7] and other components of the Smart-ID system [8].

8.1 TOE Security Functions

8.1.1 TOE management and access control

8.1.1.1 SF.Authentication

The TOE authenticates users with the following methods:

1. no personalised identification/authentication – for example, the caller to the monitoring interface are not authenticated by the TOE (only general uptime, performance and health information about the TOE is provided over the monitoring interface). Also, some operations regarding key pairs are not authenticated by the TOE. For example, querying the status of a key pair and destroying a key pair is protected by environment and network mechanisms and not by the TOE itself. For details, please refer to the section 7.2.5, where the Security Function Policy SFP/Anonymous is defined.
2. S.App authentication with the possession-based authentication data. Basically, the TOE is using the user-name and password authentication and the shared cryptographic key D.TEK to authenticate App. The TOE updates the D.OTP for each upcoming key operation and sends the new password to the App. Because the TOE can detect if App is using the wrong D.OTP, this makes the one-time password a possession-based authentication factor.
3. S.Signer authentication with the possession-based and the knowledge-based authentication data. The TOE adds private key-based authentication factor with the proof-of-possession to the S.App authentication. Because Signer has to enter the correct D.PIN to decrypt the local D.clientPart in order to create the D.applicationSignaturePart, this adds the knowledge-based authentication factor.
4. S.Admin authentication with the HSM OCS password.

When a certain (TOE administator configurable) number of incorrect S.Signer authentication attemps have been made, the TOE destroys the key pair.

This SF implements the following SFRs:

1. FIA_AFL.1 – Authentication failure handling
2. FIA_UID.1 – Timing of identification
3. FIA_UAU.1 – Timing of authentication

4. FIA_UAU.3 – Unforgeable authentication
5. FIA_UAU.4/Signer and FIA_UAU.4/App – Single-use authentication mechanisms
6. FIA_UAU.5/Signer and FIA_UAU.5/App – Multiple authentication mechanisms
7. FPT_RPL.1 - Replay detection

8.1.1.2 SF.AccessControl

In general, the TOE divides the users into three main groups:

1. anonymous users,
2. key pair owners (Signers),
3. privileged users (Admins and CA)

Anonymous users are allowed to perform some operations, which do not require authorisation. For example, querying the status of a key pair and destruction of a key pair are not authenticated by the TOE and no special authorisation is required.

The key pair owners are allowed to perform the key pair operations on their own key pair. In case the Signer is authenticated with possession-based and the knowledge-based authentication data, the TOE allows to complete the signature. In case the Signer is only authenticated with possession-based authentication factors (as is the case when the Smart-ID App is performing technical operations on behalf of the Signer and the App doesn't request authorisation and the entry of the D.PIN from the Signer), the TOE only allows to perform technical operations. This kind of access control follows naturally from the implemention of the TSSP protocol, which requires that in order to complete the Signer's D.signature, one needs the D.applicationSignaturePart and without that, it is not mathematically possible to create a signature.

The "owning" of a key pair is determined first by verifying that the claimed D.Signing_Key_Id and presented passwords and used cryptographic key D.TEK correspond to the information in the TOE database. Additionally, the "owning" of a key pair is also determined mathematically, as the presented D.applicationSignaturePart needs to match with the D.serverSignaturePart. In case somebody would claim ownership of some other key pair, the signature verification with the D.clientModulus would fail. This sort of access control feature also follows naturally from the implemention of the TSSP protocol.

Privileged users can perform key pair operations on any key pair. However, the list of operations is limited to only specific methods. Privileged users are not allowed to invoke signature completion at all, since such functions are not included in the API which is dedicated to them.

All those rules are described in more detail within the section 7.2 – Security Requirements (ASE_REQ).

This SF implements the following SFRs:

1. FDP_ACC.1/Signer, FDP_ACC.1/App, FDP_ACC.1/Anonymous, FDP_ACC.1/Admin, FDP_ACC.1/CA – Subset access control
2. FDP_ACF.1/Signer, FDP_ACF.1/App, FDP_ACF.1/Anonymous, FDP_ACF.1/Admin, FDP_ACF.1/CA – Security attribute based access control
3. FMT_MSA.1.1/Signer, FMT_MSA.1/App, FMT_MSA.1/Admin – Management of security attributes
4. FMT_MSA.2 – Secure security attributes
5. FMT_MSA.3 – Static attribute initialisation
6. FMT_MTD.1 – Management of TSF data

7. FMT_SMF.1 – Specification of Management Functions
8. FMT_SMR.1 – Security roles

8.1.1.3 SF.Audit – Security audit generation

The audit records of the important system events are generated by the TOE and saved to its database to be exported to an external system.

This SF implements the FAU_GEN.1 – Security audit generation.

8.1.2 Handling of cryptographic material and algorithms

8.1.2.1 SF.KeyGen – Key generation

The TOE uses a Common Criteria certified HSM to perform most of the key generation operations. In case the HSM doesn't support generation and management of a particular key type, the TOE handles generation of that by itself.

1. D.SVD – The TOE implements the TSSP [5] and generates the compund modulus of the key pair, using modulus multiplication of D.clientModulus and D.serverModulus
2. D.KTK – The TOE uses the HSM to generate the regular RSA key pair. The private key will be encrypted by HSM.
3. D.TEK – The TOE implements Diffie-Hellman key agreement protocols, see 7.3.2.1.3.
4. D.KWK – The TOE uses the HSM to generate the regular AES key. The key will be encrypted by HSM.
5. D.DEK – The TOE uses the RNG provided by the HSM to generate the regular AES key. The key will be wrapped with D.KWK.

This SF implements the following SFRs:

1. FCS_CKM.1/RSA_SVD, FCS_CKM.1/RSA_KTK, FCS_CKM.1/DH_TEK, FCS_CKM.1/AES_KWK, FCS_CKM.1/AES_DEK – Cryptographic key generation

8.1.2.2 SF.CryptoAlgorithms – Using standard cryptographic algorithms

The TOE uses a Common Criteria certified HSM to perform most of the key usage operations. In case the HSM doesn't support operations with a particular key type, the TOE implements those by itself.

1. computation of the signatures – The TOE implements the TSSP [5] and generates the compound signatures (D.signature) from the shares of signature.
2. creation and verification of RSA signatures – The TOE uses the HSM to generate the RSA signature and 3rd party library Bouncy Castle to verify the signatures.
3. encryption/decryption of JWE messages for transmission and database storage – The TOE uses the 3rd party library Nimbus to create and verify the JWE messages. The encryption/decryption operations are delegated to the HSM.

This SF implements the following SFRs:

1. FCS_COP.1/RSA_SCD, FCS_COP.1/RSA_Other, FCS_COP.1/AES, FCS_COP.1/HMAC, FCS_COP.1/SHA-2, FCS_COP.1/SHA-3 – Cryptographic operation

8.1.2.3 SF.KeyZer – Key destruction

The TOE destroys the following cryptographic keys after they are no longer used:

1. D.serverPart
2. D.serverShare
3. D.DEK
4. D.TEK
5. D.KWK
6. D.KTK

The TOE uses platform standard tools to destroy the keys.

This SF implements the FCS_CKM.4 – Cryptographic key destruction.

8.1.3 Protecting communication with external components

8.1.3.1 SF.TrustedPath – Trusted path with the user

The TOE uses JWE messages for communicating with the Smart-ID App TSE. JWE messages are encrypted with the D.TEK and they are integrity protected.

This SF implements the FTP_TRP.1 – Trusted path.

8.1.3.2 SF.SecureChannel – Secure channel with external components

The TOE uses vendor-specific proprietary communication channel when connecting with HSM or database, such as nCipher impath and PostgreSQL connections. Those methods provide the cryptographic checksum validation of the integrity for the transmitted data. When the TOE detects the modifications and integrity errors with the transmitted data, it aborts the operation.

This SF implements the following SFRs:

1. FTP_ITC.1 – Inter-TSF trusted channel
2. FPT_ITI.1 – Inter-TSF detection of modification
3. FPT_ITC.1 – Inter-TSF confidentiality during transmission

8.2 TOE Summary Specification Rationale

The table 28 shows the mapping between SFRs and TOE Security Functions to provide a quick overview.

Table 28. Mapping between SFRs and TSF

SFR	SF
FAU_GEN.1	SF.Audit
FCS_CKM.1/*	SF.KeyGen
FCS_CKM.4	SF.KeyZer
FCS_COP.1/*	SF.CryptoAlgorithms
FDP_ACC.1/* FDP_ACF.1/*	SF.AccessControl

Table 28. Mapping between SFRs and TSF

SFR	SF
FIA_AFL.1 FIA_UID.1 FIA_UAU.1 FIA_UAU.3 FIA_UAU.4/* FIA_UAU.5/*	SF.Authentication
FIA_MSA.1/* FMT_MSA.2 FMT_MSA.3 FMT_MTD.1 FMT_SMF.1 FMT_SMR.1	SF.AccessControl
FPT_RPL.1	SF.Authentication
FPT_ITC.1 FPT_ITI.1 FTP_ITC.1	SF.SecureChannel
FTP_TRP.1	SF.TrustedPath