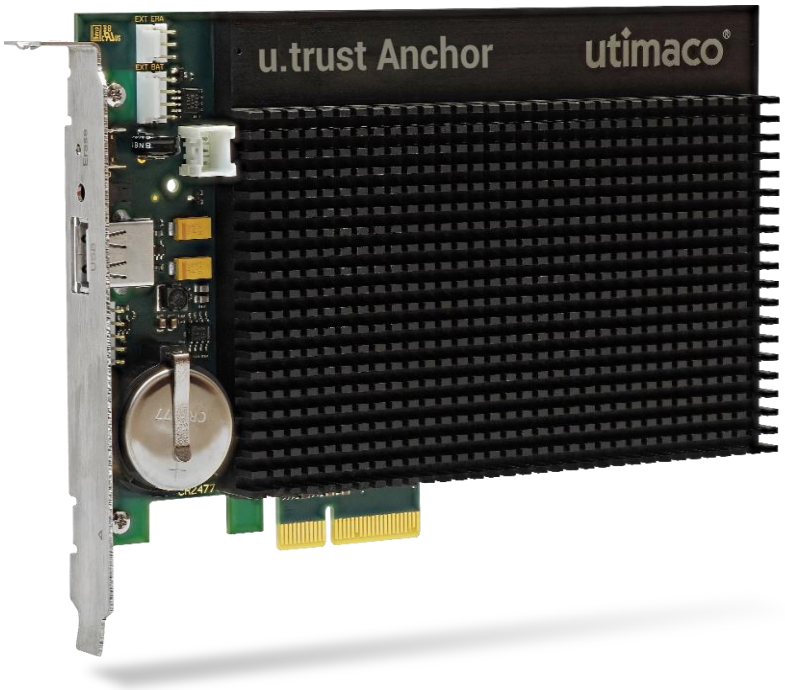


u.trust Anchor

Security Target Lite for u.trust Anchor

utimaco®

Imprint

Copyright 2022	Utimaco IS GmbH Germanusstr. 4 52080 Aachen Germany
Phone	AMERICAS: +1-844-UTIMACO (+1 844-884-6226) EMEA: +49 800-627-3081 APAC: +81 800-919-1301
Internet	https://hsm.utimaco.com
e-mail	hsm@utimaco.com
Document Version	1.0.1
Date	22nd June 2022
Status	Released

Table of Contents

1 Introduction

1.1 Change History

Version	Date	Description
1.0.1	22 June 2022	First release of Security Target Lite, based on full Security Target with same version number

1.2 Document Introduction

This Security Target (ST) was developed based on the Protection Profile (PP) EN 419 221-5: 2018 Protection Profiles for TSP Cryptographic Modules - Part 5: Cryptographic Module for Trust Services, version v1.0 [PP_CMTS], by applying some adaptions to the PP and therefore not claiming strict conformance to it.

The following subchapters provide some information for the further understanding of this document and introduce the reader to some used conventions.

1.2.1 Acknowledgement

The author would like to acknowledge the significant contributions of the Protection Profile [PP_CMTS].

1.2.2 Notations

The notation, formatting, and conventions used in this ST are consistent with those used in the Common Criteria, Version 3.1, Revision 5, April 2017 [CC1], [CC2], [CC3].

The Common Criteria allow several operations to be performed on security requirements: refinement, selection, assignment and iteration are defined in Section C.2 of [CC1]. For more details on the notations see chapter 7.1 “Typographical Conventions”.

1.2.3 Abbreviations

Assumptions, threats, organisational security policies and security objectives (for TOE and environment) are assigned with a unique label for easy reference as follows:

R.<xxx> Assets

S.<xxx> Subjects

T.<xxx> Threats

P.<xxx> Organisational security policies

A.<xxx>	Assumptions about the TOE security environment
OT.<xxx>	Security objectives for the TOE
OE.<xxx>	Security objectives for the operating environment

1.2.4 References

References in this document are specified with the help of brackets (e.g.: [<Reference>]). A list of all referenced documents can be found in chapter 10.2 “References”.

1.2.5 Terminology

A complete list of used terms and abbreviations can be found in chapter 10.1 “Glossary and Acronyms”. Thereby Common Criteria and IT technology terms relevant for this ST are described. Most of the definitions are taken from the PP [PP_CMTS] as well as from the Common Criteria.

2 Security Target Introduction

2.1 ST and TOE Reference

Title:	u.trust Anchor - Security Target Lite for u.trust Anchor
ST Version:	1.0.1
ST Date:	22nd June 2022
Author:	Utimaco IS GmbH
Developer:	Utimaco IS GmbH
Product:	u.trust Anchor
TOE-name long:	u.trust Anchor
TOE-name short:	u.trust Anchor
TOE-versions:	u.trust Anchor 4.49.0
Product Type:	Cryptographic module
Certification Authority:	TÜV Rheinland Nederland B.V, Singapore CSA
CC Version:	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017 [CC1], [CC2], [CC3]

2.2 Related Documents

All related documents can be found in chapter 10.2 “References”.

2.3 Organisation

The main chapters of this ST are Security Target Introduction with the description of the TOE (Target of Evaluation), Conformance claims, Security problem definition, Security objectives, Extended components definition, Security requirements and TOE summary specification as well as annexes. This document is structured according to the Security Target requirements of [CC1].

- **Chapter 2:** The TOE description provides general information about the TOE, its generic structure and boundaries.
- **Chapter 3:** The ST conformance claims section states conformance to Protection Profiles.
- **Chapter 4**: The security problem definition describes security aspects of the environment in which the TOE is intended to be used and the manner in which it is intended to be employed. The security problem definition includes threats relevant to secure TOE operation (section 4.3), organisational security policies (section 4.4), which must be complied by the TOE, and assumptions regarding the TOE's intended usage and environment of use (section 4.5).

- **Chapter 5:** The statement of security objectives defines the security objectives for the TOE (section 5.1) and for its environment (section 5.2). The rationale (section 8.1) presents evidence that the security objectives satisfy the threats and policies.
- **Chapter 6:** This chapter defines the extended components.
- **Chapter 7:** The security requirements are subdivided into TOE Security Functional Requirements (section 7.2) and Security Assurance Requirements (section7.3).
- **Chapter 8:** The rationale (section 8.2) explains how the set of requirements is complete relative to the security objectives.
- **Chapter 9:** The TOE summary specification provides a description of the TOE security functionality in narrative form.

The annex in **Chapter 10 Annex** offers a glossary and acronyms as well as relevant references.

2.4 TOE Overview

The scope of this Security Target is to describe the security functionality of the TOE, which is a general purpose Hardware Security Module (HSM) based on the Utimaco u.trust Anchor platform (short: u.trust Anchor), in terms of Common Criteria and to define security functional and assurance requirements for this system.

In general terms, the TOE u.trust Anchor is a new generation version of a traditional hardware security module (HSM), comprising all of the traditional hardware security features normally applicable to such a device - but additionally introducing the concept of **containerized HSMs (cHSMs)** within the protected boundary of the hardware HSM (the TOE).

As any traditional HSM, u.trust Anchor is a general purpose HSM whose primary purpose is to provide secure cryptographic services such as signing and verification of data, encryption or decryption, MAC calculation, key derivation and key agreement, hashing, on-board random number generation and secure key generation, internal as well as external protected key storage and further key management functions in a tamper-protected environment. It can be used with all cryptographic standard APIs like PKCS#11, JCE, OpenSSL, CSP/CNG and EKM. Furthermore, the TOE provides a secure software update mechanism.

This new generation of HSM is developed to improve scalability, both in single, and in multi-tenanted environments (such as data centers or cloud service providers), and to provide any service providers with a highly elastic HSM architecture, one that can rapidly scale on demand, but also an architecture that enables the service providers to deliver HSM as a Service (HSMaaS) in multiple use cases. The TOE u.trust Anchor can run up to 31 containerized HSMs in parallel, where each cHSM can be used independently from any other cHSM on the same hardware device. In particular, each cHSM can be used with all cryptographic standard APIs like PKCS#11, JCE, OpenSSL, CSP/CNG and EKM, individually, or in cluster mode for optimized performance and high availability.

The goal of the u.trust Anchor platform is to virtualize and allocate shared resources such that each cHSM has visibility only of the resource set (data, keys, configuration) that appears to be entirely its own.

Each single cHSM provides secure cryptographic services such as signing and verification of data (like ECDSA, EdDSA and RSA), encryption or decryption (for various cryptographic algorithms like AES and RSA, including mechanisms for authenticated encryption like AES GCM or CCM), MAC calculation, key derivation and key agreement, hashing, on-board

random number generation and secure key generation, internal as well as external protected key storage and further key management functions in a tamper-protected environment.

The u.trust Anchor platform consists of the following subsystems:

- The u.trust Anchor hardware
- The u.trust Anchor platform firmware COSMOS: including boot loader, Linux kernel, container management firmware and Global Administration service firmware (GLAD)
- cHSM (containerized HSM) firmware which is provided by COSMOS in cHSM firmware templates which can be loaded into containers

2.4.1 TOE Hardware

The **u.trust Anchor** hardware is a physically protected cryptographic module provided in the form of a PCI Express (PCIe) plug-in card, as shown in the following picture:

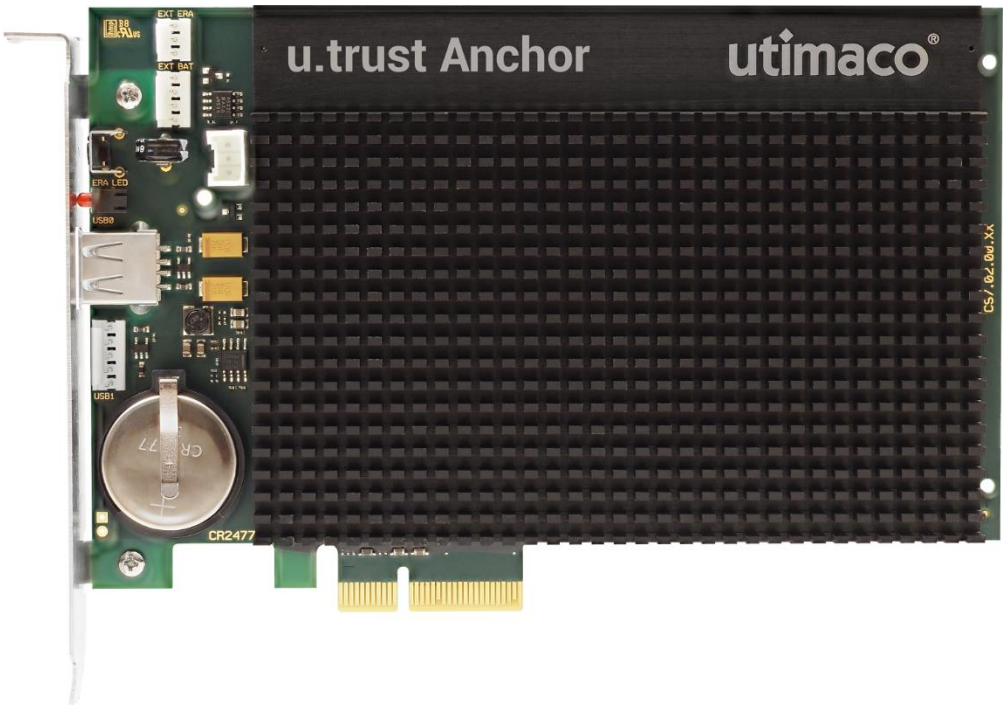


Figure 1: **u.trust Anchor**

Optionally and as a delivery variant, the PCIe plug-in card can be integrated into an Utimaco u.trust Anchor LAN, a 19-inch network appliance with display, control buttons and USB interfaces on the front panel, see Figure 2 below. The appliance contains an industry-quality PC motherboard, backplane with PCIe bus interface, flash disk (as mass storage), two redundant power supplies and a backup battery. The PCIe security module is plugged into the PCIe bus interface of the backplane. The u.trust Anchor LAN may be connected to an Ethernet network via a Gigabit network interface on the backside.

Figure 2: u.trust Anchor LAN

The main components of the TOE hardware include a multi-core ARM processor, 2 GBs of DDR4 RAM (of which, a few MBs are set aside for secure storage of keys[1]), a non-volatile RAM (NV-RAM) and a flash memory as secondary storage, a cryptographic accelerator with support for RSA and ECC operations, a soft cryptographic accelerator IP block in the FPGA used for acceleration of certain ECC curve operations, and a noise source for a highest-quality physical random number generator (RNG). Secret keys and sensitive data will never be stored unencrypted on NV-RAM and FLASH devices.

All hardware components of the cryptographic module, including the Central Processing Unit (CPU), all memory chips, Real Time Clock (RTC), and hardware noise generator for random number generation, are located on a printed circuit board (PCI express board). These hardware components are completely covered with potting material (epoxy resin) and a heat sink. This hard, opaque enclosure protects the sensitive **u.trust Anchor** hardware components from physical attacks.

2.4.2 TOE Firmware

The **u.trust Anchor** platform firmware COSMOS (see Figure 3) consists of:

- A bootloader
- A bespoke Linux kernel compiled with the minimum features necessary to allow the platform to function, and including security components such as mandatory access control, resource control and other sandboxing techniques.
- Custom drivers and services as part of the platform firmware image, to enable communication, for instance, with the random number generator and the cryptographic accelerators;
- The Global Administrator instance GLAD and the container management middleware
- Individual cHSM firmware templates. Each cHSM template can be loaded into one or more containers as provided by COSMOS. Each container provides the crypto functionality of an HSM. For the TOE version of u.trust Anchor, only one cHSM template is used.

[1] Keys stored in Secure RAM are stored unencrypted but are erased if certain extraordinary physical circumstances are detected by internal sensors.

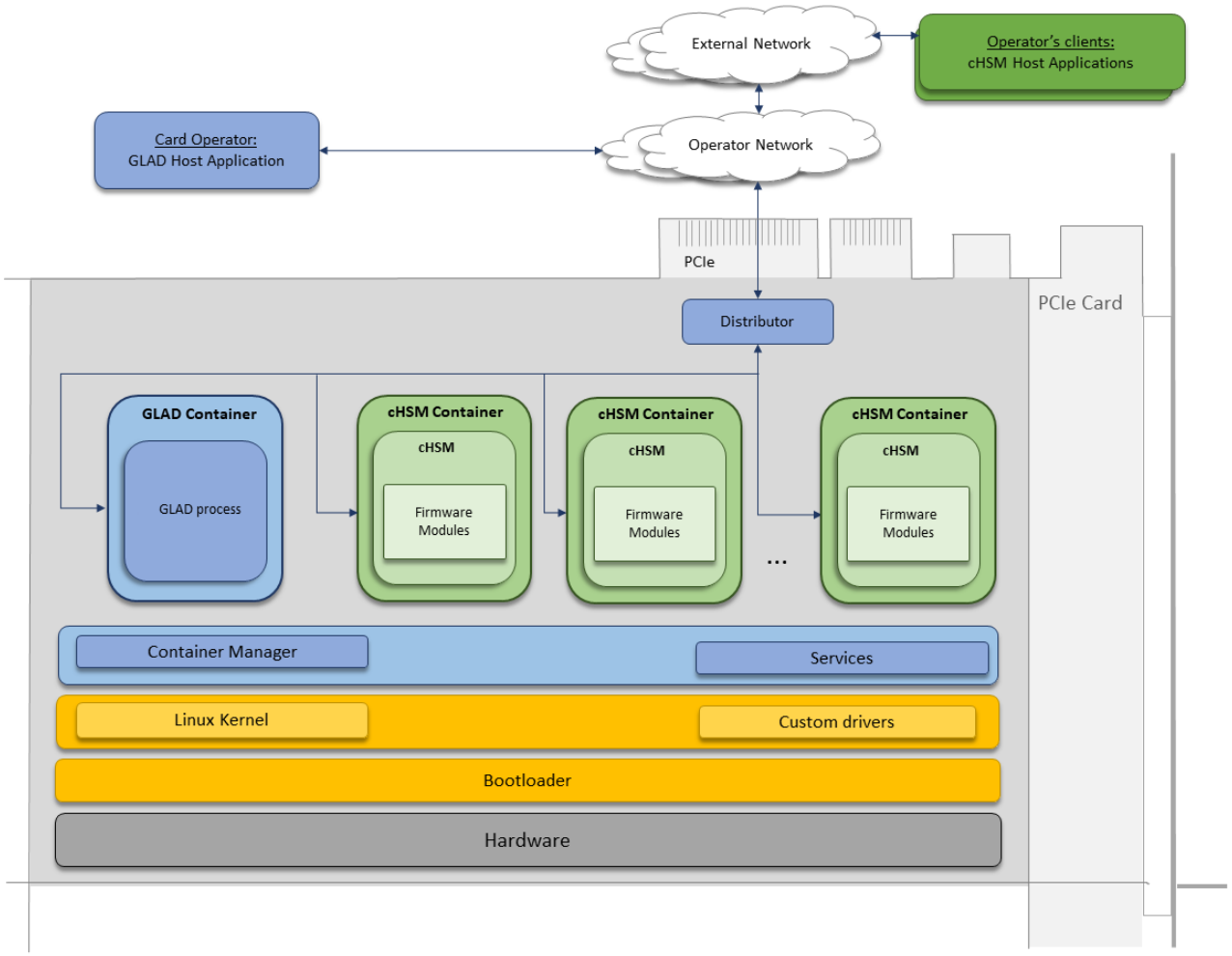


Figure 3: **u.trust Anchor** Firmware Overview

The **u.trust Anchor** platform firmware constitutes a limited operational environment. Loaded cHSM firmware cannot be modified and must pass a firmware integrity test on every cHSM start-up.

The **u.trust Anchor** platform firmware is responsible for the segregation of processes running on different containers: A process running in a container cannot detect, access or modify data belonging to a process running in a different container, or the base operating system.

The containers are isolated from each other and the base operating system by a multi-layered set of technologies (comprising namespaces, mandatory access control and resource controls), allowing multiple cHSM instances to run on a single system without interference.

Management of the containers, including creation, deletion, start, stop, backup and restore of the containers is part of the Global Administrator role. The Global Administrator role and its authentication mechanisms are completely separate from the cHSM roles and authentication mechanisms. The Global Administrator has - by design of the operator roles - no mechanism to access unencrypted data from individual cHSMs.

The cHSM firmware is a collection of firmware components (called modules) instantiated from a cHSM template that provides the required cryptographic functionality like AES, RSA, ECC,

and hashing as well as supporting functionality like key storage and communication with external devices/host applications.

cHSMs can be used in almost all proprietary environments in which cryptographic services and highest security are required, such as archiving systems and payment systems. They can serve as a signature server, time stamp server, and generator for PINs, cryptographic keys, or random numbers.

u.trust Anchor offers hardware-based random bit generation (entropy) as well as Approved deterministic random bit generators (DRBG) for GLAD and for cHSMs. The hardware based random bit generation is used to seed and re-seed these DRBGs.

2.4.3 TOE Interfaces

Being a PCIe plug-in card, for the communication with a host, the TOE offers a PCIe interface and a serial log interface.

A Secure Messaging concept uses message encryption and MAC authentication to protect communication to and from the TOE – from any client application towards the Global Administrator command interface as well as for the communication with any individual cHSM and its command interfaces.

Together with Utimaco's appropriate host application software the cHSMs also provide cryptographic standard interfaces like PKCS#11, JCE, OpenSSL, CSP/CNG and EKM.

2.5 TOE Description

This chapter contains the following sections:

- TOE configuration and TOE environment (section 2.5.1)
- Physical scope (section 2.5.2)
- Logical scope (section 2.5.3)
- Deliverables (section 2.5.4)

2.5.1 TOE Configuration and TOE Environment

The TOE is provided in different configuration variants[2], reflecting different capability and flexibility in usage of containerized HSMs (cHSMs):

- u.trust Anchor CSAR Premium (with a flexible number of up to 31 cHSMs)
- u.trust Anchor CSAR Plus (with a flexible number of up to 16 cHSMs)
- u.trust Anchor CSAR Standard (with a flexible number of up to 8 cHSMs)
- u.trust Anchor Se40k (with up to 12 cHSMs, to be used in one cluster)
- u.trust Anchor Se15k (with up to 4 cHSMs, to be used in one cluster)

In all variants the TOE should be hosted by a card operator whose operational environment (see Figure 4) is assumed trustworthy and secure. The operator may provide remote access

[2] All configuration variants are based on the same TOE hardware and software, the configuration is fixed upon delivery.

to cHSMs to its clients via a network. The external environment (as depicted in Figure 4) is not under the control of the operator.

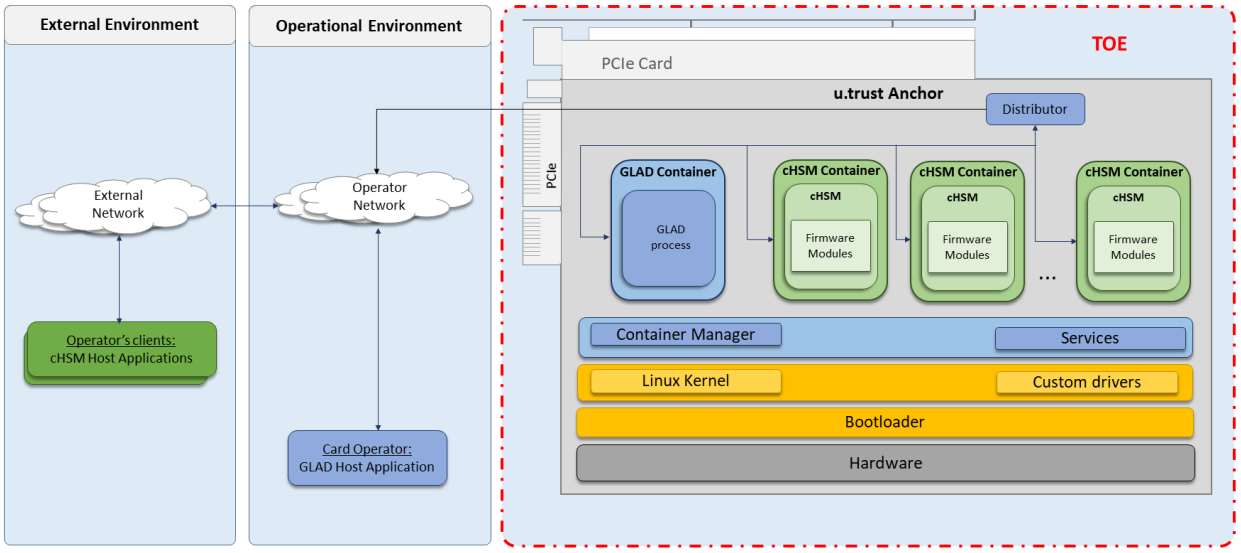


Figure 4: Target of Evaluation (TOE) Boundary and Environment

2.5.2 Physical Scope

The TOE boundary is defined as the outer perimeter of the heat sink on the top side and the epoxy surface on the bottom side of the module.

Figure 5 and Figure 6 below show views of the cryptographic boundary from the side and the top, and from the bottom. The red dashed line indicates the cryptographic boundary.

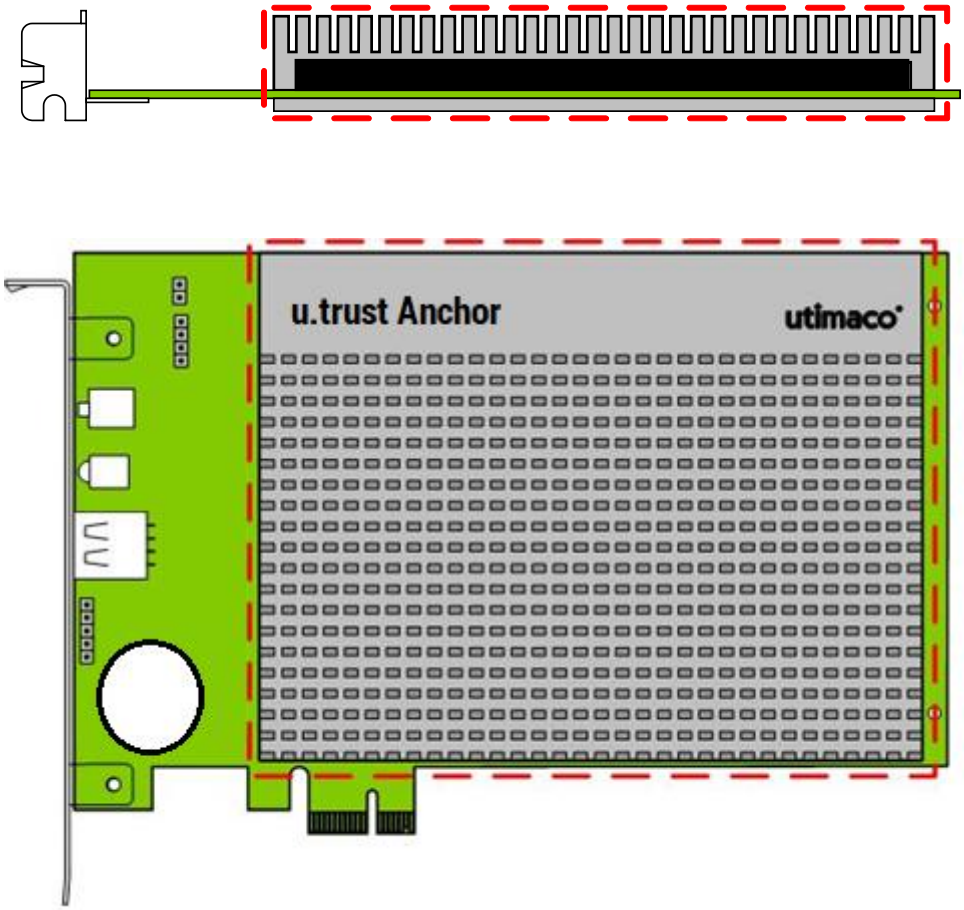


Figure 5: **u.trust Anchor** – side view and top view

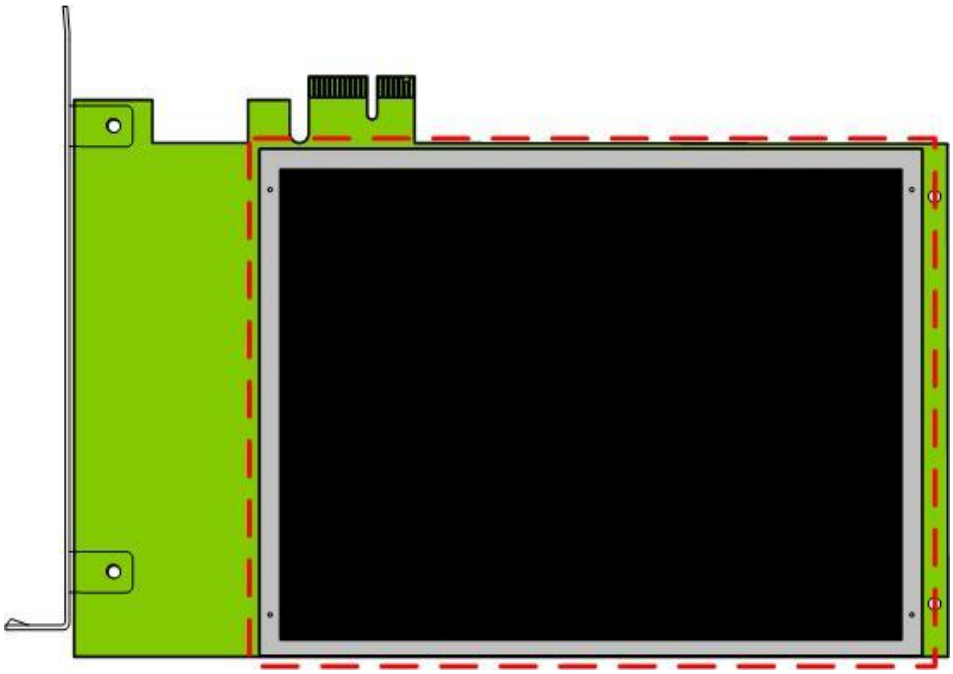


Figure 6: **u.trust Anchor** – bottom view

2.5.3 Logical Scope

The u.trust Anchor implements the following cryptographic algorithms:

- AES in various modes for encryption, decryption, CMAC and GMAC calculation, key (un)wrapping and Secure Messaging
- TDES in various modes for encryption and decryption
- ECDSA and EdDSA with key size >= 224 bit on dedicated elliptic curves for signature generation and signature verification
- RSA with key size >= 2048 bit and <= 16,384 bit for signature generation and signature verification and key (un)wrapping
- SHA-2, SHA-3 and HMAC for hashing, pseudo random function and MAC calculation

Furthermore the u.trust Anchor implements functionality for key establishment:

- AES key generation
- TDES key generation
- Generation of generic secret keys, e. g. for HMAC algorithm
- Elliptic curve cryptography (ECC) key generation, e. g. for ECDSA, EdDSA and ECDH
- RSA key generation
- DSA domain parameter generation and DH key generation
- Diffie-Hellman and EC Diffie-Hellman Key Agreement
- Key Derivation

For random number generation and generation of all cryptographic keys, challenges and nonces, the u.trust Anchor implements a hybrid deterministic random number generator that relies on an implemented hardware random noise generator and fulfills the requirements of [AIS 20/31].

For operation purposes, the u.trust Anchor supports the following cryptographic services:

- Functions for Initialisation:
  - Generation of RSA OAEP key establishment keys for secure import of Operator Base Secret
  - Import of wrapped Operator Base Secret
  - Generation of cHSMs with various cHSM-individual assigned system keys and certificates
  - Generation and export of user controlled Master Backup Keys
  - Import of user controlled Master Backup Keys

- Functions for Key Management (for keys in internal as well as external key store):

  - Key generation (AES keys, TDES keys, generic secret keys, ECC key pairs, RSA and DH key pairs)
  - Encrypted import and export of private and secret keys (AES, RSA)
  - Key agreement (DH, ECDH)
  - Backup and restore of keys
  - Key deletion

- Cryptographic Functions:
  - Signature generation and verification (ECDSA, EdDSA, RSA)
  - Encryption and decryption (AES, TDES, RSA)
  - MAC calculation and verification (AES GMAC, AES CMAC, HMAC)
  - Hashing (SHA-2, SHA-3)
  - Generation of random bytes

For the operation purpose, the u.trust Anchor supports the following administrative services:

- User administration (creation, deletion, change of reference authentication data (RAD))
- System time setting/display
- Export and deletion of audit data
- cHSM management (e.g. create, start, stop, delete cHSM)
- Backup ('snapshot') and restore of cHSMs

To support the security of the above mentioned features of the TOE, the u.trust Anchor provides appropriate countermeasures for resistance especially against the following attacks:

- Cloning of the product
- Unauthorised disclosure of confidential data (during generation, storage and processing)
- Unauthorised manipulation of data (during generation, storage and processing)
- Unauthorised usage of private and secret keys
- Derivation of information on the private key from publicly available data like the related public part of the generated key pair, by implementing sufficiently strong and approved cryptographic mechanisms
- Physical and chemical attacks

Furthermore, the TOE provides a secure software update mechanism. Software revisions shall be granted security certification before their installation in the TOE.

2.5.4 Deliverables

The following list contains an overview of all deliverables associated to the TOE:

- Hardware, the version number is given below
- Software, pre-installed on the hardware, version numbers see below
- Guidance documents for the Global Administrator and for users of a cHSM of the u.trust Anchor, delivered as electronic files.

The table of TOE deliverables can therefore be described as follows:

TOE deliverable	Type/Form, Name	Exact reference	Delivery
Hardware	*The TOE hardware is provided in form of a PCIe plug-in card.* Hardware P/N CSAR-7.3.0.3-PCIe-CC (PCIe security module)	Version 7.03.0.3	per courier
Software	*All TOE software (apart from the sensory controller) is provided as binary image in form of a RAUC bundle (*.raucb format).*		
	Operational Image (glados-utrust-anchor-1.22.5.raucb)	Version 1.22.5	pre-installed on TOE hardware (as primary and secondary-/backup Operational Image), and additionally per web download via Utimaco Portal
	Recovery Image (glados-recovery-1.22.5.raucb)	Version 1.22.5	pre-installed on TOE hardware
	Sensory Controller	Version 3.02.0.8	pre-installed on TOE hardware
Guidance documents	*All TOE guidance documentation is provided in form of pdf documents.* *Operating Manual in two variants (delivery variant PCIe/LAN):*		per web download via Utimaco Portal
	u.trust Anchor PCIe CC - Operating Manual	2021-0084, version 1.0.4, date 2022-06-09	
	u.trust Anchor LAN V5 CC - Operating Manual	2021-0069, version 1.0.7, date 2022-06-09	
	User Manuals:		
	u.trust Anchor CC - Administration Manual (*Administration Manual for Global Administration*)	2021-0078, version 1.0.7, date 2022-06-21	
	u.trust Anchor CC - Containerized Hardware Security Module (cHSM) - Administration Manual (*Administration Manual for cHSM*)	2021-0077, version 1.0.9, date 2022-06-07	

TOE deliverable	Type/Form, Name	Exact reference	Delivery
	u.trust Anchor CC - Containerized Hardware Security Module (cHSM) - User Manual (*User Manual for cHSM*) u.trust Anchor CC - Global Admin Management Tool (gladm) - Reference Manual u.trust Anchor CC - csadm Manual	2021-0076, version 1.1.3, date 2022-05-12 2021-0074, version 1.1.3, date 2022-06-13 2021-0075, version 1.0.3, date 2022-06-07	

Table 1: TOE deliverables

2.6 Required Non-TOE Hardware/Software/Firmware

The following hardware and software which do not belong to the TOE is required for the operating environment and is always delivered per courier together with the TOE:

Additional deliverables	Type/Form
PIN pad (smartcard reader with keypad)	HW/SW Utimaco cyberJack one
10 smartcards (for administrative purposes)	HW/SW Java Card with NXP Chip and JCOP operating system

The TOE is delivered in two different variants:

- u.trust Anchor PCIe (PCIe plug-in card)
- u.trust Anchor LAN (network-attached appliance)

Depending on the delivery variant, apart from the TOE itself, the following non-TOE hardware, software and further data is delivered with the TOE (non-TOE-deliverables, not necessarily required but help to run the TOE):

Additional Deliverable:	CSLAN	Cable	Product bundle
Delivered variant:			
u.trust Anchor PCIe	-	-	1
u.trust Anchor LAN	1	2	1

Herein denotes:

- **CSLAN**: CryptoServer LAN (19-inch network appliance with two redundant power supplies) (non-TOE hardware, delivered together with TOE hardware per courier)
- **Cable**: power supply cable (non-TOE hardware, delivered together with TOE hardware per courier)
- **Product bundle**: The product bundle containing the following firmware, software and data (available per web download via Utimaco Portal):
  - The u.trust Anchor driver (for Linux) (non-TOE software)
  - Various cryptographic APIs (non-TOE software, to be used on host)
  - The documentation of the cryptographic APIs in PDF format (non-TOE documentation)
  - The installation files of various administration tools and key management tools (non-TOE software, to be used on host)
  - Further guidance documents, e. g. for all administration tools (non-TOE documentation)
  - The keyfile with the authentication key for the default Global Administrator (initial authentication key) of the u.trust Anchor (non-TOE data)

3 Conformance Claims

3.1 CC Conformance Claim

This ST claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017 [CC1]
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017 [CC2]
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2017-04-003, Version 3.1, Revision 5, April 2017 [CC3]

as follows

- **CC Part 2 extended**
- **CC Part 3 conformant**

The

- Common Criteria for Information Technology Security Evaluation, Evaluation methodology; CCMB-2017-04-004, Version 3.1, Revision 5, April 2017 [CEM]

has to be taken into account.

3.2 PP Claim

The ST has no claim for conformance to any Protection Profile but it is inspired by the Protection Profile *EN 419 221-5:2018 Protection Profiles for TSP Cryptographic Modules – Part 5: Cryptographic Module for Trust Services* [PP_CMTS].

3.3 Package Claim

The assurance level for this Security Target is **EAL4 augmented with AVA_VAN.5 and ALC_FLR.3** (EAL4+ conformant).

3.4 Conformance Rationale

The ST has no claim for conformance to any Protection Profile but it is inspired by the Protection Profile [PP_CMTS].

4 Security Problem Definition

This chapter contains the following sections:

- Assets (section 4.1)
- Subjects (section 4.2)
- Threats (section 4.3)
- Organisational Security Policy (section 4.4)
- Assumptions (section 4.5)

4.1 Assets

The assets that need to be protected by the TOE are identified below.

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

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

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

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

4.2 Subjects

The types of subjects identified in this ST are:

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

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

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

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

4.3 Threats

The following threats are defined for the TOE. The attacker (i. e. the 'threat agent') described in each of the threats is a subject who is not authorised for the relevant action, but who may present themselves as either a completely unknown user, or as one of the subjects in section 4.2 (but in this case the attacker will not have access to the authentication data for the subject).

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

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

T.KeyMod **Unauthorised modification of a key**
An attacker makes an unauthorised modification to a secret or public key (R.SecretKey or R.PubKey) while it is stored in, or under the control of, the TOE, including external storage, export and backups. This includes replacement of a key as well as making changes to the value of a key, or changing its critical attributes such as, usage constraints or identifier (changing the identifier to the identifier used for another key would allow unauthorised substitution of the original key with a key known to the attacker). The threat therefore includes the case where an attacker is able to break the binding between a key and its critical attributes[3].

T.KeyMisuse **Misuse of a key**
An attacker uses the TOE to make unauthorised use of a secret key (R.SecretKey) that is managed by the TOE (including the unauthorised use of a secret key for a cryptographic function that is not permitted for that key[4]), without necessarily obtaining access to the value of the key.

[3] See OT.KeyIntegrity in section 4.1 for further discussion of critical attributes of a key.

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

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

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

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

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

4.4 Organisational Security Policies

P.Algorithms **Use of approved cryptographic algorithms**
The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognised authorities such as e. g. SOG-IS or NIST.

P.KeyControl **Support for control of keys**
The life cycle of the TOE and any secret keys that it manages (where such keys are associated with specific entities, such as the signature creation data associated with one or more signatories), shall be implemented in such a way that the secret keys can be reliably protected against use by users or entities that are not authorized to use the keys, and in such a way that the use of the secret keys by the TOE can be confined to a set of authorised cryptographic functions.

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

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

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

4.5 Assumptions

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

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

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

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

A.AuditSupport **Audit data review**
The audit trail generated by the TOE will be collected, maintained and reviewed by a System Auditor according to a defined audit procedure for the specific system.

Application Note 2 (from [PP_CMTS])
As noted for P.Audit in [PP_CMTS] section 3.4[5], the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.

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

[5] See this document chapter 4.4.

5 Security Objectives

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

5.1 Security Objectives for the TOE

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

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

OT.Algorithms Use of approved cryptographic algorithms
The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognised authorities such as e. g. SOG-IS or NIST. This ensures that the algorithms used do not enable publicly known data to be used to derive secret keys.

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

OT.Auth Authorisation for use of TOE functions and data
The TOE carries out an authentication/authorisation check on all subjects before allowing them to use the TOE. In particular, the TOE always requires authentication/authorisation before using a secret key.

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

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

Application Note 3 (from [PP_CMTS])
Protection of secret keys (as a specific type of sensitive data) is also subject to additional protection specified in other TOE objectives. Any requirements for secure storage and control of access to other types of client application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport. For example, if a client application uses the TOE to perform cryptographic functions on data that represent a passphrase value and the passphrase value is to be stored on the TOE, then the client application would need to use an appropriate encryption function before storing the data on the TOE.

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

Application Note 4 (from [PP_CMTS])
Any requirements for integrity protection of client application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport.

OT.ImportExport Secure import and export of keys
The TOE allows import and export of secret keys only by using a secure method that protects the confidentiality and integrity of the data during transmission – in particular, secret keys must be exported only in encrypted form (it is not sufficient to rely on properties of a secure channel to provide the protection: the key itself must be encrypted). The TOE also allows individual secret keys under its control to be identified as non-exportable, in which case any attempt to export them will be rejected automatically. Public keys may be imported and exported in a manner that protects the integrity of the data during transmission.

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

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

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

OT.FailureDetect Detection of TOE hardware or software failures
The TOE detects faults that would cause some other security property to be weakened or to fail, including:

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

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

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

5.2 Security Objectives for the Operational Environment

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

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

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

- Protection against loss or theft of the TOE or any of its externally stored assets.
- Inspections to deter and detect tampering (including attempts to access side-channels, or to access connections between physically separate parts of the TOE, or parts of the hardware appliance).
- Protection against the possibility of attacks based on emanations from the TOE (e.g. electromagnetic emanations) according to risks assessed for the operating environment.
- Protection against unauthorised software and configuration changes on the TOE and the hardware appliance.

- Protection to an equivalent level of all instances of the TOE holding the same assets (e.g. where a key is present as a backup in more than one instance of the TOE).

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

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

OE.AuditSupport Audit data review
The audit trail generated by the TOE will be collected, maintained and reviewed by a System Auditor according to a defined audit procedure for the specific system.

Application Note 5 (from [PP_CMTS])
As noted for P.Audit in [PP_CMTS] section 3.4[6], the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.

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

[6] See this document chapter 4.4.

6 Extended Components Definition

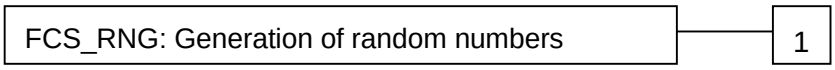
6.1 Generation of Random Numbers (FCS_RNG)

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

Family behaviour:

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

Component levelling:



Management:FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1	**Generation of random numbers**

Hierarchical to: No other components.

Dependencies: No dependencies.

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

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

Application Note 6 (from [PP_CMTS])

A physical random number generator (RNG) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (key strokes, mouse movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs where a hybrid physical RNG produces at least the amount of entropy the RNG output may contain and the internal state of a hybrid deterministic RNG output contains fresh entropy but less than the output of RNG may contain.

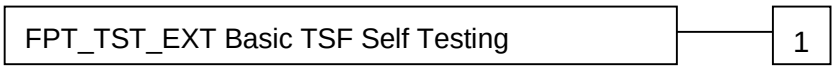
6.2 Basic TSF Self Testing (FPT_TST_EXT.1)

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

Family behaviour:

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

Component levelling:



Management:FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

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

- Indication that TSF self-test was completed.

FPT_TST_EXT.1	**Basic TSF Self Testing**

Hierarchical to: No other components.

Dependencies: No dependencies.

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

7 Security Requirements

This chapter gives the security functional requirements (SFR) and the security assurance requirements (SAR) for the TOE and the environment.
Security functional requirements components given in section 7.2 are drawn from Common Criteria part 2 [CC2]. Some security functional requirements represent extensions to [CC2], with a reasoning given in section 6. Operations for assignment, selection and refinement have been made.

The TOE security assurance requirements statements given in section 7.3 “Security Assurance Requirements” are drawn from the security assurance components from Common Criteria part 3 [CC3].

7.1 Typographical Conventions

The following conventions have been used in this Security Target in the definitions of the SFRs and SARs, in line with the conventions used in the Protection Profile [PP_CMTS]:

- **Refinements:** Refinements are denoted in one of two ways, depending on whether they add detail to an SFR or SAR (‘explanatory refinements’) or update the text of an SFR or SAR element (‘element refinements’). Explanatory refinements follow the SFR/SAR that they update and are marked by the word “**Refinement**” in **bold** followed by text describing the refinement. Element refinements are indicated by **bold** text within an SFR/SAR element, with the original text indicated in a footnote.
- **Selections and Assignments:** Selections and assignments made in the ST are *italicised*, and the original text is indicated in a footnote. If a selection or assignment was already completed in the Protection Profile [PP_CMTS], the PP text is shown in *non-underlined italic letters*. If a selection or assignment is completed by the ST author the text is shown in *underlined italic letters* or, in some cases for better readability, in *non-underlined italic letters*.
- **Iteration:** The iteration operation is used when a component is repeated with varying operations. Iterations within [PP_CMTS] or this ST are denoted by showing a slash “/” and an iteration indicator after the CC component identifier.

If an Application Note e. g. to an SFR was already added by the Protection Profile, this is denoted by “**Application Note <nn> (from [PP_CMTS])**” (if it is adopted by the ST without changes from the Protection Profile), or “**Application Note <nn> (inspired by [PP_CMTS])**” (if it is copied by the ST writer with some adaptions in the specific context of this ST, e. g. by shortening). If an additional Application Note was added by the Security Target writer, this is denoted by “**Application Note <nn>**”. The numbering <nn> of the Application Note is consecutive in the ST (and not identical as given in the Protection Profile [PP_CMTS]).

7.2 Security Functional Requirements

The following table summarises all TOE security functional requirements to meet the security objectives.

No.	SFR	Dependency
	FCS	**Cryptographic Support**
1.	FCS_CKM.1/AES	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
2.	FCS_CKM.1/TDES	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
3.	FCS_CKM.1/GenSecret	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
4.	FCS_CKM.1/RSA	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
5.	FCS_CKM.1/ECC	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
6.	FCS_CKM.1/DH	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
7.	FCS_CKM.2/KeyExport	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
8.	FCS_CKM.2/KeyImport	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
9.	FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
10.	FCS_COP.1/TDES_Crypt	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
11.	FCS_COP.1/AES_Crypt	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

No.	SFR	Dependency
12.	FCS_COP.1/AES_MAC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
13.	FCS_COP.1/RSA_Signature	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
14.	FCS_COP.1/RSA_Crypt	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
15.	FCS_COP.1/ECDSA	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
16.	FCS_COP.1/EdDSA	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
17.	FCS_COP.1/HMAC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
18.	FCS_COP.1/Hash	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
19.	FCS_COP.1/DH	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
20.	FCS_COP.1/ECDH	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
21.	FCS_COP.1/KeyDerivation	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

No.	SFR	Dependency
22.	FCS_RNG.1/PTG.2	No dependencies.
23.	FCS_RNG.1/DRG.4	No dependencies.
	FIA	**Identification and Authentication**
24.	FIA_UID.1	No dependencies
25.	FIA_UAU.1	FIA_UID.1 Timing of identification
26.	FIA_AFL.1	FIA_UAU.1 Timing of authentication
	FDP	**User data protection**
27.	FDP_IFC.1/KeyBasics	FDP_IFF.1 Simple security attributes
28.	FDP_IFF.1/KeyBasics	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
29.	FDP_ACC.1/KeyUsage	FDP_ACF.1 Security attribute based access control
30.	FDP_ACF.1/KeyUsage	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
31.	FDP_ACC.1/Backup	FDP_ACF.1 Security attribute based access control
32.	FDP_ACF.1/Backup	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
33.	FDP_SDI.2	No dependencies
34.	FDP_RIP.1	No dependencies
	FTP	**Trusted path/channels**
35.	FTP_TRP.1	No dependencies
	FPT	**Protection of the TSF**
36.	FPT_STM.1	No dependencies
37.	FPT_TST_EXT.1	No dependencies
38.	FPT_PHP.1	No dependencies
39.	FPT_PHP.3	No dependencies
40.	FPT_FLS.1	No dependencies

No.	SFR	Dependency
	FMT	**Security management**
41.	FMT_SMR.1	FIA_UID.1 Timing of identification.
42.	FMT_SMF.1	No dependencies
43.	FMT_MTD.1/AuditLog	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
44.	FMT_MTD.1/SWUpdate	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
45.	FMT_MSA.1/Keys	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
46.	FMT_MSA.3/Keys	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
	FAU	**Security audit data generation**
47.	FAU_GEN.1	FPT_STM.1 Reliable time stamps
48.	FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification
49.	FAU_STG.2	FAU_GEN.1 Audit data generation

Table 2: Security Functional Requirements

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

7.2.1 Cryptographic Support (FCS)

Please note that not all cryptographic algorithms and mechanisms claimed in this ST are also listed in [SOG-IS-Crypto]. The user is responsible for assessing the suitability of a cryptographic service with a specific algorithm in his or her particular use case. Please refer to the corresponding standards that apply such as [SOG-IS-Crypto] or others.

FCS_CKM.1/AES Cryptographic key generation

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

FCS_CKM.1.1/AES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *AES key generation*[7] and specified cryptographic key sizes *of 128, 192 or 256 bit length*[8] that meet the following: *Advanced Encryption Standard (AES) as specified in [FIPS 197] chapters 3.1 and 6, with random number generation according to FCS_RNG.1/DRG.4*[9].

FCS_CKM.1/TDES Cryptographic key generation

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

FCS_CKM.1.1/TDES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *TDES key generation*[10] and specified cryptographic key sizes *of 192 bit length*[11] that meet the following: *TDES as specified in [FIPS 46-3], with random number generation according to FCS_RNG.1/DRG.4*[12].

FCS_CKM.1/GenSecret Cryptographic key generation

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

FCS_CKM.1.1/GenSecret The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *generic secret key generation*[13] and specified cryptographic key sizes *of minimum 13 and maximum 1024 bytes*[14] that meet the following: *generic secret key generation with random number generation according to FCS_RNG.1/DRG.4*[15].

FCS_CKM.1/RSA Cryptographic key generation

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation]

[7] [assignment: cryptographic key generation algorithm]

[8] [assignment: cryptographic key sizes]

[9] [assignment: list of standards]

[10] [assignment: cryptographic key generation algorithm]

[11] [assignment: cryptographic key sizes]

[12] [assignment: list of standards]

[13] [assignment: cryptographic key generation algorithm]

[14] [assignment: cryptographic key sizes]

[15] [assignment: list of standards]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/RSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *RSA key pair generation with pre-defined or given public exponent*[16] and specified cryptographic key sizes *of even key sizes of minimum 2048 and maximum 16,384 bits modulus length*[17] that meet the following: *generation of RSA key pairs according to [FIPS 186-4] Appendix B.3.6, with random number generation according to FCS_RNG.1/DRG.4*[18].

FCS_CKM.1/ECC Cryptographic key generation

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

FCS_CKM.1.1/ECC The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *ECC key pair generation with given elliptic curve domain parameters*[19] and specified cryptographic key sizes *of minimum 224 bits*[20] that meet the following: *ECC key pair generation for ECC domain parameters as shown in Table 3: ECC Domain Parameters below, with random number generation according to FCS_RNG.1/DRG.4*[21].

ECC domain parameters	**Applicable Standard**
NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 or B-571	[FIPS 186-4], appendix D
Brainpool curves: brainpoolP224r1, brainpoolP256r1, brainpoolP320r1, brainpoolP384r1, brainpoolP512r1, brainpoolP224t1, brainpoolP256t1, brainpoolP320t1, brainpoolP384t1 or brainpoolP512t1	[RFC 5639], chapter 3
ANSSI curve: curve FRP256v1	[ANSSI]
SEC 2 curve: secp256k1	[SEC2]
Twisted Edwards curve: edwards25519	[RFC 7748]

[16] [assignment: cryptographic key generation algorithm]

[17] [assignment: cryptographic key sizes]

[18] [assignment: list of standards]

[19] [assignment: cryptographic key generation algorithm]

[20] [assignment: cryptographic key sizes]

[21] [assignment: list of standards]

ECC domain parameters	Applicable Standard
Montgomery curve: curve25519	[RFC 7748]

Table 3: ECC Domain Parameters

FCS_CKM.1/DH Cryptographic key generation

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

FCS_CKM.1.1/DH The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *Diffie-Hellman (DH) key generation*[22] and specified cryptographic key sizes *|P|/|Q| = 2048/224, 2048/256 or 3072/256 bits*[23] that meet the following: *as specified in [FIPS 186-4] Appendix A.1.1.2 and A.2.3 (for FFC domain parameter generation) and Appendix B.1.1 (for key generation), with random number generation according to FCS_RNG.1/DRG.4*[24].

FCS_CKM.2/KeyExport Cryptographic key distribution

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1/KeyExport The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method *key export*[25] that meets the following: *see list of key export methods in Table 4: Key Export and Import Methods below*[26].

FCS_CKM.2/KeyImport Cryptographic key distribution

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

[22] [assignment: cryptographic key generation algorithm]

[23] [assignment: cryptographic key sizes]

[24] [assignment: list of standards]

[25] [assignment: *cryptographic key distribution method*]

[26] [assignment: *list of standards*]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1/KeyImport The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method *key import*[27] that meets the following: *see list of key import methods in Table 4: Key Export and Import Methods below*[28].

Algorithm	Key sizes	Modes, Padding	Standard
AES	128, 192 or 256 bits	AES ECB	see FCS_COP.1/AES_Crypt
		AES CBC	see FCS_COP.1/AES_Crypt
		AES OFB	see FCS_COP.1/AES_Crypt
		AES KW	see FCS_COP.1/AES_Crypt
		AES KWP	see FCS_COP.1/AES_Crypt
		AES CCM	see FCS_COP.1/AES_Crypt
		AES GCM	see FCS_COP.1/AES_Crypt
RSA Encryption scheme	modulus length ≥ 2048 bits, maximum 8192 bits, only even lengths	RSAES-OAEP	see FCS_COP.1/RSA_Crypt
		RSAES-PKCS-v1_5	see FCS_COP.1/RSA_Crypt

Table 4: Key Export and Import Methods

FCS_CKM.4 Cryptographic key destruction

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

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *deletion*[29] that meets the following:

- *overwriting the key by zeroising in case of plaintext secret or private keys,*
- *logical deletion in case of encrypted secret or private keys or public keys*[30].

Application Note 7

Plaintext secret and private keys are destroyed by the method overwriting by zeroising, as required by this SFR.

[27] [assignment: *cryptographic key distribution method*]

[28] [assignment: *list of standards*]

[29] [assignment: cryptographic key destruction method]

[30] [assignment: list of standards]

Furthermore, for permanent storage inside the TOE, the TSF enforces all secret and private keys to be stored encrypted with one of the TOE's internal Master Keys, or by a key which is itself protected by one of the Master Keys. The commands for key deletion delete the encrypted secret and private keys by deletion of the logical addresses, respectively. After that it is no longer possible to address the memory areas of the keys. This ensures secure storage and destruction also for encrypted secret and private keys.

There is no logical access from outside of the TOE to the Master Keys itself. In case of e. g. a physical attack, the Master Keys are protected by the TOE's alarm mechanism and its hard, opaque tamper-evident enclosure. The Master Keys will be actively zeroised in case of an alarm. The Master Key will also actively be erased in case of a Clear command (by actively overwriting it with a new Master Key).

FCS_COP.1/TDES_Crypt Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/TDES_Crypt The TSF shall perform the cryptographic operations *encryption and decryption*[31] in accordance with a specified cryptographic algorithm *Triple DES block cipher in ECB or CBC mode* and cryptographic key sizes *192 bits*[32] that meet the following: *[SP 800-38A] chapter 6.1 (ECB mode) or 6.2 (CBC mode), [FIPS 46-3] (TDES block cipher)*[33].

FCS_COP.1/AES_Crypt Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/AES_Crypt The TSF shall perform *encryption and decryption*[34] in accordance with a specified cryptographic algorithm *AES block cipher in various modes*[35] and cryptographic key sizes *of 16, 24 or 32 bytes length*[36] that meet the following: *[FIPS 197] chapter 5 (for AES block cipher), and applicable standard for block cipher mode as shown in Table 5: AES Block Cipher Modes below*[37].

[31] [assignment: list of cryptographic operations]

[32] [assignment: cryptographic key sizes]

[33] [assignment: list of standards]

[34] [assignment: list of cryptographic operations]

[35] [assignment: cryptographic algorithm]

[36] [assignment: cryptographic key sizes]

[37] [assignment: list of standards]

Block Cipher Mode	Applicable Standard
AES in ECB mode	[SP 800-38A] chapter 6.1
AES in CBC mode	[SP 800-38A] chapter 6.2
AES in OFB mode	[SP 800-38A] chapter 6.4
AES in CTR mode	[SP 800-38A] chapter 6.5
AES in CCM mode	[SP 800-38C]
AES in GCM mode	[SP 800-38D]
AES in KW mode	[SP 800-38F], chapter 6.2
AES in KWP mode	[SP 800-38F], chapter 6.3

Table 5: AES Block Cipher Modes

FCS_COP.1/AES_MAC Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/AES_MAC The TSF shall perform *data integrity protection*[38] in accordance with a specified cryptographic algorithm *AES CMAC or AES GMAC*[39] and cryptographic key sizes *of 16, 24 or 32 bytes length*[40] that meet the following: *[FIPS 197] chapter 5 (for AES block cipher), [SP 800-38B] (for CMAC) and [SP 800-38D] (for GMAC)*[41].

FCS_COP.1/RSA_Sign Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/RSA_Sign The TSF shall perform *the generation and verification of a digital signature*[42] in accordance with a specified cryptographic algorithm *RSA*

[38] [assignment: list of cryptographic operations]

[39] [assignment: cryptographic algorithm]

[40] [assignment: cryptographic key sizes]

[41] [assignment: list of standards]

[42] [assignment: list of cryptographic operations]

signature scheme with appendix according to [PKCS#1], RSASSA-PSS or RSASSA-PKCS-v1_5,[43] and cryptographic key sizes *of even key sizes of minimum 2048 and maximum 16,384 bits modulus length*[44] that meet the following: *[PKCS#1], chapters 8.1.1 or 8.2.1 (signature generation) and chapters 8.1.2 or 8.2.2 (signature verification)*[45].

FCS_COP.1/RSA_Crypt Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/RSA_Crypt The TSF shall perform *encryption and decryption*[46] in accordance with a specified cryptographic algorithm *RSA encryption scheme according to [PKCS#1], RSAES-OAEP or RSAES-PKCS-v1_5,*[47] and cryptographic key sizes *of even key sizes of minimum 2048 and maximum 16,384 bits modulus length*[48] that meet the following: *[PKCS#1], chapters 7.1.1 or 7.2.1 (encryption) and chapters 7.1.2 or 7.2.2 (decryption)*[49].

FCS_COP.1/ECDSA Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/ECDSA The TSF shall perform *the generation and verification of a digital signature*[50] in accordance with a specified cryptographic algorithm *ECDSA*[51] and cryptographic key sizes *of minimum 224 bits*[52] that meet the following: *signature generation and verification according to [ANSI-X9.62], with*

[43] [assignment: cryptographic algorithm]
[44] [assignment: cryptographic key sizes]
[45] [assignment: list of standards]
[46] [assignment: list of cryptographic operations]
[47] [assignment: cryptographic algorithm]
[48] [assignment: cryptographic key sizes]
[49] [assignment: list of standards]
[50] [assignment: list of cryptographic operations]
[51] [assignment: cryptographic algorithm]
[52] [assignment: cryptographic key sizes]

signature keys based on ECC domain parameters as shown in Table 6: ECC Domain Parameters for ECDSA below [53].

ECC domain parameters	Standard
NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 or B-571	[FIPS 186-4], appendix D
Brainpool curves: brainpoolP224r1, brainpoolP256r1, brainpoolP320r1, brainpoolP384r1, brainpoolP512r1, brainpoolP224t1, brainpoolP256t1, brainpoolP320t1, brainpoolP384t1 or brainpoolP512t1	[RFC 5639], chapter 3
ANSSI curve: curve FRP256v1	[ANSSI]
SEC 2 curve: secp256k1	[SEC2][SEC2]

Table 6: ECC Domain Parameters for ECDSA

FCS_COP.1/EdDSA Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/EdDSA The TSF shall perform *the generation and verification of a digital signature*[54] in accordance with a specified cryptographic algorithm *EdDSA*[55] and cryptographic key sizes *of 256 bits*[56] that meet the following: *signature generation and verification according to [RFC 8032] with signature keys based on ECC domain parameters edwards25519 as specified in [RFC 7748]*[57].

FCS_COP.1/HMAC Cryptographic operation

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

[53] [assignment: list of standards]

[54] [assignment: list of cryptographic operations]

[55] [assignment: cryptographic algorithm]

[56] [assignment: cryptographic key sizes]

[57] [assignment: list of standards]

FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HMAC The TSF shall perform *HMAC calculation and verification*[58] in accordance with a specified cryptographic algorithm *HMAC*[59] and cryptographic key sizes *between 4 and 1024 bytes*[60] that meet the following: *[FIPS 198] and [RFC 2104], with hash value calculation according to FCS_COP.1/Hash*[61].

Application Note 8

HMAC calculation and verification in accordance with FCS_COP.1/HMAC and cryptographic key size smaller than 13 bytes can only be used in the context of command authentication. It is not provided as a cryptographic service.

HMAC calculation as a cryptographic service is provided in accordance with FCS_COP.1/HMAC and a minimum key size of 13 bytes.

FCS_COP.1/Hash Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/Hash The TSF shall perform *hash value calculation*[62] in accordance with a specified cryptographic algorithm *SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384 or SHA3-512*[63] and cryptographic key sizes *none*[64] that meet the following: *[FIPS 180-4] chapter 6 for SHA-2, and [FIPS 202] for SHA-3*[65].

FCS_COP.1/DH Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

58 [assignment: list of cryptographic operations]
59 [assignment: cryptographic algorithm]2
60 [assignment: cryptographic key sizes]
61 [assignment: list of standards]
62 [assignment: list of cryptographic operations]
63 [assignment: cryptographic algorithm]
64 [assignment: cryptographic key sizes]
65 [assignment: list of standards]

FCS_COP.1.1/DH The TSF shall perform *shared secret value agreement*[66] in accordance with a specified cryptographic algorithm *Diffie-Hellman (DH) shared secret value agreement*[67] and cryptographic key sizes *|P|/|Q| = 2048/224, 2048/256 or 3072/256 bits*[68] that meet the following: *Diffie-Hellman primitive FFC DH according to [SP 800-56A], chapter 5.7.1.1*[69].

FCS_COP.1/ECDH Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/ECDH The TSF shall perform *shared secret value agreement*[70] in accordance with a specified cryptographic algorithm *Elliptic Curve Diffie-Hellman (ECDH) shared secret value agreement*[71] and cryptographic key sizes *of minimum 224 bits*[72] that meet the following: *ECDH primitive according to the standards as shown in Table 7: ECC Domain Parameters and Standards for ECDH below, with ECDH keys based on ECC domain parameters as shown in Table 7: ECC Domain Parameters and Standards for ECDH below*[73].

ECC domain parameters	**Applicable Standard for ECC domain parameters**	**Applicable Standard for ECDH primitive**
NIST curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 or B-571	[FIPS 186-4], appendix D	[ANSI-X9.63], chapter 5.4.1 (Standard ECDH primitive), or [ANSI-X9.63], chapter 5.4.2 (Modified ECDH primitive, equivalent to ECC CDH primitive according to [SP 800-56A], chapter 5.7.1.2)
Brainpool curves: brainpoolP224r1, brainpoolP256r1, brainpoolP320r1, brainpoolP384r1, brainpoolP512r1, brainpoolP224t1, brainpoolP256t1, brainpoolP320t1, brainpoolP384t1 or brainpoolP512t1	[RFC 5639], chapter 3	
ANSSI curve:	[ANSSI]	

[66] [assignment: list of cryptographic operations]
[67] [assignment: cryptographic algorithm]
[68] [assignment: cryptographic key sizes]
[69] [assignment: list of standards]
[70] [assignment: list of cryptographic operations]
[71] [assignment: cryptographic algorithm]
[72] [assignment: cryptographic key sizes]
[73] [assignment: list of standards]

ECC domain parameters	Applicable Standard for ECC domain parameters	Applicable Standard for ECDH primitive
curve FRP256v1		
SEC 2 curve: secp256k1	[SEC2]	
Montgomery curve: Curve25519	[RFC 7748]	[ANSI-X9.63], chapter 5.4.1 (Standard ECDH primitive), or [RFC 7748], chapter 6

Table 7: ECC Domain Parameters and Standards for ECDH

FCS_COP.1/KeyDerivation Cryptographic operation

Hierarchical to: No other components.
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/KeyDerivation The TSF shall perform *key derivation*[74] in accordance with a specified cryptographic algorithm *KDF in Feedback Mode with HMAC*[75] and cryptographic key sizes *4-1024 bytes*[76] that meet the following: *[SP 800-108], chapter 5.2, with HMAC calculation according to FCS_COP.1/HMAC*[77].

Application Note 9

Key Derivation in accordance with FCS_COP.1/KeyDerivation can only be used in the context of establishing a Secure Messaging session (trusted channel according to FTP_TRP.1) and for the backup of cryptographic keys (FDP_ACC.1/Backup, FDP_ACF.1/Backup). It is not provided as a cryptographic service.

FCS_RNG.1/PTG.2 Random number generation

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

FCS_RNG.1.1/PTG.2 The TSF shall provide a *physical*[78] random number generator that implements *the security capabilities of RNG class PTG.2 of [AIS 20/31] chapter 4.4:*

[74] [assignment: list of cryptographic operations]

[75] [assignment: cryptographic algorithm]

[76] [assignment: cryptographic key sizes]

[77] [assignment: list of standards]

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

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

(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source[79].

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

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

(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered continuously[80]. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time. [81]

FCS_RNG.1.2/PTG.2 The TSF shall provide *octets of bits*[82] that meet *the quality metric of RNG class PTG.2 of [AIS 20/31] chapter 4.4*:

(PTG.2.6) Test procedure A and none[83] *does not distinguish the internal random numbers from output sequences of an ideal RNG.*

(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997. [84]

FCS_RNG.1/DRG.4 Random number generation

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

FCS_RNG.1.1/DRG.4 The TSF shall provide a *hybrid deterministic*[85] random number generator that implements *the security capabilities of RNG class DRG.4 of [AIS 20/31] chapter 4.9:*

[79] [AIS 20/31]: [selection: *prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy*]

[80] [AIS 20/31]: [selection: *externally, at regular intervals, continuously, applied upon specified internal events*]

[81] [assignment: list of security capabilities]

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

[83] [AIS 20/31]: [assignment: *additional standard test suites*]

[84] [assignment: a defined quality metric]

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

(DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2 as random source[86]*.*

(DRG.4.2) The RNG provides forward secrecy.

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

(DRG.4.4) The RNG provides enhanced forward secrecy on condition[87] *that 1000 requests for pseudo random bits have been made after last entropy input during instantiation or reseeding*[88]*.*

(DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2 according to FCS_RNG.1/PTG.2 [89] [90]*.*

FCS_RNG.1.2/DRG.4 The TSF shall provide *octets of bits*[91] that meet *the quality metric of RNG class DRG.4 of [AIS 20/31] chapter 4.9*:

(DRG.4.6) The RNG generates output for which 2·10[10] [92] *strings of bit length 128 are mutually different with probability 0.99998.*[93]

(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A[94] [95]*.*

7.2.2 Identification and Authentication (FIA)

FIA_UID.1 Timing of identification

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

FIA_UID.1.1 The TSF shall allow

(1) *Self-test according to FPT_TST_EXT.1,*

(2) *usage of commands where no user authentication is needed, including requests for the status of the TOE*[96]*,*

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

---

[86] [AIS 20/31]: [selection: *use PTRNG of class PTG.2 as random source, have [assignment: work factor], require [assignment: guess work]*]

[87] [AIS 20/31]: [selection: *on demand, on condition [assignment: condition], after [assignment: time]*]

[88] [AIS 20/31]: [*condition*]

[89] [AIS 20/31]: [selection: *selection: internal entropy source, PTRNG of class PTG.2, PTRNG of class PTG.3, [other selection]*]

[90] [assignment: list of security capabilities]

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

[92] [AIS 20/31]: [assignment: *number of strings*]

[93] [AIS 20/31]: [assignment: *probability*]

[94] [AIS 20/31]: [assignment: *additional test suites*]

[95] [assignment: a defined quality metric]

[96] [assignment: *list of additional 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.

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.
Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow

(1) *Self-test according to FPT_TST_EXT.1,*

(2) *Identification of the user by means of TSF required by FIA_UID.1,*

(3) *usage of commands where no user authentication is needed, including requests for the status of the TOE*[97]*.*

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

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

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.
Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when *one*[98] unsuccessful authentication attempts occur related to *consecutive failed authentication attempts*.[99]

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *met*[100], the TSF shall *block the respective user for access to a successful authentication attempt until a time period of 4 seconds has elapsed*[101]*.*

7.2.3 User Data Protection (FDP)

FDP_IFC.1/KeyBasics Subset information flow control

Hierarchical to: No other components.
Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/KeyBasics The TSF shall enforce the *Key Basics SFP*[102] on

[97] [assignment: *list of TSF mediated functions*]

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

[99] [PP_CMTS] [assignment: *list of authentication events*]

[100] [selection: *met, surpassed*]

[101] [PP_CMTS] [assignment: *list of actions*]

[102] [PP_CMTS] [assignment: *information flow control SFP*]

(1) *subjects: all*
(2) *information: keys*
(3) *operations: all*[103].

FDP_IFF.1/KeyBasics Simple security attributes

Hierarchical to: No other components.
Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/KeyBasics The TSF shall enforce the *Key Basics SFP*[104] based on the following types of subject and information security attributes:

(1) *whether a key is a secret or a public key*
(2) *whether channels selected to export keys are secure*
(3) *the value of the Export Flag of a key*[105].

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

(1) *Export of secret keys shall only be allowed provided that the secret key is encrypted, and that a secure channel (providing authentication and integrity protection) is used for the export, with the secure channel meeting the requirements of FTP_TRP.1*
(2) *Public keys shall always be exported with integrity protection of their key value and attributes*
(3) *Keys shall only be imported over a secure channel (providing authentication and integrity protection), with the secure channel meeting the requirements of FTP_TRP.1*
(4) *Secret keys shall only be imported in encrypted form*[106].

FDP_IFF.1.3/KeyBasics The TSF shall enforce the **following additional information flow control rules**: *none*[107].

FDP_IFF.1.4/KeyBasics The TSF shall explicitly authorise an information flow based on the following rules: *none*[108].

[103] [PP_CMTS] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]*
[104] [PP_CMTS] [assignment: *information flow control SFP*]
[105] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]
[106] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]
[107] [PP_CMTS] [assignment: *additional information flow control SFP rules*]
[108] [PP_CMTS] [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

FDP_IFF.1.5/KeyBasics The TSF shall explicitly deny an information flow based on the following rules:

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

(2) No subject shall be allowed to export a secret key in plaintext.

(3) No subject shall be allowed to export a secret key without being currently authorised to do so.

(4) No subject shall be allowed to access intermediate values in any operation that uses a secret key.

(5) A key with an Export Flag value marking it as non-exportable shall not be exported[109]*.*

Application Note 10 (inspired by [PP_CMTS])
The requirements of FDP_IFF.1/KeyBasics apply regardless of how the key is stored by the TOE, including when the key is externally stored (cf. [PP_CMTS], section 1.3.1.2).
Direct access to a key value in FDP_IFF.1.5/KeyBasics (1) is access that makes the value available for reading or modification – this includes operations that would subsequently allow reading or modification of the key (e.g. making a copy of the key with different attributes, or with a different object type that would then allow direct read access). Note that the PP [PP_CMTS] assumes that key values are never modified after they have been generated.
Export of a key as in FDP_IFF.1.5/KeyBasics (1), (2), (3) and (5) is not the same as backup (governed by FDP_ACF.1/Backup) or external storage of keys under continuing TOE control (governed by other parts of the Key Basics SFP in FDP_IFF.1/KeyBasics, and the Key Usage SFP in FDP_ACF.1/KeyUsage). Thus an Export Flag of 'non-exportable' does not prevent backup or external storage of the keys under continuing TOE control.

FDP_ACC.1/KeyUsage Subset access control

Hierarchical to: No other components.
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/KeyUsage The TSF shall enforce the *Key Usage SFP*[110] to objects based on the following

(1) Subjects: all;

(2) Object: Keys

(3) Operations: all[111]

FDP_ACF.1/KeyUsage Security attribute based access control

Hierarchical to: No other components.
Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

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

[110] [PP_CMTS] [assignment: *access control SFP*]

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

FDP_ACF.1.1/KeyUsage The TSF shall enforce the *Key Usage SFP*[112] to objects based on the following:

(1) whether the subject is currently authorised to use the secret key

(2) whether the subject is currently authorised to change the attributes of the secret key

(3) the cryptographic function that is attempting to use the secret key[113]*.*

FDP_ACF.1.2/KeyUsage The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) Attributes of a key shall only be changed by an authorised subject, and only as permitted in the Key Attributes Modification Table (see Table 8: Key Attribute Modification Table)

(2) Only subjects with current authorisation for a specific secret key shall be allowed to carry out operations using the plaintext value of that key

(3) Only cryptographic functions permitted by the secret key's Key Usage attribute shall be carried out using the secret key[114]*.*

FDP_ACF.1.3/KeyUsage The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*[115].

FDP_ACF.1.4/KeyUsage The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[116].

Application Note 11 (from [PP_CMTS])

The requirements of FDP_ACF.1/KeyUsage apply regardless of how the key is stored by the TOE, including when the key is externally stored (cf. [PP_CMTS], section 1.3.1.2).

FDP_ACC.1/Backup Subset access control

Hierarchical to: No other components.
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Backup The TSF shall enforce the *Backup SFP*[117] on

(1) subjects: all

(2) objects: keys

[112] [PP_CMTS] [assignment: *access control SFP*]

[113] [PP_CMTS] [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*]

[114] [PP_CMTS] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

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

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

[117] [PP_CMTS] [assignment: *access control SFP*]

(3) operations: backup, restore[118]*.*

FDP_ACF.1/Backup Security attribute based access control

Hierarchical to: No other components.
Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/Backup The TSF shall enforce the *Backup SFP*[119] to objects based on the following:

(1) whether the subject is an administrator[120]*.*

FDP_ACF.1.2/Backup The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) Only authorised administrators shall be able to perform any backup operation provided by the TSF to create backups of the TSF state or to restore the TSF state from a backup

(2) Any backup and restore shall preserve the confidentiality and integrity of the secret keys, and the integrity of public keys

(3) Any backup and restore operations shall preserve the integrity of the key attributes, and the binding of each set of attributes to its key[121]*.*

FDP_ACF.1.3/Backup The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*[122].

FDP_ACF.1.4/Backup The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[123]

Application Note 12 (inspired by [PP_CMTS])

Preserving the binding of a set of attributes to its key (in FDP_ACF.1.2/Backup (3)) means that it is not possible for the attributes to be changed during a backup operation, or by modification of the backup data while it is away from the TSF.

Backups may contain keys whose export flag attribute marks them as 'non-exportable'.

The following iterations of FCS_COP.1 are used to protect confidentiality and integrity of any supported backups:

- *FCS_COP.1/AES_Crypt*
- *FCS_COP.1/AES_MAC*

---

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

[119] [PP_CMTS] [assignment: access control SFP]

[120] [PP_CMTS] [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*]

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

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

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

- *FCS_COP.1/KeyDerivation*

FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to: FDP_SDI.1 Stored data integrity monitoring.
Dependencies: No dependencies.

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for *integrity errors*[124] on all **keys (including security attributes)**[125], based on the following attributes: *integrity protection data*[126].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall
(1) prohibit the use of the altered data
(2) notify the error to the user[127].

Application Note 13 (inspired by [PP_CMTS])
The protection measures provided by this SFR are supported by AES CMAC algorithm according to FCS_COP.1/AES_MAC.

This SFR is also used in the implementation of the mechanism for protection against modification access to the value of a secret key in FDP_IFF.1.5/KeyBasics, and in the requirement for export of public keys with integrity protection in FDP_IFF.1.2/KeyBasics.

The integrity protection data in FDP_SDI.2.1 is included in the list of attributes identified in FMT_MSA.1/Keys, and protects the value of the key and of its other security attributes, including when the key is externally stored by the TOE (cf. [PP_CMTS]*, section 1.3.1.2).*

FDP_RIP.1 Subset residual information protection

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

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *de-allocation of the resource from*[128] the following objects:
(1) secret authentication data
(2) secret keys[129].

[124] [PP_CMTS] [assignment: *integrity errors*]

[125] [PP_CMTS] objects

[126] [PP_CMTS] [assignment: *user data attributes*]

[127] [PP_CMTS] [assignment: *action to be taken*]

[128] [PP_CMTS] [selection: *allocation of the resource to, de-allocation of the resource from*]

[129] [PP_CMTS] [assignment: *list of objects*]

7.2.4 Trusted Path/Channels (FTP)

FTP_TRP.1 Trusted Path

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

FTP_TRP.1.1 The TSF shall provide a communication path between itself and *remote or local*[130] ***client applications***[131] that are logically distinct from other communication paths and provides assured ***authentication***[132] of its end points and protection of the communicated data from *modification and disclosure*[133].

FTP_TRP.1.2 The TSF shall permit *remote or local* ***client applications***[134] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for *protecting the confidentiality and integrity of sensitive data exchanged between the client application and the TOE over a channel that passes through an insecure environment*[135].

Application Note 14

Although local client applications and remote external client applications may run in different environments they have to use the identically same trusted communication mechanisms to communicate with the TOE, which are implemented by cryptographic means and supported by iterations of FCS_COP.1.

7.2.5 Protection of the TSF (FPT)

FPT_STM.1 Reliable time stamps

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

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

Application Note 15 (inspired by [PP_CMTS])

The TOE must provide time stamps suitable for supporting the time in an audit record for FAU_GEN.1.

---

130 [selection: *remote, local*]

131 [PP_CMTS] users

132 [PP_CMTS] identification

133 [PP_CMTS] [selection: *modification, disclosure, [assignment: other types of integrity or confidentiality violation]*]

134 [selection: *the TSF, local users, remote users*]

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

FPT_TST_EXT.1 Basic TSF Self Testing

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

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests *during initial start-up (on power-on or reset) and at the conditions firmware download, PTRNG request, DRBG request and key pair generation*[136] to demonstrate the correct operation of the TSF:

- *At initial start-up (on power-on or reset):*
  - *Software/firmware integrity test*
  - *Cryptographic algorithm tests*
  - *Random number generator tests*
- *At firmware download:*
  - *Firmware download test (via ECDSA signature verification)*
- *At each PTRNG request:*
  - *PTRNG online test according to [AIS 20/31] for RNG class PTG.2 and continuous health tests according to [SP 800-90B] §4.4.1 and §4.4.2*
- *At each DRBG request:*
  - *Conditional DRBG test according to [FIPS 140-2] §4.9.2*
- *At key pair generation:*
  - *Pair-wise consistency test according to [FIPS 140-2] §4.9.2*[137].

FPT_PHP.1 Passive detection of physical attack

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

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.
FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

Application Note 16 (from [PP_CMTS])

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

Because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 for this TOE is equivalent to the physical security mechanisms for tamper detection and response required by section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements for each physical

---

136 [selection: *during initial start-up (on power on or reset), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*]

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

security embodiment in ISO/IEC 19790:2012 [ISO/IEC 19790:2012] for Security Level 3. (Cf. refinement of AVA_VAN.5 in section 7.3.1.)

FPT_PHP.3 Resistance to physical attack

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

FPT_PHP.3.1 The TSF shall resist *physical manipulation*[138] to the *entire TOE components implementing the TSF*[139] by responding automatically such that the SFRs are always enforced.

Application Note 17 (from [PP_CMTS])

This SFR is linked to the requirements for passive detection of physical attacks in FPT_PHP.1, and should identify the relevant responses of the TOE involved in meeting the key zeroisation requirements of ISO/IEC 19790:2012 [ISO/IEC 19790:2012] Security Level 3. As in the case of FPT_PHP.1, because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.3 for this TOE is equivalent to the level of assessment for this aspect of tamper detection and response required for section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements by each physical security embodiment in ISO/IEC 19790:2012 [ISO/IEC 19790:2012] for Security Level 3. (Cf. refinement of AVA_VAN.5 in section 7.3.1.)

FPT_FLS.1 Failure with preservation of secure state

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

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

1. *Self-test according to FPT_TST_EXT.1 fails*
2. *Environmental conditions are outside normal operating range (including temperature and power)*
3. *Failures of critical TOE hardware components (including the RNG) occur*
4. *Corruption of TOE software occurs*
5. *Failures caused by sensitive TOE software components*[140].[141]

[138] [assignment: physical tampering scenarios]

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

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

[141] [PP_CMTS] [assignment: *list of types of failures in the TSF*]

7.2.6 Security Management (FMT)

In the FMT_MSA SFRs specified for cryptographic user keys, the permitted values of assignments have been restricted to identify a minimum set of attributes that must be mapped to their implementation in the TOE, and to specify a minimum set of constraints on their initialisation and subsequent modification. Additional notes regarding these attributes are as follows:

- key identifier: this must be sufficient to uniquely identify the key within the system of which the TOE is a part
- key type: this identifies at a minimum whether the key is a secret key of a symmetric cryptographic algorithm or the secret (commonly called private) key of an asymmetric cryptographic algorithm
- key usage: the cryptographic functions that are allowed to use the key as detailed in FDP_ACF.1/KeyUsage
- export flag: indicates whether the key is allowed to be exported (cf. FDP_IFF.1/KeyBasics); allowed values are referred to in this ST as 'true' (meaning that export is allowed) and 'false' (meaning that export is not allowed) but will be mapped to other suitable binary values in the TOE implementation.

FMT_SMR.1 Security roles

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1	The TSF shall maintain the roles *Global Administrator, cHSM Administrator, User Administrator, Key Manager, Security Officer, Key User, Client Application*[142].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

Application Note 18 (inspired by [PP_CMTS])

The (local or remote) Client Application role represents an identifiable subject that communicates with the TOE over a secure channel, which may either be locally, i.e. located within the same hardware appliance, or remotely, and in line with FTP_TRP.1.

The Key User role represents a normal, unprivileged subject who can invoke operations on a key according to the authorisation requirements – this role may sometimes act through a client application.

Application Note 19

The TOE implements the following roles for the different users:

- *Administrator Roles*
  - *Global Administrator (global user management; global system management incl. setting the system time and global configuration; firmware update; cHSM management)*
  - *cHSM Administrator (administration of a cHSM, like container audit log deletion)*

[142] [assignment: *the authorised identified roles*]

  - *User Administrator (cHSM user management tasks, like creation of users or deletion of users within a cHSM)*
  - *Key Manager (key management on cHSM level, like key generation, key export and import, key backup and key restore, key deletion of cryptographic keys within a cHSM)*
  - *SO (Security Officer) (creating, modifying or deleting key group specific configuration objects and initiating key groups where he belongs to, on cHSM level)*
- *Key User (uses a cHSM for cryptographic operations like signature creation)*
- *Client Application (uses the TOE for creating a secure channel; thus each authenticated user can in addition assume the role Client Application)*

FMT_SMF.1 Specification of management functions

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

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

1. *Modifying attributes of keys*
2. *Export and deletion of the audit data, which can take place only under the control of the Administrator role*
3. *backup and restore functions for keys and containers (cHSMs)*[143]
4. *key import function*[144]
5. *key export function*[145]
6. *time adjustment (FTP_SMT.1)*
7. *software update function (FMT_MTD.1/SWUpdate)*[146]*.*

Application Note 20 (inspired by [PP_CMTS])

The attributes of keys in FMT_SMF.1.1 (1) correspond to the attributes in FMT_MSA.1/Keys. Export of audit data in FMT_SMF.1.1 (2) relates to the ability to export audit data from the TOE for preservation and storage elsewhere.

FMT_MTD.1/AuditLog Management of TSF data

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

[143] [selection: backup and restore functions, no backup and restore functions]

[144] [selection: key import function, no key import function]

[145] [selection: key export function, no key export function]

[146] [assignment: *list of management functions to be provided by the TOE*]

FMT_MTD.1.1/AuditLog The TSF shall restrict the ability to *control export and deletion of*[147] the *audit log records*[148] to the *Administrator role*[149].

Application Note 22 (from [PP_CMTS])

The control of export and deletion of the audit log records helps to ensure their protection against accidental or malicious deletion (deletion should normally occur only after the records have been exported and preserved outside the TOE). Note that this does not require the Administrator to carry out these export or delete operations manually as long as the actions are controlled by the Administrator.

FMT_MTD.1/SWUpdate Management of TSF data

Hierarchical to: No other components.
Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/SWUpdate The TSF shall restrict the ability to *update*[147] the *TSF executable code stored in the TOE in form of software or firmware*[148] to *the Global Administrator role*[149].

FMT_MSA.1/Keys Management of security attributes

Hierarchical to: No other components.
Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/Keys The TSF shall enforce the *Key Usage SFP*[150] to restrict the ability to *modify*[151] the security attributes *as specified in the Table 8: Key Attribute Modification Table*[152] to *the subjects, keys, and operations among subjects and keys as specified in the Table 8: Key Attribute Modification Table*[153].

Key Attribute	Modification operation policy
Key ID	Cannot be modified
Key Type	Cannot be modified
Key Algorithm	Cannot be modified

147 [PP_CMTS] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

148 [PP_CMTS] [assignment: *list of TSF data*]

149 [PP_CMTS] [assignment: *the authorised identified roles*]

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

151 [PP_CMTS] [selection: *change_default, query, modify, delete, [assignment: other operations]*]

152 [assignment: *list of security attributes*]

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

Key Attribute	Modification operation policy
Key Usage	Cannot be modified
Export Flag	Can only be modified by users in role *Key Manager*, and only to change from 'true' (meaning that export is allowed) to 'false' (meaning that export is not allowed
Integrity Protection Data	Cannot be modified by users (maintained automatically by TSF)

Table 8: Key Attribute Modification Table

FMT_MSA.3/Keys Static attribute initialisation

Hierarchical to: No other components.
Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1/Keys The TSF shall enforce the *Key Usage SFP*[154] to provide *restrictive*[155] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Keys The TSF shall allow the *Key Manager according to the constraints in the Table 9: Key Attribute Initialisation Table*[156] to specify alternative initial values to override the default values when an object or information is created.

Key Attribute	Initialisation operation policy
Key ID	Initialised automatically by generation process
Key Type	Initialised by generation process
Key Algorithm	Initialised by generation process
Key Usage	Initialised by a user in role Key Manager during key creation on TOE (key creation by key generation, key import, key derivation, key copy), or with dedicated *SetKeyAttribute* command, or automatically at first usage of the key. A key that has been used for one mechanism group cannot be used for any other of the mechanism groups: • signature creation and/or verification • encryption and/or decryption • key transport • key agreement/derivation

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

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

[156] [PP_CMTS] [assignment: *the authorised identified roles according to the constraints in the Key Attribute Initialisation Table*]

Key Attribute	Initialisation operation policy
Export Flag	Initialised by generation process (default: false, i.e. no export allowed)
Integrity Protection Data	Initialised automatically by TSF

Table 9: Key Attribute Initialisation Table

Application Note 23

The Integrity Protection Data for a key is used to support FDP_SDI.2 and covers not only the key but also its other attributes.

7.2.7 Security Audit Data Generation (FAU)

FAU_GEN.1 Audit data generation

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

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the *not specified*[157] level of audit; and[158]
c) Startup of the TOE;
d) Cryptographic key generation (FCS_CKM.1 (all iterations));
e) Cryptographic key destruction (FCS_CKM.4);
f) Failure of the random number generator (FCS_RND.1 (all iterations));
g) Authentication failures (FIA_AFL.1);
h) All attempts to import or export keys (FDP_IFF.1/KeyBasics);
i) All modifications to attributes of keys (FDP_ACF.1/KeyUsage, FMT_MSA.1/Keys);
j) Backup and restore (FDP_ACF.1/Backup): use of any backup function, use of any restore function, unsuccessful restore because of detection of modification of the backup data;
k) Integrity errors detected for keys (FDP_SDI.2);
l) Failures to establish secure channels (FTP_TRP.1);
m) Self-test completion (FPT_TST_EXT.1);
n) Failures detected by the TOE (FPT_FLS.1);
o) All administrative actions (FMT_SMF.1, FMT_MSA.1/Keys, FMT_MSA.3/Keys,);
p) Adjustment of the internal clock by an administrator (FPT_STM.1);

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

[158] [PP_CMTS] Levels of audit are not required to be defined in the Security Target.

q) *Modifications to audit parameters (affecting the content of the audit log) (FAU_GEN.1);*

r) *none*[159].

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:

- *none*[160].

Application Note 24

For some events logging of the audit events is optional and depends on the configuration. All configuration options for the audit log and the respective default values are described in the Operational Guidance.

FAU_GEN.2 User identity association

Hierarchical to: No other components.
Dependencies: FAU_GEN.1 Audit data generation

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

FAU_STG.2 Guarantees of audit data availability

Hierarchical to: FAU_STG.1 **Protected** audit trail storage
Dependencies: FAU_GEN.1 Audit data generation

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

FAU_STG.2.2 The TSF shall be able to *prevent*[161] unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.2.3 The TSF shall ensure that *all*[162] stored audit records will be maintained when the following conditions occur: *audit storage exhaustion*[163].

[159] [assignment: *other specifically defined auditable events*]

[160] [assignment: other audit relevant information]

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

[162] [PP_CMTS] [assignment: *metric for saving audit records*]

[163] [PP_CMTS] [selection: *audit storage exhaustion, failure, attack*]

7.3 Security Assurance Requirements

The security assurance requirement level is **EAL4** augmented with **AVA_VAN.5** and **ALC_FLR.3.** The assurance components are identified in the table below (with augmentations in bold). It is noted that due to the physically protected environment in which the TOE operates (as expressed in OE.Env), it is unlikely that physical attacks will be within the scope of an evaluation against this ST (as derived from [PP_CMTS]).

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 procedures (AGD_PRE.1)
Life cycle support (ALC)	Production support, acceptance procedures and automation (ALC_CMC.4)
	Problem tracking CM coverage (ALC_CMS.4)
	Delivery procedures (ALC_DEL.1)
	Identification of security measures (ALC_DVS.1)
	Systematic flaw remediation (ALC_FLR.3)
	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)**

Table 10: Security Assurance Requirements

7.3.1 Refinement of Security Assurance Requirements

The following refinement is made to selected assurance requirements in Table 10, in line with the refinement applied by the PP [PP_CMTS] to the SAR AVA_VAN.5:

AVA_VAN.5 Advanced methodical vulnerability analysis

Refinement:
Regarding the protection of the TOE against physical attacks: because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 and FPT_PHP.3 for this TOE is equivalent to the level of assessment for this aspect of tamper detection and response in section 7.7.2 *Physical security general requirements* and section 7.7.3 *Physical security requirements for each physical security embodiment* in ISO/IEC 19790:2012 [ISO/IEC 19790:2012] for Security Level 3.

8 Rationales

8.1 Security Objectives Rationale

8.1.1 Security Objectives Rationale

The table below shows the mapping of Threats, Organisational Security Policies and Assumptions to Security Objectives for the TOE and for the TOE Environment.

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit	OE.ExternalData	OE.Env	OE.Datacontext	OE.AppSupport	OE.Uauth	OE.AuditSupport
T.KeyDisclose	X		X			X		X	X		X			X	X				
T.KeyDerive		X								X									
T.KeyMod			X					X	X		X								
T.KeyMisuse				X	X														
T.DataDisclose						X										X	X		
T.DataMod							X									X	X		
T.Malfunction												X							
P.Algorithm		X																	
P.KeyControl	X	X		X	X			X	X										
P.RNG										X									
P.Audit													X						
A.ExternalData														X					
A.Env															X				
A.DataContext																X			
A.AppSupport																	X		
A.UAuth																		X	
A.AuditSupport																			X

Table 11: Security Problem Definition mapping to Security Objectives

8.1.2 Security Objectives Sufficiency

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

8.1.2.1 Threats

T.KeyDisclose is addressed by the requirement in OT.PlainKeyConf to keep plaintext secret keys unavailable, and this is supported in terms of controls over key attributes (which might threaten the confidentiality of the key if modified) in OT.KeyIntegrity. The confidentiality of secret keys that are exported is protected partly by the use of a secure channel as described in OT.DataConf and the requirements for import and export in OT.ImportExport (including the requirement to export secret keys only in encrypted form, or to be able to exclude the export of a key entirely). Physical tamper protection of the keys is provided by OT.TamperDetect (supported by an appropriate inspection procedure as required in OE.Env). Protection of secret key confidentiality during backup is ensured by OT.Backup. The environment also contributes to maintaining secret key confidentiality by protecting any versions of a secret key that may exist outside the TOE, as in OE.ExternalData, and by protecting the operation of the TOE itself by providing a secure environment, as in OE.Env.

T.KeyDerive is addressed by the choice of algorithms that have been endorsed for the appropriate purposes, and this is described in OT.Algorithms. Where keys are generated by the TOE then the use of a suitable random number generator is required by OT.RNG in order to mitigate the risk that an attacker can guess or deduce the key value.

T.KeyMod is addressed by requiring integrity protection of secret and public keys, and their critical attributes in OT.KeyIntegrity, and by requiring use of secure channels that protect integrity if a key is imported or exported (OT.ImportExport). Protection of key integrity during backup is ensured by OT.Backup. Physical tamper protection of the keys is provided by OT.TamperDetect (supported by an appropriate inspection procedure as required in OE.Env).

T.KeyMisuse raises the possibility of a secret key being used for an unintended and unauthorised purpose, and is addressed by the requirement in OT.Auth for the TOE to carry out an authorisation check before using a secret key. OT.KeyUseConstraint expands on this to set out requirements for the granularity of authorisation.

T.DataDisclose is concerned with the transmission of data between client applications and the TOE, or between separate parts of the TOE where the transmission passes through an insecure environment. This is addressed by OT.DataConf, which requires the TOE to provide secure channels to protect such communications. The appropriate use of such channels is a requirement for the environment as expressed in OE.DataContext, as is the use of appropriate procedures in OE.AppSupport.

T.DataMod is concerned with the possibility of unauthorised modification of data transmitted between a client application and the TOE, and this is addressed by OT.DataMod which requires that the TOE provides secure channels that can be used to protect the integrity of data that they carry. As with T.DataDisclose, the appropriate use of such channels is a requirement for the environment as expressed in OE.DataContext, as is the use of appropriate procedures in OE.AppSupport.

T.Malfunction is addressed by the requirement in OT.FailureDetect for the TOE to detect certain types of fault.

8.1.2.2 Organisational Security Policies

P.Algorithms requires the use of key generation and other cryptographic functions that are endorsed by appropriate authorities, and this is addressed by OT.Algorithms.

P.KeyControl requires that the TOE can provide controls and support a key lifecycle to ensure that secret keys can be reliably protected against use by users or entities that are not authorized to use the keys, and that the keys can be confined to use for certain cryptographic functions. This is addressed by a combination of TOE objectives as follows:

- OT.PlainKeyConf protects the value of the secret key to prevent the possibility of it being used by unauthorised subjects
- OT.Algorithms ensures that endorsed algorithms that employ and support suitable properties and procedures are provided by the TOE
- OT.Auth and OT.KeyUseConstraint ensure that the TOE can provide well-defined limits on the use of a key when it is authorised (as described above for T.KeyMisuse)
- OT.ImportExport and OT.Backup ensure protection of keys when they are transmitted outside the TOE to client applications or for backup purposes.

P.RNG is directly addressed by TOE objective OT.RNG, with nearly identical wording.

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

8.1.2.3 Assumptions

Each of the Assumptions in section 4.5 is directly matched by a security objective for the operational environment in section 5.2. The wording of each objective for the operational environment includes the wording of each assumption, and no further rationale is therefore given here.

8.2 Functional Security Requirements Rationale

8.2.1 Security Requirements Coverage

The table below summarises the mapping of Security Objectives for the TOE to SFRs.

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FCS_CKM.1/AES		X											
FCS_CKM.1/TDES		X											
FCS_CKM.1/GenSecret		X											
FCS_CKM.1/RSA		X											
FCS_CKM.1/ECC		X											
FCS_CKM.1/DH		X											
FCS_CKM.2/KeyExport		X						X					
FCS_CKM.2/KeyImport		X						X					
FCS_CKM.4	X												
FCS.COP.1/TDES		X											
FCS.COP.1/AES_Crypt		X											
FCS.COP.1/AES_MAC		X											
FCS_COP.1/RSA_Sign		X											
FCS_COP.1/RSA_Crypt		X											
FCS_COP.1/ECDSA		X											
FCS_COP.1/EdDSA		X											
FCS_COP.1/HMAC		X											
FCS_COP.1/Hash		X											
FCS_COP.1/DH		X											
FCS_COP.1/ECDH		X											
FCS_COP.1/KeyDerivation		X											
FCS_RNG.1/PTG.2										X			

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FCS_RNG.1/DRG.4										X			
FIA_UID.1				X									
FIA_UAU.1				X									
FIA_AFL.1				X									
FDP_IFC.1/KeyBasics	X			X	X			X					
FDP_IFF.1/KeyBasics	X		X	X	X			X					
FDP_ACC.1/KeyUsage				X	X								
FDP_ACF.1/KeyUsage				X	X								
FDP_ACC.1/Backup									X				
FDP_ACF.1/Backup									X				
FDP_SDI.2			X										
FDP_RIP.1	X				X								
FTP_TRP.1			X	X		X	X	X					
FPT_STM.1													X
FPT_TST_EXT.1												X	
FPT_PHP.1											X		
FPT_PHP.3											X		
FPT_FLS.1												X	
FMT_SMR.1				X									X
FMT_SMF.1				X									X
FMT_MTD.1/AuditLog													X
FMT_MTD.1/SWUpdate				X									

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FMT_MSA.1/Keys					X								
FMT_MSA.3/Keys					X								
FAU_GEN.1													X
FAU_GEN.2													X
FAU_STG.2													X

Table 12: TOE Security Objectives mapping to SFRs

OT.PlainKeyConf is addressed by the requirements in the Key Basics SFP defined in FDP_IFC.1/KeyBasics and FDP_IFF.1/KeyBasics (especially FDP_IFF.1.5/KeyBasics). Secure destruction of keys according to FCS_CKM.4 protects the key value at the end of its lifetime. FDP_RIP.1 protects secret keys from being accessed after they have been deallocated.

OT.Algorithms is addressed by the need to use endorsed standards for all iterations of FCS_COP.1 and FCS_CKM.2 and the use of an appropriate random number generator in FCS_CKM.1 (all iterations).

OT.KeyIntegrity is addressed primarily by FDP_SDI.2 which requires integrity protection of keys and their attributes by the TOE. FDP_IFF.1/KeyBasics requires that any importing or exporting of keys requires the use of secure channels and integrity protection (cf. the requirement for an integrity protected channel as part of FTP_TRP.1, which is linked to the Key Basics SFP by FDP_IFF.1.2/KeyBasics (1) and (3)).

OT.Auth is addressed by FIA_UID.1, FIA_UAU.1, and FIA_AFL.1 for user authentication (with FMT_MTD.1/AuditLog, FMT_MTD.1/SWUpdate, and its dependencies on FMT_SMR.1 and FMT_SMF.1 ensuring that appropriate roles are provided). Authorisation for external client applications is provided by the requirements for authentication of endpoints in FTP_TRP.1. Authorisation for access to a secret key is additionally addressed by FDP_IFC.1/KeyBasics, FDP_IFF.1.5/KeyBasics (3), FDP_ACC.1/KeyUsage and FDP_ACF.1.2/KeyUsage (1) and (2).

OT.KeyUseConstraint is addressed by the requirements for well-defined (and securely initialised) key attributes in FMT_MSA.1/Keys and FMT_MSA.3/Keys, and the application of the attributes to operate constraints on the use of keys in FDP_IFC.1/KeyBasics, FDP_IFF.1/KeyBasics, FDP_ACC.1/KeyUsage and FDP_ACF.1/KeyUsage. FDP_RIP.1 protects secret authentication data (which enables a key to be used) from being accessed after it has been deallocated.

OT.DataConf is addressed by the authentication and confidentiality requirements for secure channels in FTP_TRP.1.

OT.DataMod is addressed by the authentication and integrity requirements for secure channels in FTP_TRP.1.

OT.ImportExport is addressed by the requirements for the use of secure import/export through a secure channel and restrictions on how keys are imported and exported to protect confidentiality and integrity in the Key Basics SFP in FDP_IFC.1/KeyBasics and FDP_IFF.1/KeyBasics, the requirements on the secure channels themselves in FTP_TRP.1.

OT.Backup separates out the requirements for any backup and restore properties that the TOE provides and is addressed directly by the Backup SFP in FDP_ACC.1/Backup and FDP_ACF.1/Backup.

OT.RNG is addressed by the requirement in FCS_RNG.1/DRG.4 and FCS_RNG.1/PTG.2 for a random number generator of an appropriate type, which meets appropriate randomness metrics.

OT.TamperDetect is addressed by the requirement for passive tamper detection in FPT_PHP.1 and the tamper response mechanisms in FPT_PHP.3.

OT.FailureDetect is addressed by the self-test requirements of FPT_TST_EXT.1 and secure failure requirements of FPT_FLS.1.

OT.Audit is addressed in terms of basic creation of audit records by the requirements for audit record generation in FAU_GEN.1 and FAU_GEN.2 and provision of timestamps for use in audit records in FPT_STM.1. Protection of the audit trail is ensured by FAU_STG.2, FMT_MTD.1/AuditLog and FMT_SMF.1. Support for the Administrator role that controls export and deletion of audit records from the TOE is required by FMT_SMR.1.

8.2.2 SFR Dependencies

The dependencies between SFRs are addressed as shown in the table below. Where a dependency is not met in the manner defined in [CC2] then a rationale is provided for why the dependency is unnecessary or else met in some other way.

No.	SFR	Dependency	Dependency satisfied by
	FCS	**Cryptographic Support**	
1.	FCS_CKM.1/AES	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.2/KeyExport FCS_CKM.2/KeyImport FCS_COP.1/AES_* FCS_CKM.4
2.	FCS_CKM.1/TDES	[FCS_CKM.2 Cryptographic key distribution or	FCS_CKM.2/KeyExport

No.	SFR	Dependency	Dependency satisfied by
		FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/TDES_Crypt FCS_CKM.4
3.	FCS_CKM.1/GenSecret	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.2/KeyExport FCS_COP.1/HMAC FCS_CKM.4
4.	FCS_CKM.1/RSA	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.2/KeyExport FCS_CKM.2/KeyImport FCS_COP.1/RSA_* FCS_CKM.4
5.	FCS_CKM.1/ECC	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.2/KeyExport FCS_COP.1/ECDSA FCS_COP.1/EdDSA FCS_COP.1/ECDH FCS_CKM.4
6.	FCS_CKM.1/DH	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.2/KeyExport FCS_COP.1/DH FCS_CKM.4
7.	FCS_CKM.2/KeyExport	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/* FCS_CKM.4
8.	FCS_CKM.2/KeyImport	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/* FCS_CKM.4
9.	FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or	FCS_CKM.1/*

No.	SFR	Dependency	Dependency satisfied by
		FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	
10.	FCS.COP.1/TDES	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/TDES FCS_CKM.4
11.	FCS.COP.1/AES_Crypt	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/AES FCS_CKM.4
12.	FCS.COP.1/AES_MAC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/AES FCS_CKM.4
13.	FCS_COP.1/RSA_Sign	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/RSA FCS_CKM.4
14.	FCS_COP.1/RSA_Crypt	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/RSA FCS_CKM.4

No.	SFR	Dependency	Dependency satisfied by
15.	FCS_COP.1/ECDSA	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/ECC FCS_CKM.4
16.	FCS_COP.1/EdDSA	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/ECC FCS_CKM.4
17.	FCS_COP.1/HMAC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/GenSecret FCS_CKM.4
18.	FCS_COP.1/Hash	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 Cryptographic key generation: not relevant because a hash function does not use any cryptographic key. No key generation can be expected here. FCS_CKM.4 Cryptographic key destruction: not relevant because a hash function does not use any cryptographic key. No key destruction can be expected here.
19.	FCS_COP.1/DH	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1/DH FCS_CKM.4

No.	SFR	Dependency	Dependency satisfied by
		FCS_CKM.4 Cryptographic key destruction	
20.	FCS_COP.1/ECDH	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/ECC FCS_CKM.4
21.	FCS_COP.1/KeyDerivation	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/GenSecret FCS_CKM.4
22.	FCS_RNG.1/PTG.2	No dependencies.	n.a.
23.	FCS_RNG.1/DRG.4	No dependencies.	n.a.
	FIA	**Identification and authentication**	
24.	FIA_UID.1	No dependencies.	n.a.
25.	FIA_UAU.1	FIA_UID.1 Timing of identification	FIA_UID.1
26.	FIA_AFL.1	FIA_UAU.1 Timing of authentication	FIA_UAU.1
	FDP	**User data protection**	
27.	FDP_IFC.1/KeyBasics	FDP_IFF.1 Simple security attributes	FDP_IFF.1/KeyBasic
28.	FDP_IFF.1/KeyBasics	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	FDP_IFC.1/KeyBasics FMT_MSA.3/Keys
29.	FDP_ACC.1/Key_Usage	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/KeyUsage
30.	FDP_ACF.1/KeyUsage	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/KeyUsage FMT_MSA.3/Keys

No.	SFR	Dependency	Dependency satisfied by
31.	FDP_ACC.1/Backup	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Backup
32.	FDP_ACF.1/Backup	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/Backup The dependency on FMT_MSA.3 is not relevant in this case since the attribute used in FDP_ACF.1/Backup is determined by the ability of the user to authenticate as an administrator according to FIA_UAU.1
33.	FDP_SDI.2	No dependencies	n.a.
34.	FDP_RIP.1	No dependencies	n.a.
	TRP	**Trusted path/channels**	
35.	FTP_TRP.1	No dependencies	n.a.
	FPT	**Protection of the TSF**	
36.	FPT_STM.1	No dependencies	n.a.
37.	FPT_TST_EXT.1	No dependencies	n.a.
38.	FPT_PHP.1	No dependencies	n.a.
39.	FPT_PHP.3	No dependencies	n.a.
40.	FPT_FLS.1	No dependencies	n.a.
	FMT	**Security management**	
41.	FMT_SMR.1	FIA_UID.1 Timing of identification.	FIA_UID.1
42.	FMT_SMF.1	No dependencies	n.a.
43.	FMT_MTD.1/AuditLog	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
44.	FMT_MTD.1/SWUpdate	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1

No.	SFR	Dependency	Dependency satisfied by
45.	FMT_MSA.1/Keys	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1/Key_Usage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
46.	FMT_MSA.3/Keys	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1/Keys FMT_SMR.1
	FAU	**Security audit data generation**	
47.	FAU_GEN.1	FPT_STM.1 Reliable time stamps	FPT_STM.1
48.	FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification	FAU_GEN.1 FIA_UID.1
49.	FAU_STG.2	FAU_GEN.1 Audit data generation	FAU_GEN.1

Table 13: SFR Dependencies Rationale

8.3 Rationale for SARs

The assurance level for this Security Target is **EAL4 augmented with AVA_VAN.5 and ALC_FLR.3**.
EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialised processes and practices. It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions.
The TOE u.trust Anchor described in this Security Target is just such a product.
Augmentation results from

- the selection of AVA_VAN.5 (Advanced methodical vulnerability analysis). All the dependencies of AVA_VAN.5 are satisfied by other assurance components in the EAL4 assurance package.
- the selection of ALC_FLR.3 (Systematic flaw remediation). ALC_FLR.3 has no dependencies.

8.3.1 AVA_VAN.5 Advanced Methodical Vulnerability Analysis

The TOE generates, uses and manages the highly sensitive data in the form of secret keys, at least some of which may be used as signature creation data. The protection of these keys and associated security of their attributes and use in cryptographic operations can only be ensured by the TOE itself. While the TOE environment is intended to protect against physical

attacks, a high level of protection against logical attacks (especially those that might be carried out remotely) is also necessary, and is therefore addressed by augmenting vulnerability analysis to deal with High attack potential.

8.3.2 ALC_FLR.3 Systematic flaw remediation

The augmentation with ALC_FLR.3 provides the assurance that the developer, Utimaco, has well-defined and appropriate policies and processes in place to react proactively upon security flaws and vulnerabilities found in the field and fixing them, including a procedure for timely response and the automatic distribution of security flaw reports and the associated corrections to registered TOE users who might be affected by the security flaw.

9 TOE Summary Specification

This chapter describes how the TOE will realise the SFRs which are defined in chapter 7.2. For that purpose, the TOE Security Functionality (TSF) will be described by means of a set of security functions (SF.XXX) implemented by the TOE. This detailed description and analysis of the TSF demonstrates how the defined security functions of the TOE work together and support each other. Furthermore, it shows that no inconsistencies exist. Each SFR is implemented by at least one security function. For all SFRs an explanation is given, why and how the defined security functions of the TOE meet the respective SFRs. The given mapping of the SFRs and the security functions of the TOE at the end of this chapter should be considered as an overview and a guidance.

9.1 SF.AUTH: Authentication and Authorisation

The use of any of the security-relevant services of the TOE is not possible without user authentication. Only if a defined authentication status has been obtained then the TOE services can be realised; here the necessary user authentication status depends from the individual service. Command authentication can only be done by subjects (so-called *users*) which have to be registered at the TOE before.

At registration, together with the user's name (Identity), his permission (Role), authentication mechanism, the reference authentication data (RAD: public key or password, depending on the authentication mechanism), the optional key group attribute (to allow access to specific user keys belonging to this key group; this attribute is optional and relevant only for users in role Key Manager, SO or Key User), and further attributes will be stored. Only the RAD may be changed later, all other user attributes cannot be changed. The command for change of a user's RAD has to be authenticated by the user himself. The user's permission decides which of the security-relevant services may be performed by this user (i. e. which user role the user may assume). The step immediately preceding the user authentication is the identification of a user. Therefore, the authentication procedure for the user fulfils directly the SFRs FIA_UID.1 (Timing of identification) and FIA_UAU.1 (Timing of authentication).

The TOE supports the following roles for the different users, thus implementing FMT_SMR.1 (Security roles):

- different administrator roles
    - *Global Administrator* (global device initialisation, device administration and cHSM management)
    - *User Administrator* (role on cHSM level: cHSM user management tasks like creation and deletion of cHSM users)
    - *Container Administrator* (role on cHSM level: general cHSM administration like management of the cHSM's Master Backup Key)
    - *Key Manager* (role on cHSM level: key management tasks for user keys on the cHSM level, like key generation, key export and import, key backup and key restore, key deletion)
    - *SO (Security Officer)* (role on cHSM level: creating, modifying or deleting key group specific configuration objects and initiating a key group)
- *Key User* (role on cHSM level: using a cHSM for cryptographic services like signature creation)
- *Client Application* (that uses the u.trust Anchor for creating a secure channel for communication; hence, each authenticated user can in addition assume the role Client Application)

For using a secret or private key, authentication of a user in role Key User is required, and key management access to a secret or private key (e. g. export or import of the key, or setting or changing key attributes) requires authentication of a user in role Key Manager. In addition to that role-based access constraints, a key can optionally be assigned to a specific key group (by setting a specific key group attribute). In this case the access to the key is further restricted to only users with appropriate role and with the same key group being set as their key group user attribute. This implements in particular FDP_ACF.1.2/KeyUsage (Security attribute based access control), (1) and (2), and FDP_ACC.1/KeyUsage (Subset access control), and FDP_IFF.1.5/KeyBasics (3) (Simple security attributes).

At registration, for every user a dedicated authentication mechanism has to be chosen. The TOE provides three different authentication mechanisms:

- **RSA Signature authentication mechanism**: The authentication is performed with an RSA signature, compliant with FCS_COP.1/RSA_Sign (RSA signature scheme according to the standard [PKCS#1], chapter 8.2 or 8.1).
- **ECDSA Signature authentication mechanism**: The authentication is performed with an ECDSA signature, compliant with FCS_COP.1/ECDSA.
- **HMAC Password authentication mechanism** (only available for users with role on cHSM level): The authentication is performed with an HMAC, using the user's authentication password as the HMAC key, and in compliance with FCS_COP.1/HMAC.

All used hashing algorithms are compliant to FCS_COP.1/Hash.

If any user made an authentication attempt that failed, the next authentication attempt of this user will only be accepted if 4 seconds (or more) have elapsed since the failed attempt, which supports FIA_AFL.1 (Authentication failure handling).

For exchanging sensitive data, a Secure Messaging session (trusted channel) has to be set up between the TOE and the client application. Such a Secure Messaging session is mandatory for each command which requires user authentication. This mechanism enforces trusted communication mechanism for (local or remote) client applications, directly fulfilling FTP_TRP.1 (Trusted Path) and FDP_IFF.1.2/KeyBasics clauses (1), (2) and (3).

SF.CRYPTO supports the user authentication and secure messaging with RSA and ECDSA signature generation and verification, hash value calculation, key derivation, HMAC calculation, EC Diffie-Hellman key agreement, AES encryption and decryption, MAC-calculation and verification and random number generation by hybrid RNG for the challenge value and ephemeral ECDH keys.

9.2 SF.ADMIN: Administration

Security-relevant administration of the TOE cannot be done without user authentication: Only if a defined authentication status has been obtained then administration tasks can be executed. The security function SF.ADMIN providing capabilities to administrate the TOE is therefore related to SF.AUTH.

SF.ADMIN provides the following administrative services, in accordance with FMT_SMF.1, FDP_ACC.1.1/KeyUsage and FMT_SMR.1:

- Backup and restore of entire cHSM including all its data (to be authenticated by a user in Global Administrator role), or of single user keys including all their attributes (to be authenticated by a user in Key Manager role) in accordance with the SFRs FDP_ACC.1/Backup and FDP_ACF.1/Backup
- Creation and deletion of a cHSM (to be authenticated by a user in Global Administrator role)
- Start and stop a running cHSM (to be authenticated by a user in Global Administrator role)
- Export and import of keys by authorised subjects in accordance with FDP_IFC.1/KeyBasics and FDP_IFF.1/KeyBasics (to be authenticated by a user in Key Manager role)
- Modifications of key attributes by authorised subjects in accordance with FDP_ACF.1/KeyUsage (to be authenticated by a user in Key Manager role)
- System time setting to support FPT_STM.1 (to be authenticated by a user in Global Administrator role)
- Export and deletion of the audit log records in accordance with FMT_MTD.1/AuditLog
- Software update in accordance with the SFR FMT_MTD.1/SWUpdate (to be authenticated by a user in Global Administrator role).

For the user management typical functions are available. Basically, the service deals with administration of the user database (creation, deletion, changing). The commands for creation or deletion of a user have to be authenticated by a user in User Administrator role (for users on cHSM level) respectively Global Administrator role (for users on global level). The command for changing the user's authentication token (password or public key) has to be authenticated by the respective user himself.

9.3 SF.KEY_MAN: Key Management

The security function SF.KEY_MAN provides key management, including internal as well as external key storage for the user keys. Key management cannot be done without user authentication: Only if a defined authentication status has been obtained then key management tasks can be executed. The key management security function SF.KEY_MAN is therefore closely related to SF.AUTH.

SF.KEY_MAN provides the following services by means of SF.CRYPTO fulfilling parts of FDP_ACC.1/KeyUsage, FDP_ACF.1/KeyUsage, FMT_SMR.1 and FMT_SMF.1:

- Generation and export of the Master Backup Key of a cHSM, in accordance with the SFR FCS_CKM.1/AES and FDP_IFF.1/KeyBasics (authenticated by a cHSM Administrator)
- Import of the Master Backup Key of a cHSM, protected by encryption in line with FCS_CKM.2/KeyImport (authenticated by a cHSM Administrator)
- Import of an Operator Base Secret, protected by encryption in line with FCS_CKM.2/KeyImport (authenticated by a Global Administrator)
- Backup and restore of user keys on cHSM level (as required by FMT_SMF.1.1 (4)), authenticated by a Key Manager, protected with the Master Backup Key in order to fulfill FDP_IFF.1.5/KeyBasics (1) and FDP_ACF.1.2/Backup (2) and (3)

- Generation of user keys on cHSM level (authenticated by a Key Manager):
  - AES keys in accordance with FCS_CKM.1/AES
  - TDES keys in accordance with FCS_CKM.1/TDES
  - Generic secret keys e. g. for HMAC algorithm in accordance with FCS_CKM.1/GenSecret
  - Elliptic curve keys in accordance with FCS_CKM.1/ECC
  - DH keys in accordance with FCS_CKM.1/DH
  - RSA keys in accordance with the SFR FCS_CKM.1/RSA
- Deletion of keys (authenticated by a Key Manager) in accordance with the SFR FCS_CKM.4
- Modification of key attributes (authenticated by a Key Manager) as required by FMT_SMF.1.1 (2)
- Import and export of keys as required by FMT_SMF.1.1 (5) and (6) (authenticated by a Key Manager):
  - Import of keys in accordance with the rules in FDP_IFF.1.2/KeyBasics (3) and (4), and fulfilling FCS_CKM.2/KeyImport: The TSF allow only encrypted key import (if the key is secret or private), and if the key has an attribute “key group” set the command requires authentication of a Key Manager that is assigned to the same key group (per user attribute “key group”).
  - Export of keys in accordance with the rules in FDP_IFF.1.2/KeyBasics (1) and (2), FDP_IFF.1.5/KeyBasics, and fulfilling FCS_CKM.2/KeyExport: The TSF allow only encrypted key export (if the key is secret or private), keys that have the appropriate “exportable” key attribute not set cannot be exported, and if the key has an attribute “key group” set the command requires authentication of a Key Manager that is assigned to the same key group (per user attribute “key group”).

FDP_ACF.1/KeyUsage (Security attribute based access control) enforces the Key Usage SFP to authenticated users who are currently authorised to change attributes of secret key (see also SF.AUTH and SF.ADMIN).

Management of security attributes of keys is performed in accordance with FMT_MSA.1/Keys (Management of security attributes) and FMT.MSA.3/Keys (Static attribute initialisation).

9.4 SF.CRYPTO: Cryptographic Support

SF.CRYPTO provides cryptographic support for the other TSFs using cryptographic mechanisms, and it enables cryptographic services like signature generation and verification for the user of the TOE.

SF.CRYPTO supports the following cryptographic operations:

- AES block cipher in various modes (ECB, CBC, OFB, CTR, CCM, GCM, KW, KWP) with a key length of 16, 24 or 32 bytes used for encryption or decryption in accordance with the SFR FCS_COP.1/AES_Crypt
- AES CMAC and GMAC generation and verification with a key length of 16, 24 or 32 bytes in accordance with the SFR FCS_COP.1/AES_MAC
- TDES block cipher in ECB and CBC mode with a key length of 192 bits used for encryption or decryption in accordance with the SFR FCS_COP.1/TDES_Crypt

- ECDSA algorithm according to the standard [ANSI-X9.62] with keys based on various ECC domain parameters and key lengths of minimum 224 bit used for ECDSA signature generation or verification in accordance with the SFR FCS_COP.1/ECDSA
- EdDSA algorithm according to the standard [RFC 8032] with keys based on ECC curve edwards25519 used for EdDSA signature generation or verification in accordance with the SFR FCS_COP.1/EdDSA
- RSA encryption scheme according to the standard [PKCS#1] with key lengths of minimum 2048 and maximum 16,384 bit modulus lengths used for RSA encryption or decryption in accordance with the SFR FCS_COP.1/RSA_Crypt
- RSA signature scheme according to the standard [PKCS#1] with key lengths of minimum 2048 and maximum 16,384 bit modulus lengths used for RSA signature generation and verification in accordance with the SFR FCS_COP.1/RSA_Sign
- HMAC calculation in accordance with the SFR FCS_COP.1/HMAC (HMAC key size shorter than 13 bytes for internal use only to support user authentication, key size 13 bytes and more also provided as cryptographic service)
- Hash algorithms SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384 and SHA3-512 in accordance with the SFR FCS_COP.1/Hash
- Diffie-Hellmann (DH) shared secret agreement in accordance with the SFR FCS_COP.1/DH
- EC Diffie-Hellmann (ECDH) shared secret agreement in accordance with the SFR FCS_COP.1/ECDH
- Key Derivation in accordance with the SFR FCS_COP.1/KeyDerivation (for internal use only to support the implementation of the trusted channel and the secure backup of keys)
- Random number generation by a hybrid RNG in accordance with the SFR FCS_RNG.1/DRG.4, seeded and re-seeded by a physical RNG in accordance with the SFR FCS_RNG.1/PTG.2.

9.5 SF.REL: Reliability

The TOE's security function to provide reliability of the TSF, SF.REL, monitors the following events:

- Self-test error
- Stored data integrity failure
- Failure of user authentication attempts
- Results of services of SF.ADMIN, SF.KEY_MAN and SF.SWUPDATE (if TSF is configured accordingly)

and provides the corresponding audit records in accordance with the SFRs FAU_GEN.1 (Audit data generation), FAU_GEN.2 (User identity association), FPT_STM.1 (Reliable time stamps) and FAU_STG.2 (Guarantees of audit data availability).

SF.REL provides services to query the audit records in accordance with FMT_MTD.1/AuditLog (Management of TSF data), see the description of SF.ADMIN. The TOE does not provide any possibility to modify the audit records except for (entire or partial) clearance, whereby the service for the clearance of the audit data has to be authenticated by a user in Global Administrator role (for audit data on global level), or in cHSM Administrator or User Administrator role (for audit data on cHSM level), in accordance with the SFRs FAU_STG.2 (Guarantees of audit data availability) and FMT_MTD.1/AuditLog (Management of TSF data).

The TOE hardware is a cryptographic module in the form of a physically protected PCI Express (PCIe) plug-in card. SF.REL preserves a secure operation state of the TOE when the following types of failures and attacks occur:

- Power supply too high/too low
- Temperature too high/too low
- Defect of any of the tamper wires
- Integrity check of cryptographic keys and stored firmware fails
- Self-test fails

The TOE provides an alarm mechanism which detects if the physical environmental conditions are outside of the normal operating range, or if a tamper attack might have occurred, and reacts by destroying all sensitive data. For this mechanism a sensory is implemented which watches temperature, voltage, and the intactness of the tamper wires.

Furthermore, the TOE with its tamper-evident enclosure implements the following physical security mechanisms against direct physical attacks:

- The cryptographic module's hardware components are covered by hard, opaque potting material and/or the heat sink, which show evidence of tampering on the enclosure when a physical attack is attempted. This provides the capability to determine physical tampering according to FPT_PHP.1 (Passive detection of physical attack).
- The potting material is hard and opaque enough to prevent direct observation and easy penetration to the depth of the underlying hardware components.

The tamper response and zeroisation circuitry is also active while the module is in standby mode (powered down).

The implemented sensory and software part of the TOE react properly to all security relevant events being generated by the hardware in response to any physical attack attempts. The resistance of the TOE hardware and sensory to physical and chemical attacks is successfully evaluated according to the requirements of FIPS 140-2 standard [FIPS 140-2], level 3. This is equivalent to the physical security requirements as laid down in [ISO/IEC 19790:2012] for Security Level 3, sections 7.7.2 and 7.7.3. Therefore, the security function SF.REL supplies effective hardware and software based mechanisms satisfying the SFR FPT_PHP.3 (Resistance to physical attack).

Due to the implemented alarm mechanism the TOE preserves a secure state also if the power supply or temperature is outside of a well-defined operational range or any of the tamper wires are disrupted: If extreme power levels occur to the TOE or if extreme temperature is monitored or if any of the tamper wires is disrupted, an alarm is triggered, all data is deleted and the TOE will be reset cleanly according to FPT_FLS.1 (Failure with preservation of secure state). The security function SF.REL realises effective hardware and software based features to preserve a secure operational state of the TOE in case of induced hardware or software failures or tampering. It satisfies directly the SFR FPT_FLS.1.

For the protection of data and firmware integrity the security function SF.REL implements various measures.

During the boot process after power-on or reset the TOE's boot chain performs further self-tests, including a temperature test and a self-test of the digitized noise data of the PTRNG which is used to seed and re-seed the DRNG.

Furthermore, the global firmware and each cHSM perform extensive cryptographic power-on self-tests in accordance with FPT_TST_EXT.1 (Basic TSF self-testing), including Known Answer Tests and Pair-wise Consistency Test for all algorithms listed in FCS_COP.1 iterations. It is only possible to execute any cryptographic or other security-relevant service after these power-on self-tests have been completed successfully. If these self-tests for the

global firmware pass but one of the power-on self-tests of a cHSM fails, this specific cHSM enters a secure Error State in which none of its cryptographic services but only status requests are available.

The TOE performs furthermore self-tests at specific conditions in accordance with FPT_TST_EXT.1 (Basic TSF self-testing), including Online Test of the digitised noise data of the physical RNG of FCS_COP.1/PTG.2, according to [AIS 20/31] for RNG class PTG.2, and continuous health tests according to [SP 800-90B] chapters 4.4.1 and 4.4.2, and Firmware Load Test (via ECDSA signature verification) for every Operational Image when being loaded, see SF.SWUPDATE.

If one of these conditional self-tests fails, the requested action is not performed (e. g. firmware image to be loaded is not loaded, generated key is not stored etc.), and the command is aborted with an error code. The successful completion of all self-tests or the secure Error State is indicated by the “Get State” command.

Secret or private keys are deleted in accordance with the SFR FCS_CKM.4 (Cryptographic key destruction). SF.REL ensures that any previous information content is not available after deletion.

SF.REL monitors stored data and prohibits usage of altered data and notifies the user if integrity errors are detected in accordance to FDP_SDI.2. This holds for internally stored keys as well as for externally stored keys which are integrity protected with an AES CMAC according to FCS_COP.1/AES_MAC.

The mechanism used for fulfilling FCS_CKM.4 for key destruction, namely overwriting the key by zeroising in case of secret or private keys, applies to all secret and private keys and data, and therefore also ensures that any previous information content of a resource is made unavailable upon the de-allocation of secret authentication data and secret keys, which is in accordance to FDP_RIP.1.

9.6 SF.SWUPDATE: Software Update

SF.SWUPDATE allows to perform a secure software update on the TOE by providing the “System Update” service. This service has to be authenticated by a user with the Global Administrator role, in accordance with FMT_MTD.1/SWUpdate.

The “System Update” service allows the download of an Operational Image which contains a signature calculated over the whole image. The signature is calculated with a dedicated Operational Image Signing Key, OISK, owned by the manufacturer, Utimaco (ECDSA signature in line with FCS_COP.1/ECDSA), which is held in the manufacturer’s secure production environment. If the signature cannot be verified, the download is prohibited and the “System Update” service will return an error code instead.

On power-up, the device performs firmware integrity checks during the whole boot chain. If any of the firmware integrity checks during boot of the Operational Image fails *before* any cHSM has been started, the device may try to boot the Backup Operational Image instead and including the integrity checks on the Backup Operational Image firmware during boot. The Backup Operational Image provides the same interface like the Operational Image and in particular allows to execute another “System Update” command.

If any integrity check of cHSM firmware fails when starting or re-starting one of the cHSMs, the failing cHSM will be set to a secure Error State. In this Error State only status request commands but no cryptographic services are available to any user of this cHSM. But this affects only the failed cHSM: other cHSMs may be started successfully.

9.7 Coverage of SFRs by Security Functions

The following table shows that all TOE Security Functional Requirements (SFRs) are realised by the TSF (TOE Security Functionality) described in terms of security functions (SF.XXX).

SFR	**SF.AUTH**	**SF.ADMIN**	**SF.KEY_MAN**	**SF.CRYPTO**	**SF.REL**	**SF.SWUPDATE**
FCS_CKM.1/AES (Cryptographic key generation)			**X**			
FCS_CKM.1/TDES (Cryptographic key generation)			**X**			
FCS_CKM.1/GenSecret (Cryptographic key generation)			**X**			
FCS_CKM.1/RSA (Cryptographic key generation)			**X**			
FCS_CKM.1/ECC (Cryptographic key generation)			**X**			
FCS_CKM.1/DH (Cryptographic key generation)			**X**			
FCS_CKM.2/KeyExport (Cryptographic key distribution)			**X**			
FCS_CKM.2/KeyImport (Cryptographic key distribution)			**X**			
FCS_CKM.4 (Cryptographic key destruction)			**X**		**X**	
FCS.COP.1/TDES (Cryptographic operation)				**X**		
FCS.COP.1/AES_Crypt (Cryptographic operation)				**X**		
FCS.COP.1/AES_MAC (Cryptographic operation)				**X**		

SFR	SF.AUTH	SF.ADMIN	SF.KEY_MAN	SF.CRYPTO	SF.REL	SF.SWUPDATE
FCS_COP.1/RSA_Sign (Cryptographic operation)	x			x		
FCS_COP.1/RSA_Crypt (Cryptographic operation)				x		
FCS_COP.1/ECDSA (Cryptographic operation)	x			x		x
FCS_COP.1/EdDSA (Cryptographic operation)				x		
FCS_COP.1/HMAC (Cryptographic operation)	x			x		
FCS_COP.1/Hash (Cryptographic operation)				x		
FCS_COP.1/DH (Cryptographic operation)				x		
FCS_COP.1/ECDH (Cryptographic operation)				x		
FCS_COP.1/KeyDerivation (Cryptographic operation)				x		
FCS_RNG.1/PTG.2 (Generation of random numbers)				x		
FCS_RNG.1/DRG.4 (Generation of random numbers)				x		
FIA_UID.1 (Timing of identification)	x					
FIA_UAU.1 (Timing of Authentication)	x					
FIA_AFL.1 (Authentication failure handling)	x					
FDP_IFC.1/KeyBasics (Subset Information Control)		x				

SFR	**SF.AUTH**	**SF.ADMIN**	**SF.KEY_MAN**	**SF.CRYPTO**	**SF.REL**	**SF.SWUPDATE**
FDP_IFF.1/KeyBasics (Simple security attributes)		**x**	**x**			
FDP_ACC.1/KeyUsage (Subset access control)	**x**	**x**	**x**			
FDP_ACF.1/KeyUsage (Security attribute based access control)	**x**	**x**	**x**			
FDP_ACC.1/Backup (Subset access control)		**x**				
FDP_ACF.1/Backup (Security attribute based access control)		**x**	**x**			
FDP_SDI.2 (Stored data integrity monitoring and action)					**x**	
FDP_RIP.1 (Subset residual information protection)					**x**	
FTP_TRP.1 (Trusted path)	**x**					
FPT_STM.1 (Reliable time stamps)		**x**			**x**	
FPT_TST_EXT.1 (Basic TSF self testing)					**x**	
FPT_PHP.1 (Passive detection of physical attack)					**x**	
FPT_PHP.3 (Resistance to physical attack)					**x**	
FPT_FLS.1 (Failure with preservation of secure state)					**x**	
FMT_SMR.1 (Security roles)	**x**	**x**	**x**			
FMT_SMF.1 (Security management functions)		**x**	**x**			

SFR	**SF.AUTH**	**SF.ADMIN**	**SF.KEY_MAN**	**SF.CRYPTO**	**SF.REL**	**SF.SWUPDATE**
FMT_MTD.1/AuditLog (Management of TSF Data)		x			x	
FMT_MTD.1/SWUpdate (Management of TSF Data)		x				x
FMT_MSA.1/Keys (Management of security attributes)			x			
FMT_MSA.3/Keys (Static attribute initialisation)			x			
FAU_GEN.1 (Audit data generation)					x	
FAU_GEN.2 (User identity association)					x	
FAU_STG.2 (Guarantees of audit data availability)					x	

Table 14: Mapping SFRs to Security Functions

10 Annex

This Annex contains the following sections:

Glossary and Acronyms

References

10.1 Glossary and Acronyms

The following glossary includes all used terms of this Security Target regarding to the Common Criteria and IT technology terms in alphabetical order.

Term	Description
Administrator	An authenticated user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given to them.
Authentication keys	General term for keys used for authentication of data (i.e. Data authentication keys) or the identity of an entity (i.e. Entity authentication keys)
Confidentiality	The property that sensitive information is not disclosed to unauthenticated individuals, entities, or processes
Cryptographic algorithm	A well-defined computational procedure that takes variable inputs that usually includes a cryptographic key and produces an output, e. g. encryption, decryption, a private or a public operation in a dynamic authentication, signature creation, signature verification, generation of hash value.
Cryptographic boundary	An explicitly defined continuous perimeter that establishes the physical bounds of a cryptographic module and contains all the hardware, software, and/or firmware components of a cryptographic module.
Cryptographic checksum	A checksum that is created by performing a cryptographic algorithm. The cryptographic checksum can be associated with the original data in order to provide a mechanism to verify that the original data has not been changed.
Cryptographic functions	TSF implementing cryptographic algorithms and/or protocols for • encryption and decryption, • signature creation or verification, • calculation of Message Authentication Code, • authentication

Term	Description
Cryptographic key (key)	A parameter used in conjunction with a cryptographic algorithm that determines • the transformation of plaintext data into ciphertext data, • the transformation of ciphertext data into plaintext data, • a digital signature computed from data, • the verification of a digital signature computed from data, • a Message Authentication Code computed from data, • a proof of the knowledge of a secret, • a verification of the knowledge of a secret or • an exchange agreement of a shared secret.
Cryptographic key component (key component)	A parameter used in conjunction with other key components in an Endorsed security function to form a plaintext cryptographic key by a secret sharing algorithm (e.g. the cryptographic plaintext key is the XOR-sum of two key components)
Cryptographic module	The set of hardware, software and/or firmware that implements Endorsed security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.
Cryptographic protocol	A cryptographic algorithm including interaction with an external entity (e.g. key exchange)
Data path	The physical or logical route over which data passes; a physical data path may be shared by multiple logical data paths.
Decryption algorithm	Algorithm of decoding a cipher text into the plaintext using a decryption key. The decryption algorithm reproduces the plaintext that is used to calculate the cipher text with the corresponding encryption algorithm and the corresponding encryption key.
Destruction of data	A method of erasing electronically stored data, e. g. cryptographic keys, by altering or deleting the contents of the data storage to prevent recovery of the data.
Digital signature	The result of an asymmetric cryptographic transformation of data which, when properly implemented, provides the services of 1. Origin authentication, 2. Data integrity, and 3. Signer non-repudiation.
Encrypted key	A cryptographic key that has been encrypted using an Endorsed security function with a key encrypting key, a PIN, or a password in order to disguise the value of the underlying plaintext key.

Term	Description
Encryption algorithm	Algorithm of processing a plaintext into a cipher text using an encryption key in a way that decoding of the cipher text into the plain text without knowledge of the corresponding decryption key is computationally infeasible.
Endorsed	For this security target, endorsed by the certification body for the evaluation of products of an intended type and resistance against attacks with attack potential addressed by the vulnerability analysis component in the security target[164].
Endorsed security function	For this security target, a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either a) specified in an Endorsed standard, b) adopted in an Endorsed standard and specified either in an appendix of the Endorsed standard or in a document referenced by the Endorsed standard, or c) specified in the list of Endorsed security functions.
Error detection code (EDC)	A code computed from data and comprised of redundant bits of information designed to detect, but not correct, unintentional changes in the data.
Error mode	Mode of operation when the cryptographic module has encountered an error condition as defined in FPT_FLS.1.
Error state	State related to the Error mode
Firmware	The programs and data components of a cryptographic module that are stored in hardware (e.g., ROM, PROM, EPROM, EEPROM or FLASH) and cannot be dynamically written or modified during execution.
Hardware	The physical equipment used to process programs and data.
Hash-based message authentication code (HMAC)	A message authentication code that utilises a keyed hash.
Information processing	The organisation, manipulation and distribution of information.
Initialisation vector (IV)	A vector used in defining the starting point of an encryption process within a cryptographic algorithm.

[164] Endorsed algorithms and functions could be similar to the list of cryptographic algorithms and parameters published for qualified electronic signatures by the notified body Bundesnetzagentur in Germany, the agreed cryptographic mechanisms from [SOG-IS-Crypto], or the Approved algorithms published by NIST in the USA.

Term	Description
Input data	Information that is entered into a cryptographic module for the purposes of transformation or computation using an Endorsed security function.
Integrity	The property that sensitive data has not been modified or deleted in an unauthorised and undetected manner.
Internal secrets	Confidential data inside the cryptographic boundary not intended for export (e.g. secret or private plaintext keys, authentication reference data).
Key establishment	The process by which cryptographic keys are securely distributed among cryptographic modules using manual transport methods (e.g., key loaders), automated methods (e.g., key transport and/or key agreement protocols), or a combination of automated and manual methods (consists of key transport plus key agreement).
Key management	The activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.
Key transport	Secure transport of cryptographic keys from one cryptographic module to another module.
Key usage type	Type of cryptographic algorithm a key can be used for (e.g. AES encryption, RSA signature-creation)
Key User	An individual (subject) that accesses a cryptographic module in order to obtain cryptographic services with a cryptographic key.
Logical external interface	A logical entry or exit point of a cryptographic module that provides access to the module for logical information flows representing physical signals (see also the term “port” for the physical aspects of a logical external interface). In the CC terminology it covers all logical external interfaces of the TOE (direct or indirect interfaces to the TSF or interfaces to the non-TSF portion of the TOE).
Maintenance mode	Mode of operation for maintaining and servicing a cryptographic module, including physical and logical maintenance testing.
Maintenance state	State related to the Maintenance mode .
Message authentication with appendix	A digital signature scheme which requires the message as input to the verification algorithm. The signature is attached to the message.
Microcode	The elementary processor instructions that correspond to an executable program instruction.

Term	Description
Operating conditions	Any environmental condition being accidental or induced outside of the normal range intended for the TOE may affect the correct operation or compromise of confidential information. These conditions include but are not limit to voltage of power supply, temperature, emanation which TOE environmental conditions.
Output data	Data containing information that is produced from a cryptographic module.
Password	A string of characters (letters, numbers, and other symbols) used to authenticate an identity.
Permanent stored keys	Keys remains stored in the TOE after power off or reset.
Physical protection	The safeguarding of a cryptographic module, including its cryptographic keys and other critical security parameter, using physical means.
Plaintext key	An unencrypted cryptographic key.
Port	A physical input or output interface of a cryptographic module that provides access to the module for physical signals, represented by logical information flows. Physically separated ports do not share the same physical pin or wire. In the CC terminology a port is a physical external interface of the TOE (direct or indirect interface to the TSF or interface to the non-TSF portion of the TOE).
Power interface/port	Interface respective port providing all external electrical power supply.
Power On/Off mode	Mode of operation that indicates whether the cryptographic module is supplied by a power source. These modes may distinguish between different power sources (e.g., primary, secondary, backup power source or none) being applied to a cryptographic module.
Power On/Off state	State related to the Power On/Off mode (cf. ADV_ARC.1).
Private key	A cryptographic key, used with a public key cryptographic algorithm, that is uniquely associated with an entity and is not made public.
Protection Profile	An implementation-independent set of security requirements for a category of Targets of Evaluation (TOEs) that meet specific consumer needs.
Public key	A cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that may be made public.

Term	Description
Public key (asymmetric) cryptographic algorithm	A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible.
Public key certificate	A set of data that uniquely identifies an entity, contains the entity's public key, and is digitally signed by a trusted party, thereby binding the public key to the entity.
Random Number Generator	Random Number Generators (RNGs) used for cryptographic applications produce a sequence of zero and one bits that may be combined into sub-sequences or blocks of random numbers. There are three basic classes physical true RNG, non-physical true RNG, and deterministic RNG. A physical true RNG produces output that dependents on some physical random source inside the TOE boundary only. A non-deterministic true RNG gets its entropy from sources from outside the TOE boundary (e.g. by system data like RAM data or system time of a PC, output of API functions etc., or human interaction like key strokes, mouse movement etc.). A deterministic RNG consists of an algorithm that produces a sequence of bits from an initial random value (seed).
Reference authentication data	Data known for the claimed identity and used by the TOE to verify the verification authentication data provided by an entity in an authentication attempt to prove their identity.
Reset	Action to clear any pending errors or events and to bring a system to normal condition or initial state (e.g. after power-on).
Secret key	A cryptographic key, used with a secret key cryptographic algorithm that is uniquely associated with one or more entities and should not be made public.
Secret key (symmetric) cryptographic algorithm	A cryptographic algorithm the keys of which for both encryption and decryption respective MAC calculation and MAC verification are the same or can easily be derived from each other and therefore must be kept secret.
Seed key	A secret value used to initialise a cryptographic function or operation.
Self-test mode	Mode of operation in which the cryptographic module performs initial start-up self-test, self-test at power-on, self-test at the request of the authorised user and may perform other self-tests identified in FPT_TST_EXT.1
Self-test state	State related to the Self-test mode (cf. ADV_ARC.1).

Term	Description
Shutdown	Shutdown of the TOE initiated by the user (may not include reset after detection of error or power-off due to loss of power supply)
Signature-creation key	Private key for the creation of digital signatures
Signature-verification key	Public key for the verification of digital signatures
Software	The programs and data components, usually stored on erasable media (e.g., disk), that can be dynamically written and modified during execution.
Split knowledge	A process by which a cryptographic key is split into multiple key components, individually sharing no knowledge of the original key, that can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.
Status information	Information that is output from a cryptographic module for the purposes of indicating certain operational characteristics or modes of the module.
Status output interface/port	Interface respective port intended for all input commands, signals, and control data (including calls and manual controls such as switches, buttons, and keyboards) used to control the operation of the cryptographic module).
System software	The special software within the cryptographic boundary (e.g., operating system, compilers or utility programs) designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system, and associated programs, and data.
Tamper detection	The automatic determination by a cryptographic module that an attempt has been made to compromise the physical security of the module.
Target of Evaluation (TOE)	An information technology product or system and associated administrator and user guidance documentation that is the subject of an evaluation.
Timing analysis	Analysis of timing behaviour of a device, equipment, or system to gain information about its internal secrets or processes
TOE Security Functionality (TSF)	Combined functionality of all hardware, software, and firmware of a TOE that must be relied upon for the correct enforcement of the SFRs.

Term	Description
TOE security functions interface (TSFI)	A set of interfaces, whether interactive (man-machine interface) or machine (machine-machine interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.
Trusted channel	A means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSF.
Trusted path	A means by which a user and a TSF can communicate with necessary confidence to support the TSF.
Unauthenticated User	An identified user not being authenticated and having rights as identified in the component FIA_UAU.1.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE (includes both authenticated and unauthenticated entities).

Table 15: Glossary

The following table includes all used acronyms of this Security Target regarding to the Common Criteria and IT technology terms in alphabetical order.

Acronym	Term
Common Criteria and general	
CC	Common Criteria
DTBS	Data To Be Signed
DTBS/R	Data to be signed or its unique representation
MBK	*Master Backup Key*
n. a.	Not applicable
RAD	Reference authentication data
SAR	Security assurance requirement
SFR	Security functional requirement
TOE	Target of Evaluation
TSF	TOE security functionality
TSP	Trusted Service Provider

Acronym	Term
Cryptographic Algorithms	
AES	The *Advanced Encryption Standard* (AES) is a symmetric cryptographic algorithm specified for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.
ECC	*Elliptic Curve Cryptography*
ECDSA	The *Elliptic Curve Digital Signature Algorithm (ECDSA)* is a variant of the asymmetric cryptographic algorithm *Digital Signature Algorithm (DSA)* which uses elliptic curve cryptography. The *DSA* was developed by the United States government for digital signatures. It can be used only for signing data and it cannot be used for encryption.
RSA	*RSA* stands for *Rivest*, *Shamir* and *Adleman*. *RSA* is an asymmetric cryptographic algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem.
SHA	The term *Secure Hash Algorithm (SHA)* denotes a group of standardised cryptographic hash functions used for calculation of a unique check value (digital signature) for arbitrary digital data.
IT technology terms	
LAN	Local Area Network
PCI	Peripheral Component Interconnect
PCIe	PCI express
PIN	Personal Identification Number

Table 16: Acronyms

10.2 References

[AIS 20/31] Application Notes and Interpretation of the Scheme (AIS): AIS 20/AIS 31: A proposal for: Functionality classes for random number generators, Version 2.0 / Wolfgang Killmann (T-Systems GEI GmbH, Bonn), Werner Schindler (Bundesamt für Sicherheit in der Informationstechnik/BSI, Bonn), 18. September 2011

[ANSI-X9.31] ANS X9.31-1998: Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA) / ANSI (American National Standards Institute)

[ANSI-X9.42] ANS X9.42-2003 (R2013): Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography / ANSI (American National Standards Institute)

[ANSI-X9.62] ANS X9.62-2005: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA) / ANSI (American National Standards Institute)

[ANSI-X9.63] ANS X9.63-2001: Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport using Elliptic Curve Cryptography / ANSI (American National Standards Institute)

[CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017

[CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017

[CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components; CCMB-2017-04-003, Version 3.1, Revision 5, April 2017

[CEM] Common Methodology for Information Technology Security Evaluation, CCMB-2017-04-004, Version 3.1, Revision 5, April 2017

[FIPS 46-3] FIPS PUB 46-3, Data Encryption Standard (DES) / National Institute of Standards and Technology (NIST), USA, October 1999

[FIPS 140-2] FIPS PUB 140-2, Security Requirements for Cryptographic Modules / National Institute of Standards and Technology (NIST), USA, May 2001

[FIPS 180-4] FIPS PUB 180-4, Secure Hash Standard (SHS) / National Institute of Standards and Technology (NIST), USA, March 2012

[FIPS 186-4] FIPS PUB 186-4, Digital Signature Standard / National Institute of Standards and Technology (NIST), USA, July 2013

[FIPS 197] FIPS PUB 197, Advances Encryption Standard (AES) / National Institute of Standards and Technology (NIST), USA, 26th November 2001

[FIPS 198] FIPS PUB 198, The Keyed-Hash Message Authentication Code (HMAC) / National Institute of Standard and Technology (NIST), USA, 6th March 2002

[FIPS 202] FIPS PUB 202 (Federal Information Processing Standards Publication) – SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions / National Institute of Standards and Technology (NIST), August 2015

[ISO/IEC 19790:2012] ISO/IEC 19790:2012(E): Information Technology — Security Techniques — Security requirements for cryptographic modules / International Organization for Standardization, Geneva, Switzerland, 15th August 2012

[ANSSI] ANSSI: "Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français" in: Journal Officiel de la République Française (JORF),

n° 0241 du 16 octobre 2011 page 17533 text n° 30 (Announcement about elliptic curve parameters set by the French government). NOR: PRMD1123151V. Available: https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000024668816

[SP 800-38A] NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques / National Institute of Standards and Technology (NIST), USA, December 2001

[SP 800-38B] NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication / National Institute of Standards and Technology (NIST), USA, May 2005

[SP 800-38C] NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality / National Institute of Standards and Technology (NIST), May 2004

[SP 800-38D] NIST Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC / National Institute of Standards and Technology (NIST), November 2007

[SP 800-38F] NIST Special Publication 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping / National Institute of Standards and Technology (NIST), December 2012

[SP 800-56A] NIST Special Publication 800-56A, Revision 3: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography / National Institute of Standards and Technology (NIST), April 2018

[SP 800-56C] NIST Special Publication 800-56C, Revision 1: Recommendation for Key Derivation Methods in Key Establishment Schemes / National Institute of Standards and Technology (NIST), April 2018

[SP 800-67] NIST Special Publication 800-67, Revision 2, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher / National Institute of Standards and Technology (NIST), USA, November 2017

[SP 800-90B] NIST Special Publication 800-90B, Recommendation for Entropy Sources Used for Random Bit Generation / National Institute of Standards and Technology (NIST), January 2018

[SP 800-108] NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions (Revised) / National Institute of Standards and Technology (NIST), USA; October 2009

[PKCS#1] PKCS#1: RSA Cryptography Standard v2.2, 27thOctober 2012 / RSA Laboratories, http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm

[RFC 2104] RFC 2104: HMAC: Keyed-Hashing for Message Authentication, Internet Engineering Task Force (IETF), February 1997

[RFC 5639] RFC 5639: Elliptic Curve Cryptography (ECC) Brainpool Standard - Curves and Curve Generation / M. Lochter and J. Merkle, March 2010. https://datatracker.ietf.org/doc/html/rfc5639

[RFC 6954] RFC 6954, Using the Elliptic Curve Cryptography (ECC) Brainpool Curves for the Internet Key Exchange Protocol Version 2 (IKEv2)

[RFC 7748] RFC 7748: Elliptic Curves for Security / Internet Research Task Force (IRTF), January 2016, ISSN 2070-1721, including Errata ID 4730 reported and verified on 2016-07-05

[RFC 8032] RFC 8032: Edwards-Curve Digital Signature Algorithm EdDSA / Internet Research Task Force (IRTF), January 2017, ISSN 2070-1721

[PP_CMTS] EN 419 221-5:2018 Protection Profiles for TSP Cryptographic Modules - Part 5: Cryptographic Module for Trust Services, v1.0, registered under the reference ANSSI-CC-PP-2016/05-M01, 18 May 2020

[Regulation] 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

[SEC1] SEC1: Elliptic Curve Cryptography – Certicom Research – May 21, 2009, Version 2.0

[SEC2] SEC2: Recommended Elliptic Curve Domain Parameters – Certicom Research – September 20, 2000, Version 1.0

[SOG-IS-Crypto] SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, v1.2, January 2020

[Shamir] A. Shamir, How to share a secret, Communications of the ACM, 22 (1979), 612-613

[TR03111] Technical Guideline TR-03111, Elliptic Curve Cryptography, Version 1.11, April 2009 / Bundesamt für Sicherheit in der Informationstechnik (BSI)

[TS 119 312] ETSI TS 119 312 Electronic Signatures and Infrastructures (ESI); Cryptographic Suites, V1.1.1 (2014-11)

[IEEE 1363a] 1363a-2004 - IEEE Standard Specifications for Public-Key Cryptography – Amendment 1: Additional Techniques, 2004-03-25