Security Target

fiskaly Cloud Crypto Service Provider

TOE Version 1.0.4

Document Version 1.1.7

February 4, 2021

fiskaly GmbH

Contents

Chapter 1

Introduction

This document is the Security Target for the Common Criteria evaluation of the *fiskaly Cloud Crypto Service Provider*.

1.1 ST and TOE Reference

Document Type:	Security Target
Document Version:	1.1.7
Document built from commit:	973803516ca56eba3b4b699a1f07317f4ae7e006
Date:	February 4, 2021
Author:	fiskaly GmbH
Certification-ID:	BSI-DSZ-CC-1153
TOE Identification:	fiskaly Cloud Crypto Service Provider
TOE Version:	1.0.4
CC Version:	3.1 Revision 5
Assurance Level:	EAL2 augmented with ALC_LCD.1 and ALC_CMS.3
PP Conformance:	BSI-CC-PP-0111-2019, BSI-CC-PP-0112-2020, BSI-CC-PP-0113-2020

1.2 TOE Overview

Usage and Major Security Features

The *fiskaly Crypto Service Provider* is a software TOE. Its main purpose is providing cryptographic operations and security services for security module applications such as the *Security Module Application for Electronic Record-Keeping Systems (SMAERS)* defined by BSI-CC-PP-0105-V2-2020 (cf. [11]). Its security services include

- digital signature creation and verification (including timestamping and key usage counters),
- secure channel establishment with remote entities,
- authenticity, integrity and confidentiality protection of data transferred via a channel,
- data encryption/decryption with integrity protection,
- clustering for performance scalability and high availability.

While the TOE implements all required functions according to BSI-CC-PP-0111-2019, BSI-CC-PP-0112-2020, BSI-CC-PP-0113-2020, it can be specifically configured to provide only the necessary functions. Functions that are not required for using the TOE in the SMAERS context can be disabled by the use of a configuration file.

The following functions can be deactivated:

- Hash generation with SHA-384 and SHA-512 (according to FCS_COP.1/Hash),
- RSA key pair generation (according to FCS_CKM.1/RSA),
- ECKA-EG key generation (according to FCS_CKM.1/ECKA-EG)
- ECKA-EG key derivation (according to FCS_CKM.5/ECKA-EG)
- Key generation and RSA encryption (according to FCS_CKM.1/AES_RSA),
- RSA key derivation and decryption (according to FCS_CKM.5/AES_RSA),
- Key wrapping (according to FCS_COP.1/KW, FCS_COP.1/KU, FPT_TCT.1/CK, FPT_TIT.1/CK),
- HMAC (according to FCS_COP.1/HMAC),
- Creation and verification of RSA signatures (according to FCS_COP.1/CDS-RSA, FCS_COP.1/VDS-RSA),
- Proof of identity by chip authentication (according to FIA_API.1/CA),
- Key agreement by terminal and chip authentication (according to FCS_CKM.1/TCAP),
- Password authentication, certificate based terminal authentication, simplified TA, chip authentication (according to FIA_UAU.5).

TOE type

The Target of Evaluation (TOE) is a Cryptographic Service Provider Light (CSPLight) claiming the *Common Criteria Protection Profile Configuration Cryptographic Service Provider Light - Time Stamp Service and Audit - Clustering (PPC-CSPLight-TS-Au-Cl)* (cf. [10]). Consequently, the TOE is compliant with the Base-PP (cf. [9]) and in addition with the *Protection Profile-Module CSPLight Time Stamp Service and Audit* (cf. [8]) and the *Protection Profile-Module CSPLight Clustering* (cf. [10]). The TOE is dedicated to provide cryptographic services for the protection of the confidentiality and the integrity of user data, and for entity authentication. Parts of these features are supported by the underlying hardware/software platform.

Non-TOE hardware/software/firmware available to the TOE

The TOE is run within a dedicated hardware platform together with a Linux operating system that supports execution of the CSPLight. The TOE is executed in single-user mode as required by [11] "Appendix: Operational Requirements for CSPLight". Please note that the single-user mode in this context refers to the fact that no other software is executed on the OS and does not refer to the run-level of linux. The operating system delivered with the TOE supports hard disk encryption. The hardware platform emits tamper events to the TOE. The TOE tests the correct functioning of the hardware platform as part of its self-testing.

For its operation, the TOE needs the following hardware and software requirements fulfilled in its environment.

- The PrimeKey SEE as the execution platform[1] which has been certified according to FIPS 140-2 level 3.
- alpine GNU/Linux
- OpenJDK 11 (Linux Package)
- userland tools (based on BusyBox [2] as this is the default of Alpine Linux)
- A PostgreSQL Database [3]
- The chrony NTP client [4]
- A CPU with at least 1 GHz
- 2 GB ECC RAM dedicated to the execution of the TOE (in addition to the RAM consumed by the OS)
- 500 GB of hard disk space

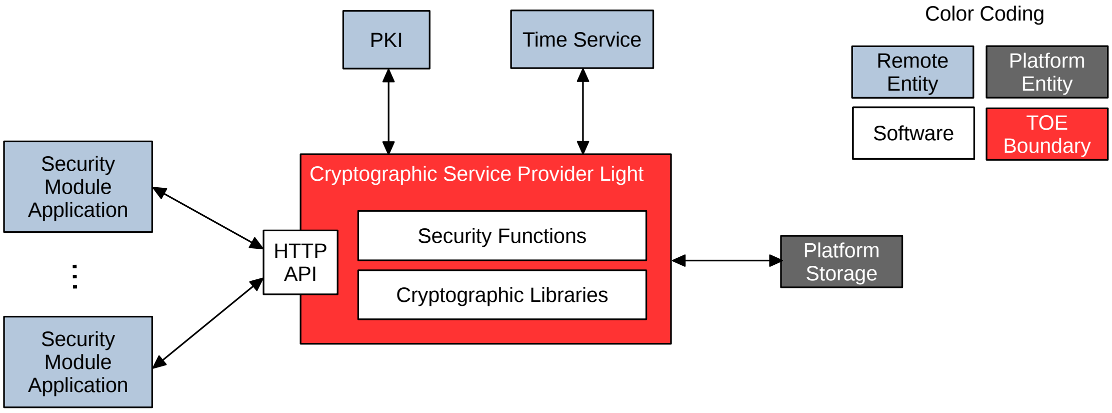


Figure 1.1: Overview of the composed IT product

1.3 TOE Description

The TOE is defined as a software component, i.e. a cryptographic library. The TOE is installed on and runs on a dedicated hardware platform. The hardware platform is not part of the TOE, but the TOE adheres to the platform guidance and the TOE relies on functionality provided by the operating system.

The TOE security functionality (TSF) is logically defined by a common set of security services for users and security mechanisms for internal use. The cryptographic services for users comprise

- authentication of users,
- authentication and attestation of the TOE to entities,
- data authentication and non-repudiation including time stamps,
- encryption and decryption of user data,
- trusted channel functionality including mutual authentication of the communicating entities, encryption and message authentication proof for the sent data, decryption and message authentication verification for received data,
- management of cryptographic keys with security attributes including key generation, key derivation and key agreement, internal storage of keys, import and export of keys with protection of their confidentiality and integrity,
- generation of random bits which may be used for security services outside the TOE.

The TOE is intended to be used as one part within a larger, composed IT product (cf. Figure 1.1). Such products consist of the TOE and one or more application components ("Security

[1] https://www.primekey.com/products/hardware/see/
[2] https://busybox.net/
[3] https://www.postgresql.org/
[4] https://chrony.tuxfamily.org/

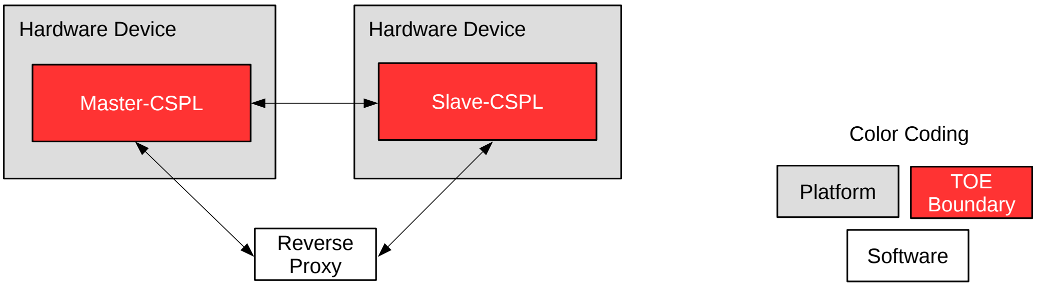


Figure 1.2: Clustering

Module Applications" or SMAs). The TOE provides the security services for these application components.

The TOE and the application component are physically separated components interacting through a trusted/secure channel. The application component (in client role) uses the security services of the TOE (in server role). The secure channel is protected by means of cryptographic mechanisms and provides authenticity, integrity and confidentiality. The TOE functionally always supports to establish a cryptographically protected secure channel between the TOE and external entities.

The TOE supports downloading, authenticity verification and decryption of update code packages (UCPs) for the CSPLight. The UCP contains the operating system, necessary non-TOE software required by the TOE, and the TOE.

The TOE provides functionality to establish a cluster of TOE samples. Each cluster consists of at least two TOE samples and supporting non-TOE components (cf. Figure 1.2). Consequently, this security target claims conformance to the PP configuration consisting of the Base-PP (cf. [9]) together with the PP-Module "CSPLight Clustering (PPM-Cl)" (cf. [10]).

The TOE provides a time service, time stamp service and secure auditing functionality. Consequently, this security target claims conformance to the PP configuration consisting of the Base-PP (cf. [9]) together with the PP-Module "CSPLight Time Stamp Service and Audit (PPM-TS-Au)" (cf. [8]). The TOE uses an external trustworthy entity (cf. "Time Service" in Figure 1.1) for time synchronization via authenticated/signed NTP.

Method of use

The TOE is intended to be used with different applications. The TOE security services are logically separated and provided through well-defined external interfaces towards these applications. The operational environment must not affect the security and correctness of the TSF, and it must support the availability of the TSF.

The TOE provides time service and time stamp service as additional method of use compared with those of the TOE defined in the Base-PP. The time service provides users with reliable time as known to the TOE. The time stamp service provides evidence some user data are provided to the TOE at given point in time. The security audit can be used to make the user responsible for their actions including those described in the Base-PP. The audit records can be exported in a signed and time stamped form.

The TOE provides clustering as additional method of use compared with those of the TOE defined in the Base-PP.

Life cycle

The life cylce of the TOE can be summarized by the following phases

Development and Test
In this phase, the TOE is developed and tested.

Release
The release phase basically represents the decision that a certain version of a TOE shall be published. This decision to authorize a release is taken by the CTO and CTO-designated users in each project.

Delivery
This phase represents the delivery to the customer. It is the phase, in which the TOE leaves the secure premises of fiskaly GmbH .

Personalization
The fiskaly Cloud Crypto Service Provider requires a personalization phase to generate required key material and other TSF data to bind the TOE to a specific customer.

Operation
After personalization, the TOE starts the actual usage phase. All security functionality of the TOE is operating as specified within this Security Target.

Update
In case of bug fixes or addition of new features to the TOE, fiskaly GmbH will create updated releases of the TOE. Please note that the Update can also be seen as the end of life of the previous TOE as it will not longer be existing.

End of Life
Each product of fiskaly GmbH that undergoes certification has a defined End of Life phase. This phase is entered intentionally if the operational phase of a TOE should be ended permanently.

Physical Scope of the TOE

The TOE is installed by the developer on a non-TOE secure hardware platform. Updates are delivered by the developer via an Update Code Package (UCP). The TOE checks the authenticity of the UCP and decrypts it internally, so that the confidentiality of the TOE software is protected. In the general case, the hardware platform and software platform are not part of the TOE.

This ST is part of the TOE and publicly available. As the only non-strictly-confidential guidance document, the “Preparative Procedures & Operational User Guidance Documentation – fiskaly

Cloud Crypto Service Provider" document (cf. [15]) is part of the TOE. This document can be made available to parties who have signed an NDA with fiskaly GmbH. Other evaluation evidence is considered strictly confidential, is only available to the evaluator, and is therefore not enlisted in this ST.

Logical Scope of the TOE

The TOE provides the following security features:

- Key Management:
  - Access control to key management functions (e.g., *key export*)
  - Set and change default values (e.g., for *key validity time period*)
  - Internal management of a key's properties (such as the *key usage counter*)
  - Restrictions (e.g., prevent export of keys with *key usage counter*)
  - Key Generation & Derivation:
    - AES
    - Elliptic Curve Cryptography
    - RSA
  - Key Agreement:
    - Elliptic Curve Diffie-Hellman
    - ECKA-EG
  - Key Destruction: Prevention of leakage of cryptographic keys after their deallocation
  - Key Wrapping: Key export from the TOE and key import to the TOE with protection of the confidentiality of the key
  - Hash Generation: Hashing of data based on
    - SHA-256
    - SHA-384
    - SHA-512
  - Certificate Management:
    - Management of root public key of a PKI
    - Verification of the integrity of certificates
    - Import of TSF data from valid certificates
  - Random Number Generation:
    - Random number service for security module applications
    - Random bit generation for key generation and key agreement
- AES Data Encryption & Decryption
- Hybrid Data Encryption with Message Authentication Code: Encrypt-then-MAC
- Data Integrity Protection:
  - HMAC

  - CMAC
- Digital Signature Service:
  - ECDSA
  - RSA
- Trusted Channel Establishment:
  - PACE
  - Terminal authentication
  - Chip authentication
- Attestation: Verification of the genuineness of the TOE sample
- User identification and authentication
  - Management of authentication reference data and authentication data records
  - Authentication failure handling
  - User-Role binding
  - Time-limited authorization
  - Multiple authentication mechanisms
  - Termination of a session under certain conditions (e.g., power on)
- Access Control: Role-based access control
- Security Management: Secure management of
  - security functions,
  - security roles,
  - security attributes, and
  - security functions behavior
- Testing & Preservation of Secure State
- Code Updates: Verification and decryption of Update Code Packages (UCPs)
- Auditing & Time stamping:
  - Generation of audit logs of auditable events
  - Time synchronization with a trusted time service
- Clustering: Scalability of performance and availability

Chapter 2

Conformance Claims

2.1 CC Conformance Claims

This ST claims conformance to CC version 3.1 revision 5. Particularly, the conformance to CC Part 2 (security functional requirements) [6] is CC Part 2 extended, the conformance to CC Part 3 (security assurance requirements) [7] is CC Part 3 conformant.

2.2 PP Claims

This ST claims *strict* conformance to the Base-PP: Protection Profile Cryptographic Service Provider Light (CSPLight) Version 1.0, BSI-CC-PP-0111-2019 [9].

This ST claims *strict* conformance to the PP-Module: Protection Profile-Module CSPLight Time Stamp Service and Audit (PPM-TS-Au), Version 1.0, BSI-CC-PP-0112-2020 [8].

This ST claims *strict* conformance to the PP-Module: Protection Profile-Module CSPLight Clustering (PPM-Cl), Version 1.0, BSI-CC-PP-0113-2020 [10].

2.3 Package Claims

The evaluation assurance level of the TOE is EAL2 augmented with ALC_CMS.3 and ALC_LCD.1.

2.4 Conformance Claim Rationale

The TOE type is a Cryptographic Service Provider Light (CSPLight) component consistent with the TOE type of the claimed PPs (cf. [9, 10, 8]).

Chapter 3

Security Problem Definitions

3.1 Introduction

Assets

The assets of the TOE are

- user data, whose integrity and confidentiality shall be protected,
- user data and time stamps, whose integrity shall be protected,
- cryptographic services and keys which shall be protected against unauthorized use or misuse, and whose integrity shall be protected,
- update code packages (UCP), whose integrity and confidentiality shall be protected,
- time services which time base shall be protected against manipulation,
- additional TSF-data (e.g. security flags), whose integrity and/or confidentiality shall be protected,
- other TOE resources, whose unauthorized use and misuse shall be prevented.

The cryptographic keys are TSF data because they are used for cryptographic time stamp operations protecting user data and audit records, and the enforcement of the SFR relies on these data for the operation of the TOE.

The audit records are TSF data generated by the TSF and exported to the user.

The TOE protects the TSF data, the security attributes of the known users and the cryptographic keys with their security attributes transferred between Master-CSPLight and Slave-CSPLight(s).

Users and subjects

The TOE knows external entities (users) as

- *human user* communicating with the TOE for security management of the TOE,
- *application component* using the cryptographic and other security services of the TOE and supporting the communication with remote entities (e. g. by providing certificates),
- *cluster-CSPLight* being another TOE sample in a cluster with the TOE.
- *remote entity* exchanging user data and TSF data with the TOE over insecure media.

The TOE communicates with

- *human user* through a secure channel,
- *application component* through a secure channel,
- *cluster-CSPLight* in encrypted and integrity protected form,
- remote entities over a trusted channel using cryptographic mechanisms including mutual authentication.

The subjects as active entities in the TOE perform operations on objects. Objects obtain their associated security attributes from the authenticated users, or the security attributes are defined by default values.

Objects

The TSF operates user data objects and TSF data objects (i. e. passive entities, that contain or receive information, and upon which subjects perform operations). The TSF data objects contain the security attributes of the known users and the cryptographic keys with their security attributes transferred between Master-CSPLight and Slave-CSPLights. User data objects are imported, used in cryptographic operation, temporarily stored, exported and destroyed after use. User data objects of the time stamp service are imported, used in time stamp operation, exported and destroyed after use. The update code packages are user data objects that are imported and stored in the TOE until they are used to create an updated version of the CSPLight. TSF data objects are created, temporarily or permanently stored, imported, exported and destroyed as objects of the security management. They may contain e. g. cryptographic keys with their security attributes, certificates, or authentication data records with authentication reference data of a user. Cryptographic keys are objects of the key management.

Security attributes

A *Role* is a set of certain access rights and permissions. By defining roles, and associating users with roles (“a user or a subject takes a role”) it is immediately clear, what access rights and permissions this user is granted.

The security attributes of users known to the TOE are stored in *Authentication Data Records* containing

- *User Identity* (User-ID),
- *Authentication reference data*,
- *Role*.

Passwords as Authentication Reference Data have the security attributes

- *status*: values *initial password, operational password*,
- *number of unsuccessful authentication attempts.*

Certificates contain security attributes of users including User Identity, a public key and security attributes of the key. If certificates are used as authentication reference data for cryptographic entity authentication mechanisms they may contain the *Role* of the entity.

The TOE knows the following roles that can be taken by a user or a subject:

- *Unidentified User*: this role is associated with any user not (successfully) identified by the TOE. This role is assumed after start-up of the TOE. The TSF associated actions allowed for the Unidentified User are defined in SFR FIA_UID.1.
- *Unauthenticated User*: this role is associated with an identified user but not (successfully) authenticated user. The TSF associated actions allowed for the Unauthenticated User are defined in SFR FIA_UAU.1.
- *Administrator*: a successful authenticated user in this role is allowed to access the TOE in order to perform management functions. It is taken by a human user or a subject acting on behalf of a human user after successful authentication as an Administrator.

  The Administrator role is split into more detailed roles:

  - the *Crypto-Officer* is allowed to access the TOE in order to manage a cryptographic TSF.
  - the *User Administrator* is allowed to access the TOE in order to manage users and to generate cluster keys.
  - the *Update Agent* is allowed to import and install update code packages.
  - the *Auditor* role that is allowed to configure the audit functionality, review audit data and export audit trails.
  - the *Timekeeper* is allowed to adjust the internal time.

  The SFR uses the general term Administrator or a selection between Administrator role and these detailed roles in case they are supported by the TOE and separation of duties is appropriate.
- *Key Owner*: successful authenticated user allowed to perform cryptographic operation with his own keys. This role may be claimed by human user or an entity.

- *Application Component*: subjects in this role are allowed to use assigned security services of the TOE without being authenticated as a human user (e. g. exporting and importing of wrapped keys). This role may be assigned to an entity communicating through a physically separated secure channel or through a trusted channel (which requires assured identification of its end points).
- *Cluster-CSPLight:* another TOE sample in a cluster with the TOE with security attribute *Master-CSPLight or Slave-CSPLight.* This role is bound to the communication through the trusted channel between cluster CSPLights established by the administrator.

The user uses authentication verification data to prove its identity to the TOE. The TSF uses Authentication reference data to verify the claimed identity of a user. The TSF supports

- human user authentication by knowledge, where the authentication verification data is a password and the authentication reference data is a password or an image of the password e. g. a salted hash value or a derived cryptographic key,
- human user authentication by possession of a token, or as user of a terminal by implementing user authentication by cryptographic entity authentication mechanisms,
- cryptographic entity authentication mechanisms where the authentication verification data is a secret or private key and the authentication reference data is a secret or public key.

A human user may authenticate himself to the TOE, and the TOE authenticates itself to an external entity in charge of the authenticated authorized user.

The TOE is delivered with initial Authentication Data Records for Unidentified User, Unauthenticated User and administrator role(s). The Authentication Data Records for Unidentified User and Unauthenticated User have no Authentication Reference Data. The roles are not exclusive, i. e. a user or subject may be in more then one role, e. g. a human user may claim the Crypto-Officer and Key Owner role at the same time. The SFR may define limitation on roles (especially combinations of roles) a user may be associated with.

Cryptographic keys have at least the security attributes

- *Key identity*, i.e. an attribute that uniquely identifies the key,
- *Key Owner*, i.e. the identity of the owner this key is assigned to,
- *Key type*, i.e. whether the key is as secret key, a private key, or a public key,
- *Key usage type*, an attribute that identifies the cryptographic mechanism or services the key can be used for; keys for time stamp service (cf. FDP_DAU.2/TS) have the key usage type "*TimeStamp*"; the PP-Module PPM-Cl uses the clustering encryption key for cryptographic operation according to FCS_COP.1/ED and clustering MAC keys for cryptographic operation according to FCS_COP.1/MAC,
- *Key access control attributes*, i.e. a list of combinations of the identity of the user, the role for which the user is authenticated, and the allowed key management functions or cryptographic operations; this includes that
  - the *import* of the key is allowed or forbidden,

  - the *export* of the key is allowed or forbidden,
  - *Clustering:* transfer of the key in a cluster of TOE samples (i.e. export by TOE as Master-CSPLight and import by TOE as Slave-CSPLight) is allowed or forbidden,

and may have the security attributes

- *Key validity time period*, i.e. the time period for operational use of the key; the key must not be used before or after this time slot,
- *Key usage counter*, i.e. the number of operations performed with this key, where the key usage counter of the private key used for counts the number of created signature.

The UCP have at least the security attributes

- *issuer* of the UCP,
- *version number* of the UCP,

3.2 Threats

T.DataCompr Compromise of communication data
An unauthorized entity gets knowledge of information that are stored on media controlled by the TSF, or an unauthorized entity gets knowledge of information that are transferred between the TOE and an authenticated external entity.

T.DataMani Unauthorized generation or manipulation of communication data
An unauthorized entity generates or manipulates user data that are stored on media controlled by the TSF or transferred between the TOE and an authenticated external entity, and manipulates such data so that they are accepted as valid by the recipient.

T.Masqu Masquerade authorized user
A threat agent masquerades as an authorized entity in order to gain unauthorized access to user data, TSF data, or TOE resources.

T.ServAcc Unauthorized access to TOE security services
An attacker gets unauthorized access to security services of the TOE.

T.PhysAttack Physical attacks
An attacker gets physical access to the underlying hardware platform that the TOE is running on and may (1) disclose or manipulate user data under TSF control and TSF data, and (2) affect TSF by (a) physical probing and manipulation, (b) applying environmental stress or (c) exploiting information leakage from the TOE.

T.FaUpD Faulty Update Code Package
An unauthorized entity provides and installs a faulty update code package. Thus attacks against the integrity of the TSF implementation, and against the confidentiality and integrity of user data and TSF data becomes possible.

3.3 Organizational Security Policies

OSP.SecCryM Secure cryptographic mechanisms
The TOE uses only secure cryptographic mechanisms as confirmed by the certification body for the specified TSF, the assurance security requirements and the operational environment.

OSP.SecService Security services of the TOE
The TOE provides security services to the authorized users for encryption and decryption of user data, authentication prove and verification of user data, entity authentication to external entities including attestation, trusted channels and random bit generation.

OSP.KeyMan Key Management
The key management ensures the integrity of all cryptographic keys and the confidentiality of all secret or private keys over the whole life cycle. The life-cycle comprises key generation, storage, distribution, application, archival and deletion. The cryptographic keys and cryptographic key components shall be generated, operated and managed by secure cryptographic mechanisms, assigned to the secure cryptographic mechanisms they are intended to be used with, and to the entities authorized for their use.

OSP.TC Trust centre
Trust centres provide secure certificates for trustworthy certificate holders with correct security attributes. The TOE uses certificates for identification and authentication of users, access control and secure use of security services of the TOE. In particular, this includes key management and attestation.

OSP.Update Authorized Update Code Packages
Update Code Packages are delivered in encrypted form, and are signed by the authorized issuer. The TOE verifies the authenticity of the received Update Code Package using the CSP before storing any update data in the TOE. The TOE restricts the storage of authentic Update Code Package to authorized users.

The PP-Module PPM-TS-Au [8] adds new organizational security policies OSP.TimeService and OSP.Audit:

OSP.Audit Audit for key management and cryptographic operations
The TOE provides security auditing related to activities controlled by the TSF and security critical events. The security auditing provides evidence to make users responsible for actions they are authorized for and to protect users against unwarranted accusation. The Administrator is allowed to select auditable events.

OSP.TimeService Time Service and Time stamp service
The TOE provides non-cryptographic time service and cryptographic time stamp service for user data and TSF data. The time stamp service provides evidence that user data were presented to the TSF and exported audit data were generated at certain point in time and in a verifiable sequence.

The PP-Module PPM-Cl [10] adds new organizational security policy OSP.Cluster:

OSP.Cluster Cluster of TOE samples
The administrator establishes and manages a cluster of multiple TOE samples for secure transfer of the security attributes of the known users and the cryptographic keys as necessary for

scalability of performance and availability of security services.

3.4 Assumptions

A.SecComm Secure communication
Remote entities support trusted channels by cryptographic mechanisms. The operational environment shall protect the local communication channels by trusted channels using cryptographic mechanisms, or by secure channels using non-cryptographic security measures. The operational environment must be subject to a security audit that verifies that the communication between the TOE and the application is indeed protected.

The PP-Module PPM-Cl [10] adds the following assumption additional to those defined in the Base-PP:

A.ClusterAppl Cluster management by application
The application using the security services of the TOE transfers security attributes of the known users and cryptographic keys with their security attributes between Master-CSPLight and Slave-CSPLights as necessary for scalability of performance and availability of security services.

Chapter 4

Security Objectives

4.1 Security Objectives for the TOE

O.AuthentTOE Authentication of the TOE to external entities
The TOE authenticates itself in charge of authorized users to external entities by means of secure cryptographic entity authentication and attestation.

O.Enc Confidentiality of user data by encryption and decryption
The TOE provides secure encryption and decryption as security services for the users to protect the confidentiality of exported or imported user data, or user data stored on media that is within the scope of control of the TSF.

O.DataAuth Data authentication by cryptographic mechanisms
The TOE provides secure symmetric and asymmetric data authentication mechanisms as security services for the users to protect the integrity and authenticity of user data.

O.RBGS Random bit generation service
The TOE provide cryptographically secure random bit generation for the users.

O.TChann Trusted channel
The TSF provides trusted channel functionality using secure cryptographic mechanisms for the communication between the TSF and external entities. The TOE provides authentication of all communication end points, and ensures the confidentiality and integrity of the communication data that are exchanged through the trusted channel.

Note that the TSF can establish the trusted channel by means of secure cryptographic mechanisms only if the other external entity supports these secure cryptographic mechanisms as well. If the trusted channel cannot be established by means of secure cryptographic mechanisms - i.e. due to missing security functionality on the user side - then the operational environment shall provide a secure channel that protects the communication by non-cryptographic security mechanisms, cf. A.SecComm and OE.SecComm.

O.I&A Identification and authentication of users
The TOE shall uniquely identify users and verify the claimed identity of the user before pro-

viding access to any controlled resources; The TOE shall authenticate IT entities using secure cryptographic mechanisms.

O.AccCtrl Access control
The TOE provides access control of security services, operations on user data, and management of TSF and TSF data.

O.SecMan Security management
The TOE provides security management of users, TSF, TSF data and cryptographic keys by means of secure cryptographic mechanisms and certificates. The TSF generates, derives, agrees, imports and exports cryptographic keys as a security service for users and for internal use. The TSF shall destruct unprotected secret or private keys in such a way that any previous information content of the resource is made unavailable.

O.TST Self-test
The TSF performs self-tests during initial start-up, and after power-on. The TSF enters a secure state if the self-test fails or if attacks are detected. It relies on the underlying hardware platform and operating system (cf. OE.SecPlatform) to implement this functionality.

O.SecUpCP Secure import of Update Code Packages
The TSF verifies the authenticity of a received encrypted Update Code Package, decrypts the Update Code Package if it is verified to be authentic, and installs it after verifying that it is suitable for the TOE and does not downgrade the TOE's firmware to a previous version.

The PP-Module PPM-TS-Au [8] adds new security objectives O.Audit and O.TimeService:

O.Audit Audit
The TSF provides security auditing of selected user activities controlled by the TSF and security critical events. The Administrator is allowed to select auditable events, to manage the audit functionality and the export of audit records.

O.TimeService Time services
The TOE provide an internal time service and time stamp service for the user

The PP-Module PPM-Cl [10] adds new security objective O.Cluster:

O.Cluster Cluster
The TSF supports cluster of TOE samples by secure transfer of the security attributes of the known users and the cryptographic keys with their security attributes between Master-CSPLight and Slave-CSPLights in encrypted and integrity protected form.

4.2 Security Objectives for the Operational Environment

OE.CommInf Communication infrastructure
The operational environment shall provide a public key infrastructure for entities in the relevant communication networks. Trust centres must generate secure certificates for trustworthy certificate holders with correct security attributes. They must distribute their certificate signing public key securely such that a verification of the digital signature of the generated certificates is possible. Trust centres should further operate a directory service for dissemination of certificates and provision of revocation status information of certificates.

OE.AppComp Support of the Application component
The Application component supports the TOE for communication with users and trust centres.

OE.SecManag Security management
The operational environment shall implement appropriate security management functionality for secure use of the TOE. This includes user management as well as key management. It ensures secure key management outside of the TOE and uses the trust centre's services to determine the validity of certificates. Cryptographic keys and cryptographic key components shall be assigned to the secure cryptographic mechanisms they are intended to be used with, and to the entities authorized for their use.

OE.SecComm Protection of communication channel
Remote entities shall support establishing trusted channels with the TOE by using cryptographic mechanisms. The operational environment shall protect the local communication channels by trusted channels using cryptographic mechanisms, or by secure channels using non-cryptographic security measures. In the latter case, the operational environment must be subject to a security audit that verifies that the communication between the TOE and the application is indeed protected.

OE.SUCP Signed Update Code Packages
The secure Update Code Package is delivered in encrypted form and signed by the authorized issuer together with its security attributes.

OE.SecPlatform Secure Hardware Platform
The TOE runs on a secure hardware platform. The hardware platform and its operating system support the implementation of the TSF; this in particular includes the protection of the confidentiality and integrity of user data, TSF data and its correct operation against physical attacks and environmental stress.

The PP-Module PPM-TS-Au [8] adds new security objectives for the operational environment OE.Audit and OE.TimeSource:

OE.Audit Review and availability of audit records
The Administrator shall ensure the regular audit review and the availability of exported audit records.

OE.TimeSource External time source
The operational environment provides reliable external time source for the adjustment of the TOE internal time source.

The PP-Module PPM-Cl [10] adds new security objectives for the operational environment OE.ClusterCtrl and OE.TSFdataTrans:

OE.ClusterCtrl Control of the cluster
The administrator establishes and manages a cluster only of trustworthy samples of the TOE as necessary for scalability of performance and availability of security services.

OE.TSFdataTrans Transfer of TSF data within the CSPLight cluster
The administrator and the application using the security services of the TOE, transfer the security attributes of the known users and the cryptographic keys with their security attributes between Master-CSPLight and Slave-CSPLights as necessary for scalability of performance and availability of security services.

4.3 Security Objective Rationale

Table 4.1 traces the security objectives for the TOE back to threats countered by that security objective and OSPs enforced by that security objective, and the security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

The following part of the chapter demonstrate that the security objectives counter all threats and enforce all OSPs, and the security objectives for the operational environment uphold all assumptions.

The threat T.DataCompr "Compromise of communication data" is countered by the security objectives for the TOE and the operational environment:

- O.Enc requires the TOE to provide encryption and decryption as a security service for the users to protect the confidentiality of user data,
- O.TChann requires the TOE to support establishing a trusted channel between the TSF and the application component, between the TSF and other users, and between the application component and other users. The trusted channel ensures authentication of all communication end points, and protected communication for the confidentiality and integrity of the communication and to prevent misuse of sessions of authorized users.
- OE.AppComp requires the application component to support the TOE for communication with users and trust centres.
- OE.CommInf requires the operational environment to provide a communication infrastructure; especially w.r.t. trust centre services.
- OE.SecComm requires the operational environment to protect the confidentiality and integrity of communication over local communication channels by physical security measures, and requires remote entities to support trusted channels by means of cryptographic mechanisms. If a trusted channel cannot be established due to missing security functionality of the application component, the operational environment shall protect the communication, cf. A.SecComm and OE.SecComm. Note that OE.SecComm requires measures that the operational environment must be subject to a security audit that verifies that the communication between the TOE and the application is indeed protected.

The threat T.DataMani "Unauthorized generation or manipulation of communication data" is countered by the security objectives for the TOE and the operational environment:

- O.DataAuth requires the TOE to provide symmetric and asymmetric data authentication mechanisms as a security service for the users to protect the integrity and authenticity of user data.
- O.TChann requires the TOE to support trusted channels for the authentication of all communication end points, for the protected communication with the application component, and for other users. This ensures the confidentiality and integrity of the communication between the TOE and the other parties and prevents misuse of sessions of authorized users.

	T.DataCompr	T.DataMani	T.Masqu	T.ServAcc	T.PhysAttack	T.FaUpD	OSP.SecCryM	OSP.SecService	OSP.KeyMan	OSP.TC	OSP.Update	OSP.Audit	OSP.TimeService	OSP.SecCryM	OSP.Cluster	A.ClusterAppl	A.SecComm
O.AccCtrl				×													
O.AuthentTOE							×	×									
O.DataAuth		×					×	×									
O.Enc	×						×	×									
O.I&A			×	×			×	×									
O.RBGS							×	×									
O.SecMan			×				×		×	×							
O.SecUpCP						×					×						
O.TChann	×	×	×	×			×	×									
O.TST					×												
O.Audit												×					
O.TimeService													×				
O.Cluster														×	×		
OE.AppComp	×	×		×						×							
OE.CommInf	×	×		×				×	×	×							
OE.SecComm	×	×		×													×
OE.SecManag			×					×	×								
OE.SUCP						×					×						
OE.SecPlatform					×												
OE.Audit												×					
OE.TimeService													×				
OE.ClusterCtrl															×		
OE.TSFdataTrans															×	×	

Table 4.1: Security objective rationale (Table 1 in Base-PP [9])

- OE.AppComp requires the application component to support the TOE for communication with users and trust centres.
- OE.CommInf requires the operational environment to provide trust centre services and securely distribute root public keys.
- OE.SecComm requires the operational environment to protect the confidentiality and integrity of communication with the TOE. Remote entities shall support trusted channels with the TOE using cryptographic mechanisms. The operational environment shall protect local communication channels by trusted channels using cryptographic mechanisms, or by secure channels using non-cryptographic security measures.

The threat T.Masqu “Masquerade authorized user” is countered by the security objectives for the TOE and the operational environment:

- O.I&A requires the TSF to identify uniquely users and verify the claimed identity of the user before providing access to any controlled resources.
- O.TChann requires the TSF to provide authentication of all communication end points of the trusted channel.
- O.SecMan requires the TSF to provides security management of users, TSF, TSF data and cryptographic keys by means of secure cryptographic mechanisms and certificates.
- OE.SecMan requires the operational environment to implement appropriate security management functionality for the secure use of the TOE. This includes user management.

The threat T.ServAcc “Unauthorized access to TOE security services” is countered by the security objectives for the TOE and the operational environment:

- O.I&A requires the TSF to uniquely identify users and to authenticate users before providing access to any controlled resources.
- O.AccCtrl requires the TSF to control access of security services, operations on user data, and management of TSF and TSF data.
- O.TChann requires mutual authentication of the external entity and the TOE, and the authentication of communicated data between them to prevent misuse of the communication with external entities. The operational environment is required by OE.SecComm to ensure that a secure channel is available if a trusted channel cannot be established.
- The operational environment OE.CommInf requires the provision of a public key infrastructure for entity authentication. OE.AppComp requires the application to support the communication with trust centres.

The threat T.PhysAttack “Physical attacks” is countered by the next security objectives:

- OE.SecPlatform ensures that the TOE runs on a secure hardware platform and operating system that provides protection against physical attacks.

- As means to ensure robustness against perturbation O.TST requires the TSF to perform self-tests and to enter a secure state if the self-test fails or attacks are detected.

The threat T.FaUpD "Faulty Update Code Package" is directly countered by the security objective O.SecUpCP verifying the authenticity of UCP under the condition that trustworthy UCPs are signed as required by OE.SUCP

- O.SecUpCP "Secure import of Update Code Package" requires the TOE to verify the authenticity of received encrypted Update Code Packages before decrypting and storing an authentic Update Code Package
- OE.SUCP "Signed Update Code Packages" requires the *Issuer* to sign both the secure Update Code packages as well as its security attributes.

The organizational security policy OSP.SecCryM "Secure cryptographic mechanisms" is implemented by means of secure cryptographic mechanisms required in

- O.I&A "Identification and authentication of users" and O.AuthentTOE "Authentication of the TOE to external entities" which require secure entity authentication of users and the TOE,
- O.Enc "Confidentiality of user data by means of encryption and decryption" and O.DataAuth "Data authentication by cryptographic mechanisms" require secure cryptographic mechanisms for protection of the confidentiality and integrity of user data,'
- O.TChann "Trusted channel" require secure cryptographic mechanisms for entity authentication of users and the TOE, and the protection of confidentiality and integrity of communication data.
- O.RBGS "Random bit generation service" requires the TOE to provide a cryptographically secure random bit generation service for the users.
- O.SecMan "Security management" requires secure management of TSF data and cryptographic keys by means of secure cryptographic mechanisms and certificates.

The organizational security policy OSP.SecService "Security services of the TOE" is directly implemented by security objectives for the TOE O.Enc "Confidentiality of user data by means of encryption and decrypti", O.DataAuth "Data authentication by cryptographic mechanisms", O.I&A "Identification and authentication of users", O.AuthentTOE "Authentication of the TOE to external entities", O.TChann "Trusted channel" and O.RBGS "Random bit generation servic", which require the TSF to provide cryptographic security services for the user. The OSP.SecService is supported by OE.CommInf "Communication infrastructure" and OE.SecManag "Security management" which provide the necessary measures for the secure use of these services.

The organizational security policy OSP.KeyMan "Key Management" is directly implemented by O.SecMan "Security management" and supported by trust centre services according to OE.CommInf "Communication infrastructure" and OE.SecManag "Security management".

The organizational security policy OSP.TC "Trust centre" is implemented by security objectives for the TOE and the operational environment:

- O.SecMan "Security management" uses certificates for secure management of users, TSF, TSF data and cryptographic keys.
- OE.CommInf "Communication infrastructure" requires trust centres to generate secure certificates for trustworthy certificate holders with correct security attributes, and to distribute certificates and revocation status information.
- OE.AppComp "Support of the Application component" requires the Application component to support the TOE for the communication with trust centres.

The organizational security policy OSP.Update "Authorized Update Code Packages" is implemented directly by the security objectives for the TOE O.SecUpCP and the operational environment OE.SUCP.

The assumption A.SecComm "Secure communication" assumes that the operational environment protects the confidentiality and integrity of communication data and ensures reliable identification of its end points. The security objective for the operational environment OE.SecComm require the operational environment to protect local communication physically or via trusted channel, and remote entities to support trusted channels using cryptographic mechanisms.

The organizational security policy OSP.Audit "Audit for key management and cryptographic operations" is directly implemented by

- the security objective for the TOE O.Audit requiring security auditing and
- the security objective for the operational environment OE.Audit requiring the regular audit review and the availability of exported audit records.

The organizational security policy OSP.TimeService "Time Service and Time stamp service" is directly implemented by

- the security objective for the TOE O.TimeService "Time services" requiring the TOE to provide an internal time service and time stamp service for the user, and
- the security objective for the operational environment OE.TimeSource "External time source" requiring the operational environment to provide reliable external time stamps for adjustment of TOE internal time source.

The organizational security policy OSP.SecCryM "Secure cryptographic mechanisms" defined in the BasePP is implemented by means of secure cryptographic mechanisms required in

- O.Cluster "Cluster" requiring secure transfer in encrypted and integrity protected form of the security attributes of the known users and the cryptographic keys with their security attributes between MasterCSPLight and Slave-CSPLights.

The organizational security policy OSP.Cluster "Cluster of TOE samples" is implemented by security objectives for the TOE and the operational environment:

- O.Cluster requiring support for cluster of TOE samples as CSPLights with distribution of Authentication Data Records and cryptographic keys between Master-CSPLight and Slave-CSPLights through a trusted channel keeping the confidentiality and integrity of the security attributes of the known users and of the cryptographic keys with their security attributes.

- OE.ClusterCtrl requiring administrator to build a cluster only of trustworthy samples of the TOE as needed for scalability of performance and availability of security services.

- OE.TSFdataTrans requires the administrator and the application using the security services of the TOE transfer security attributes of the known users and cryptographic keys with their security attributes between Master-CSPLight and Slave-CSPLights as necessary for scalability of performance and availability of security services.

The assumption A.ClusterAppl is directly ensured by OE.TSFdataTrans.

Chapter 5

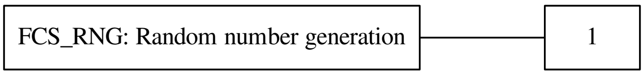
Extended Component Definition

5.1 Generation of random numbers (FCS_RNG)

Family Behaviour

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

Component levelling:



FCS_RNG.1 Generation of random numbers, requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no auditable events foreseen.

FCS_RNG.1 Random number generation

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 random numbers that meet [assignment: *a defined quality metric*].

5.2 Cryptographic key derivation (FCS_CKM.5)

This chapter describes a component of the family Cryptographic key management (FCS_CKM) for key derivation as process by which one or more keys are calculated from either a pre-shared key or a shared secret and other information. Key derivation is the deterministic repeatable process by which one or more keys are calculated from both a pre-shared key or shared secret, and other information, while key generation required by FCS_CKM.1 uses internal random numbers.

The component FCS_CKM.5 is on the same level as the other components of the family FCS_CKM.

Management: FCS_CKM.5

There are no management activities foreseen

Audit: FCS_CKM.5

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

(a) Minimal: Success and failure of the activity.

(b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).

FCS_CKM.5 Requires the TOE to provide key derivation.

FCS_CKM.5 Cryptographic key derivation

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.5.1 The TSF shall derive cryptographic keys [assignment: *key type*] from [assignment: *input parameters*] in accordance with a specified cryptographic key derivation algorithm [assignment: *cryptographic key derivation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

5.3 Authentication Proof of Identity (FIA_API)

To describe the IT security functional requirements of the TOE a sensitive family (FIA_API) of the Class FIA (Identification and authentication) is defined here. This family describes the functional requirements for the proof of the claimed identity for the authentication verification by an external entity where the other families of the class FIA address the verification of the identity of an external entity.

Family Behaviour

This family defines functions provided by the TOE to prove its identity and to be verified by an external entity in the TOE IT environment.

Component levelling:

FIA_API Authentication Proof of Identity
1

FIA_API.1 Authentication Proof of Identity, provides prove of the identity of the TOE to an external entity.

Management: FIA_API.1

The following actions could be considered for the management functions in FMT:

a) Management of authentication information used to prove the claimed identity.

Audit: FIA_API.1

There are no auditable events foreseen.

FIA_API.1 Authentication Proof of Identity

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1 The TSF shall provide a [assignment: *authentication mechanism*] to prove the identity of the [assignment: *object, authorized user or role*] to an external entity.

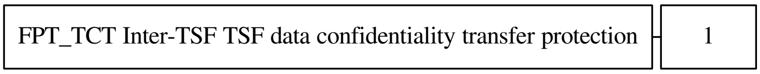
5.4 Inter-TSF TSF data confidentiality transfer protection (FPT_TCT)

This section describes the functional requirements for confidentiality protection of inter-TSF transfer of TSF data. The family is similar to the family Basic data exchange confidentiality (FDP_UCT) which defines functional requirements for confidentiality protection of exchanged user data.

Family Behaviour

This family requires confidentiality protection of exchanged TSF data.

Component levelling:


FPT_TCT Inter-TSF TSF data confidentiality transfer protection
1

FPT_TCT.1 Requires the TOE to protect the confidentiality of information in exchanged the TSF data.

Management: FPT_TCT.1

There are no management activities foreseen.

Audit: FPT_TCT.1

There are no auditable events foreseen.

FPT_TCT.1 TSF data confidentiality transfer protection

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FMT_MTD.1 Management of TSF data or
FMT_MTD.3 Secure TSF data]

FIA_TCT.1.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] by providing the ability to [selection: *transmit, receive, transmit and receive*] TSF data in a manner protected from unauthorised disclosure.

5.5 Inter-TSF TSF data integrity transfer protection (FPT_TIT)

This section describes the functional requirements for integrity protection of TSF data exchanged with another trusted IT product. The family is similar to the family Inter-TSF user data integrity transfer protection (FDP_UIT) which defines functional requirements for integrity protection of exchanged user data.

Family Behaviour

This family requires integrity protection of exchanged TSF data.

Component levelling:

FPT_TIT: TSF data integrity transfer protection
1

FPT_TIT.1 Requires the TOE to protect the integrity of information in exchanged the TSF data.

Management: FPT_TIT.1

There are no management activities foreseen.

Audit: FPT_TIT.1

There are no auditable events foreseen.

FPT_TIT.1 TSF data integrity transfer protection

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FMT_MTD.1 Management of TSF data or
FMT_MTD.3 Secure TSF data]

FPT_TIT.1.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to [selection: *transmit, receive, transmit and receive*] TSF data in a manner protected from [selection: *modification, deletion, insertion, replay*] errors.

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

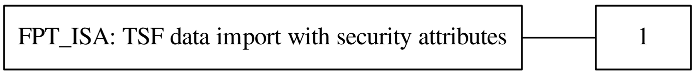
5.6 TSF data import with security attributes (FPT_ISA)

This section describes the functional requirements for TSF data import with security attributes from another trusted IT product. The family is similar to the family Import from outside of the TOE (FDP_ITC) which defines functional requirements for user data import with security attributes.

Family Behaviour

This family requires TSF data import with security attributes.

Component levelling:



FPT_ISA.1 Requires the TOE to import TSF data with security attributes.

Management: FPT_ISA.1

There are no management activities foreseen.

Audit: FPT_ISA.1

There are no auditable events foreseen.

FPT_ISA.1 Import of TSF data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FMT_MTD.1 Management of TSF data or
FMT_MTD.3 Secure TSF data]
[FMT_MSA.1 Management of security attributes, or
FMT_MSA.4 Security attribute value inheritance]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_ISA.1.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] when importing TSF data, controlled under the SFP, from outside of the TOE.

FPT_ISA.1.2 The TSF shall use the security attributes associated with the imported TSF data.

FPT_ISA 1.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the TSF data received.

FPT_ISA 1.4 The TSF shall ensure that interpretation of the security attributes of the imported TSF data is as intended by the source of the TSF data.

FPT_ISA 1.5 The TSF shall enforce the following rules when importing TSF data controlled under the SFP from outside the TOE: [assignment: *additional importation control rules*].

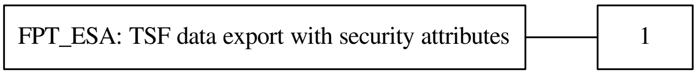
5.7 TSF data export with security attributes (FPT_ESA)

This section describes the functional requirements for TSF data export with security attributes to another trusted IT product. The family is similar to the family Export to outside of the TOE (FDP_ETC) which defines functional requirements for user data export with security attributes.

Family Behaviour

This family requires TSF data export with security attributes.

Component levelling:



FPT_ESA.1 Requires the TOE to export TSF data with security attributes.

Management: FPT_ESA.1

There are no management activities foreseen.

Audit: FPT_ESA.1

There are no auditable events foreseen.

FPT_ESA.1 Export of TSF data with security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data] [FMT_MSA.1 Management of security attributes, or FMT_MSA.4 Security attribute value inheritance] FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_ESA.1.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] when exporting TSF data, controlled under the SFP(s), outside of the TOE.

FPT_ESA.1.2 The TSF shall export the TSF data with the TSF data's associated security attributes.

FPT_ESA 1.3 The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported TSF data.

FPT_ESA 1.4 The TSF shall enforce the following rules when TSF data is exported from the TOE: [assignment: *additional exportation control rules*].

Chapter 6

Security Requirements

The CC allows several operations to be performed on functional requirements: refinement, selection, assignment, and iteration. Each of these operations is used in this PP.
The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is (i) denoted by the word "refinement" in **bold** text and the added/changed words are in bold text, or (ii) directly included in the requirement text as **bold** text. In cases where words from a CC requirement component were deleted, these words are ~~crossed out~~. Refinements made by the ST author are bold and colored blue (example: **refinement**). Deletion refinements by the ST author are bold, colored blue and crossed out (~~**deletion**~~).
The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the PP authors are denoted as italic text and the original text of the component is given by a footnote. Selections made by the ST author are colored blue and have a footnote documenting the selection by the ST author (example: option 1[1]).
The **assignment** operation is used to assign a specific value to an unspecified parameter, such as e.g. the length of a password. Assignments that have been made by the PP authors are denoted by showing as italic text and the original text of the component is given by a footnote. Assignments made by the ST author are italic, colored blue, and have a footnote documenting the assignment by the ST author (example: *made assignment*[2])
The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/" and the iteration indicator after the component identifier.

6.1 Security Functional Requirements

The TOE provides cryptographic security services for encryption and decryption of user data, entity authentication of external entities and to external entities, authentication prove and verification of user data, trusted channel establishment and random number generation.

The TOE enforces the Cryptographic Operation SFP for protection of theses crypto-

[1] [selection (by ST author): option 1, option 2]

[2] [assignment (by ST author): *assignment to be made*]

graphic services. Corresponding Subjects, objects, and operations are defined in the SFRs FDP_ACC.1/Oper and FDP_ACF/Oper.

The TOE provides hybrid encryption and decryption combined with data integrity mechanisms for the cipher text as a cryptographic security service of the TOE. The encryption FCS_COP.1/HEM combines the generation of a data encryption key and message authentication code (MAC) key, the asymmetric encryption of the data encryption key with an asymmetric key encryption key, cf. FCS_CKM.1/ECKA-EG, FCS_CKM.1/RSA, and the symmetric encryption of the data with the data encryption key and data integrity mechanism with MAC calculation for the cipher text. The receiver reconstructs the data encryption key and the MAC key, cf. FCS_CKM.5/ECKA-EG, calculates the MAC for the cipher text and compares it with the received MAC. If the integrity of the cipher text is determined, then the receiver decrypts the cipher text with the data decryption key, cf. FCS_COP.1/HDM.

In general, authentication is the provision of assurance of the claimed identity of an entity. The TOE authenticates human users by passwords, cf. FIA_UAU.5.1 clause 1 (1-Factor Authentication). But a human user may also authenticate himself to a token and the token authenticates to the TOE (2-Factor Authentication). Cryptographic authentication mechanisms allow an entity to prove its identity or the origin of its data to a verifying entity by demonstrating its knowledge of a secret. The entity authentication is required by FIA_UAU.5.1 clauses (2) to (6). Chapter 5.3 describes SFRs for the authentication of the TOE to external entities required by the SFR FIA_API.1. This authentication may include attestation of the TOE as a genuine TOE sample, cf. 6.1.4. The authentication may be mutual as required for trusted channels in chapter 6.1.5.

Protocols may use symmetric cryptographic algorithms, where the proving and the verifying entity using the same secret key may demonstrate that the proving entity belongs to a group of entities sharing this key, e.g. the sender and receiver (cf. FTP_ITC.1, FCS_COP.1/TCM). In case of asymmetric entity authentication mechanisms, the proving entity uses a private key, and the verifying entity uses the corresponding public key, where the latter is usually closely linked to the claimed identity by means of a certificate. Depending on the security attributes of the cryptographic keys - e.g. encoded in the certificate (cf. FPT_ISA.1/Cert) -, the same cryptographic mechanisms for digital signature generation (FCS_COP.1/CDS-*) and signature verification(cf. FCS_COP.1/VDS-*) may be used for entity authentication, data authentication and nonrepudiation as well.

A trusted channel requires mutual authentication of both endpoints with a key exchange of a key agreement, and the protection of confidentiality by encryption and cryptographic data integrity protection.

The TSF provide security management for user and TSF data, including cryptographic keys. Key management comprises administration and use of keying material in accordance with a security policy. This includes generation, derivation, registration, certification, deregistration, distribution, installation, storage, archival, revocation and destruction of keying material. The key management functionality of the TOE supports the generation, derivation, export, import, storage and destruction of cryptographic keys. The cryptographic keys are managed together with their security attributes.

The TOE enforces the *Key Management* SFP to protect all cryptographic keys (as data objects of TSF data) and key management services (as operation, cf. to SFR of the FMT class) provided for Administrators, Crypto-Officers, and Key Owners. Note that the cryptographic keys will be used for cryptographic operations under the Cryptographic Operation SFP as well.

Eliptic curve	Key size	Standard
brainpoolP256r1	*256 bits*	*RFC5639 [32], TR-03111, section 4.1.3 [3]*
brainpoolP384r1	*384 bits*	*RFC5639 [32], TR-03111, section 4.1.3 [3]*
brainpoolP512r1	*512 bits*	*RFC5639 [32], TR-03111, section 4.1.3 [3]*
Curve P-256	*256 bits*	*FIPS PUB 186-4 B.4 and D.1.2.3 [22]*
Curve P-384	*384 bits*	*FIPS PUB 186-4 B.4 and D.1.2.4 [22]*
Curve P-521	*521 bits*	*FIPS PUB 186-4 B.4 and D.1.2.5 [22]*

Table 6.1: Elliptic curves, key sizes and standards (Table 2 in Base-PP [9])

Name	IANA no.	Specified in
256-bit random ECP group	19	RFC5903 [33]
384-bit random ECP group	20	RFC5903 [33]
521-bit random ECP group	21	RFC5903 [33]
brainpoolP256r1	28	RFC6954 [34]
brainpoolP384r1	29	RFC6954 [34]
brainpoolP512r1	30	RFC6954 [34]

Table 6.2: Recommended groups for the Diffie-Hellman key exchange (Table 3 in Base-PP [9])

The subjects, objects and operations of the Update SFP are defined in the SFR FDP_ACC.1/UCP and FDP_ACF.1/UCP.

The SFRs for cryptographic mechanisms based on elliptic curves refer to Table 6.1 for selection of curves, key sizes and standards.

For Diffie-Hellman key exchange refer to the groups in Table 6.2

The PP-Module PPM-TS-Au [8] adds the following new SFRs concerning Time-Stamps and the Audit mechanism to the Base-PP: FAU_GEN.1, FAU_STG.1, FAU_STG.3, FDP_ACF.1/TS, FDP_DAU.2/TS, FDP_ETC.2/TS, FDP_ITC.2/TS, FMT_MTD.1/Audit, FMT_MOF.1/TSA, FMT_SMF.1/TSA, FMT_SMR.1/TSA, FPT_STM.1, FPT_TIT.1/Audit.

The PP-Module PPM-Cl [10] adds the following new SFRs concerning the cluster functionality to the Base-PP: FAU_GEN.1/CL, FCS_CKM.5/CLDH, FDP_ACC.1/CL, FMT_MTD.1/CL, FPT_ESA.1/CL,
FPT_ISA.1/CL, FPT_TCT.1/CL, FPT_TDC.1/CL, FPT_TIT.1/CL.

The TOE enforces the *Clustering SFP* for protection of the security attributes of the known users and the cryptographic keys with their security attributes

6.1.1 Key management

6.1.1.1 Management of security attributes

FDP_ACC.1/KM Subset access control - Cryptographic operation

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/KM	The TSF shall enforce the *Key Management SFP*[3] on

(1) *subjects:* Crypto-Officer[4]*, Key Owner;*

(2) *objects: operational cryptographic keys;*

(3) *operations: key generation, key derivation, key import, key export, key destruction*[5]

[3] [assignment: *access control SFP*]

[4] [selection (by ST author): Administrator, Crypto-Officer]

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

FMT_MSA.1/KM Management of security attributes - Key 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/KM The TSF shall enforce the *Key Management SFP and Cryptographic Operation SFP*[6] to restrict the ability to

(1) *set and change default values for*[7] the security attributes *Identity of the key, Key owner of the key, Key type, Key usage type, Key access control attributes, Key validity time period*[8] to **no role**,

(2) ***modify or delete***[9] **the security attributes *Identity of the key, Key owner, Key type, Key usage type, Key validity time period of an existing key***[10] **to *none***[11]**,**

(3) ***modify independent on key usage***[12] **the security attribute~~s~~ *Key usage counter of an existing key***[13] **to *none.***[14]

(4) ***modify***[15] **the security attribute~~s~~ *Key access control attribute of an existing key***[16] **to** Crypto-Officer[17],

(5) ***query***[18] **the security attributes *Key type, Key usage type, Key access control attributes, Key validity time period and Key usage counter of an identified key***[19] **to** Crypto-Officer, Key Owner[20].

***ST application note 1*: It should be noted that the refinement to allow the operation to no role in FMT_MSA.1.1/KM has the meaning that this operation is not allowed for anybody. With other words: this operation is not provided at all.**

Application note 1: The refinements repeats parts of the SFR component in order to avoid iteration of the component.

[6] [assignment: *access control SFP, information flow control SFP*]
[7] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[8] [assignment: *list of security attributes*]
[9] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[10] [assignment: *list of security attributes*]
[11] [assignment: *the authorised identified roles*]
[12] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[13] [assignment: *list of security attributes*]
[14] [assignment: *the authorised identified roles*]
[15] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[16] [assignment: *list of security attributes*]
[17] [selection (by ST author): Administrator, Crypto-Officer, Key Owner]
[18] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[19] [assignment: *list of security attributes*]
[20] [selection (by ST author): Administrator, Crypto-Officer, Key Owner]

FMT_MSA.3/KM Static attribute initialisation - Key management

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1/KM	The TSF shall enforce the *Key Management SFP, Cryptographic Operation SFP and Update SFP*[21] to provide *restrictive*[22] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/KM	The TSF shall allow the **no role** to specify alternative initial values to override the default values when a **cryptographic key** ~~object or information~~ is created.

***ST application note 2*: The TOE does not allow any role to specifiy alternate initial default values. This has been modelled by a refinement in FMT_MSA.3.2/KM and allowing the operation to no role. As the default values that are provided by the TOE are restrictive, this is considered to be more strict than the options that the PP provided originally.**

[21] [assignment: access control SFP, information flow control SFP]

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

FMT_MTD.1/KM Management of TSF data - Key management

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1/KM	The TSF shall restrict the ability to

(1) *create according to FCS_CKM.1*[23] the *cryptographic keys*[24] to Crypto-Officer[25],

(2) ***import according to FPT_TCT.1/CK, FPT_TIT.1/CK and FPT_ISA.1/CK***[26] **the *cryptographic keys***[27] **to** Crypto-Officer[28],

(3) ***export according to FPT_TCT.1/CK, FPT_TIT.1/CK and FPT_ESA.1/CK***[29] **the *cryptographic keys***[30] **to** Crypto-Officer[31] **if security attribute of the key allows export (keys with security attribute Key Usage Counter must never be exported),**

(4) ***delete according to FCS_CKM.4***[32] **the *cryptographic keys***[33] **to** Crypto-Officer[34].

Application note 2: The bullets (2) to (4) are refinements to avoid an iteration of component and therefore printed in bold.

[23] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[24] [assignment: *list of TSF data*]
[25] [selection (by ST author): Administrator, Crypto-Officer, Key Owner]
[26] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]
[27] [assignment: *list of TSF data*]
[28] [selection (by ST author): Administrator, Crypto-Officer, Key Owner]
[29] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]
[30] [assignment: *list of TSF data*]
[31] [selection (by ST author): Administrator, Crypto-Officer, Key Owner]
[32] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]
[33] [assignment: *list of TSF data*]
[34] [selection (by ST author): Administrator, Crypto-Officer, Key Owner]

6.1.1.2 Hash based functions

FCS_COP.1/Hash Cryptographic operation - Hash

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 generation*[35] in accordance with a specified cryptographic algorithm *SHA-256, SHA-384, SHA-512*[36] and cryptographic key sizes *none*[37] that meet the following: *FIPS 180-4 [26]*[38].

Application note 3: The hash function is a cryptographic primitive used for HMAC, cf. FCS_COP.1/HMAC, digital signature creation, cf. FCS_COP.1/CDS-*, digital signature verification, cf. FCS_COP.1/VDS-*, and key derivation, cf. FCS_CKM.5.

6.1.1.3 Management of Certificates

FMT_MTD.1/RK Management of TSF data - Root key

Hierarchical to:	No other components.
Dependencies	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1/RK	The TSF shall restrict the ability to (1) *create*[39]*, modify, clear and delete*[40] *the root key pair*[41] *to* Crypto-Officer[42], (2) ***import and delete***[43] ***a known as authentic public key of a certification authority in a PKI***[44] ***to*** Crypto-Officer[45]*,*

Application note 4: The root key is defined here with respect to the key hierarchy known to the TOE. In case of clause (1), i. e. may be a key pair of an TOE internal key hierarchy. In

[35] [assignment: *list of cryptographic operations*]
[36] [assignment: *cryptographic algorithm*]
[37] [assignment: *cryptographic key sizes*]
[38] [assignment: *list of standards*]
[39] "create" denotes initial setting a root key
[40] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[41] [assignment: *list of TSF data*]
[42] [selection (by ST author): Administrator, Crypto-Officer]
[43] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]
[44] [assignment: *list of TSF data*]
[45] [selection (by ST author): Administrator, Crypto-Officer]

clause (2) it may be a root public key of a PKI or a public key of another certification authority in a PKI known as being an authentic certificate signing key. The PKI may be used for user authentication, key management and signature-verification. The second bullet is a refinement to avoid an iteration of component and therefore printed in bold.

FPT_TIT.1/Cert TSF data integrity transfer protection - Certificates

Hierarchical to:	No other components.
Dependencies	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]
FPT_TIT.1.1/Cert	The TSF shall enforce the *Key Management SFP*[46] to *receive*[47] a **certificate** ~~TSF data~~ in a manner protected from *modification and insertion*[48] errors.
FPT_TIT.1.2/Cert	The TSF shall be able to determine on receipt of a **certificate** ~~TSF data~~, whether *modification and insertion*[49] has occurred.

[46] [assignment: *access control SFP, information flow control SFP*]
[47] [selection: *transmit, receive, transmit and receive*]
[48] [selection: *modification, deletion, insertion, replay*]
[49] [selection: *modification, deletion, insertion, replay*]

FPT_ISA.1/Cert Import of TSF data with security attributes - Certificates

Hierarchical to: No other components.

Dependencies [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FMT_MTD.1 Management of TSF data or
FMT_MTD.3 Secure TSF data]
[FMT_MSA.1 Management of security attributes, or
FMT_MSA.4 Security attribute value inheritance]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_ISA.1.1/Cert The TSF shall enforce the *Key management SFP*[50] when importing **certificates** ~~TSF data~~, controlled under the SFP, from outside of the TOE.

FPT_ISA.1.2/Cert The TSF shall use the security attributes associated with the imported **certificate** ~~TSF data~~.

FPT_ISA.1.3/Cert The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the **certificates** ~~TSF data~~ received.

FPT_ISA.1.4/Cert The TSF shall ensure that **the** interpretation of the security attributes of the imported **certificates** ~~TSF data~~ is as intended by the source of the **certificates** ~~TSF data~~.

FPT_ISA.1.5/Cert The TSF shall enforce the following rules when importing **certificates** ~~TSF data~~ controlled under the SFP from outside the TOE:

(1) *The TSF imports the TSF data in certificates only after successful verification of the validity of the certificate in the certificate chain until it is known as an authentic certificate according to FMT_MTD.1/RK.*

(2) *The validity verification of the certificate shall include*

   (a) *except for root certificates, the verification of the digital signature of the certificate issuer and*

   (b) *a verification that the security attributes in the certificate pass the interpretation according to FPT_TDC.1*[51]

[50] [assignment: *access control SFP, information flow control SFP*]
[51] [assignment: *additional importation control rules*]

FPT_TDC.1/Cert Inter-TSF basic TSF data consistency - Certificate

Hierarchical to:	No other components.
Dependencies	No dependencies.
FPT_TDC.1.1/Cert	The TSF shall provide the capability to consistently interpret *security attributes of cryptographic keys in the certificate and the identity of the certificate issuer*[52] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2/Cert	The TSF shall use **the following rules:** (1) *the TOE reports about conflicts between the Key identities of stored cryptographic keys and cryptographic keys to be imported,* (2) *the TOE does not change the security attributes Key identity, Key owner, Key type, Key usage type and Key validity time period of a public key that is imported from the certificate,* (3) *the identity of the certificate issuer shall meet the identity of the signer of the certificate*[53] when interpreting **the certificate from a trust centre** ~~TSF data from another trusted IT product~~.

Application note 5: The security attributes assigned to a certificate holder and the cryptographic key in the certificate are used as TSF data of the TOE. The certificate is imported from a trust centre directory service, but must be verified by the TSF (i.e. if it is verified successfully that the source is the trust centre's directory server of the trusted IT product).

6.1.1.4 Key generation, agreement and destruction

Key generation (cf. FCS_CKM.1/ECC, FCS_CKM.1/RSA) is a randomized process which uses random secrets (cf. FCS_RNG.1), applies key generation algorithms and defines security attributes depending on the intended use of the keys. It has the property that it is computationally infeasible to deduce the output without prior knowledge of the secret input. *Key derivation* (cf. FCS_CKM.5/ECC) is a deterministic process by which one or more keys are calculated from a pre-shared key or shared secret or other information. It allows repeating the key generation if the same input is provided. *Key agreement* (cf. FCS_CKM.5/ECDHE) is a key-establishment procedure process for establishing a shared secret key between entities in such a way that neither of them can predetermine the value of that key independently of the other party's contribution. Key agreement allows each participant to enforce the cryptographic quality of the agreed key. The component FCS_CKM.1 was refined for key agreement because it normally uses random bits as input. Hybrid cryptosystems (FCS_CKM.1/ECKA-EG, FCS_CKM.1/AES_RSA) are a combination of a public key cryptosystem with an efficient symmetric key cryptosystem.

The user may need to specify the type of key, the cryptographic key generation algorithm, the security attributes and other necessary parameters.

[52] [assignment: *list of TSF data types*]
[53] [assignment: *list of interpretation rules to be applied by the TSF*]

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies No dependencies.

FCS_RNG.1.1/Cert The TSF shall provide a deterministic[54] random number generator that implements: *DRG.3.1 If initialized with a random seed using a PTRNG (NeuG of PrimeKey SEE hardware) the internal state of the RNG shall have at least 125*[55] *bits of entropy. DRG.3.2 The RNG provides forward secrecy. DRG.3.3 The RNG provides backward secrecy even if the current internal state is known* [56].

FCS_RNG.1.2/Cert The TSF shall provide random numbers that meet *(DRG.3.4) The RNG gets initialized with a random seed during every start-up of the TOE and generates output for which* $\mathbf{2^{14}}$ *strings of bit length 128 are mutually different with probability* $\mathbf{1-2^{-8}}$. *(DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A* [57].

Application note 6: The random bit generation shall be used for key generation and key agreement according to all instantiations of FCS_CKM.1, challenges in cryptographic protocols and cryptographic operations using random values according to FCS_COP.1/HEM and FCS_COP.1/TCE. The TOE also provides the random number generation as security service for the user.

FCS_CKM.1/AES Cryptographic key generation - AES key

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 **AES** keys in accordance with a specified cryptographic key generation algorithm *AES*[58] and specified cryptographic key sizes 128 bits, 256 bits[59] that meet the following: *ISO 18033-3 [19]*[60].

Application note 7: The cryptographic key(s) may be also used together with FCS_COP.1/ED, e.g. for internal purposes.

[54] [selection (by ST author): physical, non-physical true, deterministic, hybrid physical, hybrid deterministic]

[55] 256 bit of random bit are obtained from the PTRNG to meet the 125 bit entropy

[56] [assignment (by ST author): *list of security capabilities*]

[57] [assignment (by ST author): *a defined quality metric*]

[58] [assignment: *cryptographic key generation algorithm*]

[59] [selection (by ST author): 256 bits, [assignment: additional cryptographic key sizes > 128 bits]]

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

FCS_CKM.5/AES Cryptographic key derivation - AES key derivation

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.5.1/AES The TSF shall derive cryptographic *AES keys*[61] from *random input parameters*[62] in accordance with a specified cryptographic key derivation algorithms *AES key generation using a bit string derived from input parameters with a KDF*[63] and specified cryptographic key sizes *128 bits,* 256 bits[64] that meet the following: *NIST SP800-56C [25]*[65].

FCS_CKM.1/ECC Cryptographic key generation - Elliptic curve key pair ECC

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 **elliptic curve** key~~s~~ **pairs** in accordance with a specified cryptographic key generation algorithm *ECC key pair generation with* Curve P-256[66] and specified cryptographic key sizes 256 bits[67] that meet the following: FIPS PUB 186-4 B.4 and D.1.2.3[68].

Application note 8: The elliptic key pair generation uses a random bit string as input for the ECC key generation algorithm. The keys generation according to FCS_CKM.1/ECC and key derivation according to FCS_CKM.5/ECC are intended for different key management use cases but the keys itself may be used for same cryptographic operations.

***ST application note 3*: The selections in FCS_CKM.1/ECC refer to Table 2 in the Base-PP [9] and to Table 6.1 in this ST document.**

[61] [assignment: *key type*]
[62] [assignment (by ST author): *input parameters*]
[63] [assignment: *cryptographic key derivation algorithm*]
[64] [selection (by ST author): 256 bits, [assignment: additional cryptographic key sizes > 128 bits]]
[65] [assignment: *list of standards*]
[66] [selection (by ST author): elliptic curves in table 2]
[67] [selection (by ST author): key size in table 2]
[68] [selection (by ST author): standards in table 2]

FCS_CKM.5/ECC Cryptographic key derivation - ECC key pair derivation

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.5.1/ECC The TSF shall derive cryptographic *elliptic curve* key~~s~~ pairs[69] from *seed provided by external entity*[70] in accordance with a specified cryptographic key derivation algorithm *ECC key pair generation with* Curve P-256[71] *using bit string derived from input parameters with X9.63 Key Derivation Function [3] (Section 4.3.3)*[72] and specified cryptographic key sizes 256 bits[73] that meet the following: FIPS PUB 186-4 B.4 and D.1.2.3[74], *TR-03111 [3].*

Application note 9: The elliptic key pair derivation applies a key derivation function (KDF), e.g. from *[3] (Section 4.3.3.)* to the input parameter. It uses the output string of a KDF instead of the random bit string as input for the ECC key generation algorithm (*[3], Section 4.1.1, Algorithms 1 or 2*). The input parameters shall include a secret of the length of at least of the key size to ensure the confidentiality of the private key. The input parameters may include public known values or even values provided by external entities.

***ST application note 4*: The selections in FCS_CKM.5/ECC refer to Table 2 in the Base-PP [9] and to Table 6.1 in this ST document.**

FCS_CKM.1/RSA Cryptographic key generation - RSA key pair

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/RSA The TSF shall generate cryptographic **RSA** key **pairs** in accordance with a specified cryptographic key generation algorithm *RSA*[75]and specified cryptographic key sizes *4096 bits*[76] that meet the following: *PKCS #1 v2.2 [36]*[77].

Application note 10: The cryptographic key sizes assigned in FCS_CKM.1/RSA must be at least 2000 bits. Cryptographic key sizes of at least 3000 bits are recommended. The SFR

[69] [assignment: *key type*]
[70] [assignment (by ST author): *input parameters*]
[71] [selection (by ST author): elliptic curves in table 2]
[72] [assignment (by ST author): *KDF*]
[73] [selection (by ST author): key size in table 2 [assignment: *cryptographic key sizes*]]
[74] [selection (by ST author): standards in table 2, [3] [assignment: *list of standards*]]
[75] [assignment: *cryptographic key generation algorithm*]
[76] [assignment (by ST author): *cryptographic key sizes*]
[77] [assignment: *list of standards*]

FCS_CKM.1/RSA assigns given security attributes *Key identity* and *Key owner*.

FCS_CKM.5/ECDHE Cryptographic key derivation - Elliptic Curve Diffie-Hellman ephemeral key agreement

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.5.1/ECDHE	The TSF shall derive cryptographic *ephemeral* keys[78] **for data encryption and MAC with AES-128,** AES-256[79] from *an agreed shared secret*[80] in accordance with a specified cryptographic key derivation algorithm *Elliptic Curve Diffie-Hellman ephemeral key agreement* Curve P-256[81] *and* 256-bit random ECP group[82] *with a key derivation from the shared secret SHA-1 for AES-128, SHA-256 for AES-256*[83] and specified cryptographic key sizes *128 bits* or 256 bits[84] that meet the following: *TR-03111 [3]*[85].

Application note 11: The input parameters for key derivation is an agreed shared secret established by means of Elliptic Curve Diffie-Hellman. Table 2 lists elliptic curves and table 3 lists Diffie-Hellman Groups for the agreement of the shared secret. SHA-1 shall be supported for generation of 128 bits AES keys. SHA-256 shall be selected and used to generate 256 bits AES keys.

***ST application note 5*: The selections in FCS_CKM.5/ECDHE refer to Table 2 / Table 3 in the Base-PP [9] and to Table 6.1 / Table 6.2 in this ST document.**

[78] [assignment: *key type*]
[79] [selection (by ST author): AES-256, none other]
[80] [assignment: *input parameters*]
[81] [selection (by ST author): elliptic curves in table 2]
[82] [selection (by ST author): DH group in table 3]
[83] [assignment (by ST author): *key derivation function*]
[84] [selection (by ST author): 256 bits, none other]
[85] [assignment: *list of standards*]

FCS_CKM.1/ECKA-EG Cryptographic key generation - ECKA-EG key generation with ECC encryption

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/ECKA-EG The TSF shall generate **ephemeral** cryptographic **elliptic curve** key **pairs for ECKGA-EG** *[3], sender role*) in accordance with a specified cryptographic key generation algorithm *ECC key pair generation with* Curve P-256[86] and specified cryptographic key sizes 256 bits[87] that meet the following: FIPS PUB 186-4 B.4 and D.1.2.3[88].

***ST application note 6*: The selections in FCS_CKM.1/ECKA-EG refer to Table 2 in the Base-PP [9] and to Table 6.1 in this ST document.**

FCS_CKM.5/ECKA-EG Cryptographic key derivation - ECKA-EG key derivation

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.5.1/ECKA-EG The TSF shall derive cryptographic *data encryption and MAC* keys for *AES-128,* AES-256[89] from a private and a *public ECC key*[90] in accordance with a specified cryptographic key derivation algorithm *ECKGA-EG [3]* Curve P-256[91] *and X9.63 Key Derivation Function* and specified cryptographic **symmetric** key sizes *128 bits* or 256 bits[92] that meet the following: *TR03111 [3], chapter 4.3.2.2*[93].

Application note 12: FCS_CKM.5/ECKA-EG is used by both the sender (encryption) and the recipient (decryption) to compute a secret point S_{AB} on an elliptic curve and derived a shared secret Z_{AB}. The shared secret is then used as the input to the key derivation function to derive two symmetric keys: the encryption key and the MAC key. These are then used to encrypt or decrypt messages according to FCS_COP.1/HEM or FCS_COP.1/HDM, respectively. Sender and recipient use however different inputs to FCS_CKM.5/ECKA-EG. The sender first generates an ephemeral ECC key pair according to FCS_CKM.1/ECKA-EG and uses the generated

[86] [selection (by ST author): elliptic curves in table 2]
[87] [selection (by ST author): key size in table 2]
[88] [selection (by ST author): standards in table 2]
[89] [selection (by ST author): AES-256, none other]
[90] [assignment: *input parameters*]
[91] [selection (by ST author): elliptic curves in table 2]
[92] [selection (by ST author): 256 bits, none other]
[93] [assignment: *list of standards*]

ephemeral private key and the static public key of the recipient as input. The recipient first extracts the ephemeral public key from the message and uses the ephemeral public key and the static private key (cf. FCS_CKM.1/ECC for key generation) as the input to derive the symmetric keys. The selection of the elliptic curve, the ECC key size and length of the shared secret shall correspond to the selection of the AES key size, e. g. brainpoolP256r1 and 256 bits seed for ECC key and AES keys. FCS_CKM.1/ECKA-EG and FCS_CKM.5/ECKA-EG do not provide self-contained security services for the user but are necessary steps for FCS_COP.1/HEM and FCS_COP.1/HDM (refer to the next section 6.1.3).

***ST application note 7*: The selections in FCS_CKM.5/ECKA-EG refer to Table 2 in the Base-PP [9] and to Table 6.1 in this ST document.**

FCS_CKM.1/AES_RSA Cryptographic key generation - Key generation and RSA encryption

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_RSA	The TSF shall generate **and encrypt a seed, derive** cryptographic keys **from the seed for data encryption and MAC with AES-128,** AES-256[94] in accordance with a specified cryptographic key generation algorithm *X9.63 Key Derivation Function [1] and RSA EME-OAEP [36]*[95] and specified cryptographic **symmetric** key sizes *128 bits* 256 bits[96] that meet the following: *ISO/IEC18033-3 [19], PKCS #1 v2.2 [36]*[97].

Application note 13: The asymmetric cryptographic key sizes used in FCS_CKM.1/AES_RSA must be at least 2000 bits. Cryptographic key sizes of at least 3000 bits are recommended. FCS_CKM.1/AES_RSA and FCS_CKM.5/AES_RSA do not provide self-contained security services for the user but they are only necessary steps for FCS_COP.1/HEM respective FCS_COP.1/HDM (refer to the next section 6.1.3).

[94] [selection (by ST author): *AES-256, none other*]
[95] [assignment: *cryptographic key generation algorithm*]
[96] [selection (by ST author): 256 bits, none other]
[97] [assignment: *list of standards*]

FCS_CKM.5/AES_RSA Cryptographic key derivation - RSA key derivation and decryption

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.5.1/AES_RSA The TSF shall derive cryptographic *data encryption keys and MAC keys for AES-128,* AES-256[98] from a **decrypted** *RSA encrypted seed*[99] in accordance with a specified cryptographic key derivation algorithm *RSA EME-OAEP [36] and X9.63 [1] Key Derivation Function*[100] and specified cryptographic **symmetric** key sizes 128 bits 256 bits[101] that meet the following: *ISO/IEC 14888-2 [18]*[102].

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 by overwriting with zeros, random values or a new key*[103] that meets the following: *none*[104].

Refinement: The destruction of cryptographic keys shall ensure that any previous information content of the resource about the key is made unavailable upon the deallocation of the resource.

[98] [selection (by ST author): AES-256, none other]
[99] [assignment: *input parameters*]
[100] [assignment: *cryptographic key derivation algorithm*]
[101] [selection (by ST author): 256 bits, none other]
[102] [assignment: *list of standards*]
[103] [assignment (by ST author): *cryptographic key destruction method*]
[104] [assignment (by ST author): *list of standards*]

6.1.1.5 Key import and export

FCS_COP.1/KW Cryptographic operation - Key wrap

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, FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/KW	The TSF shall perform *key wrap*[105] in accordance with a specified cryptographic algorithm *AES-Keywrap* KWP[106] and cryptographic key sizes **of the key encryption key** *128 bits* 256 bits[107] that meet the following: *NIST-SP800-38F [30]*[108].

Application note 14: The selection of the length of the key encryption key shall be equal or greater than the security bits of the wrapped key for its cryptographic algorithm.

FCS_COP.1/KU Cryptographic operation - Key unwrap

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/KU	The TSF shall perform *key unwrap*[109] in accordance with a specified cryptographic algorithm *AES-Keywrap* KWP[110] and cryptographic key sizes **of the key encryption key** *128 bits* 256 bits[111] that meet the following: *NIST SP800-38F [30]*[112].

[105] [assignment: *list of cryptographic operations*]
[106] [selection (by ST author): KW, KWP]
[107] [selection (by ST author): 256 bits, none other]
[108] [assignment: *list of standards*]
[109] [assignment: *list of cryptographic operations*]
[110] [selection (by ST author): KW, KWP]
[111] [selection (by ST author): 256 bits, none other]
[112] [assignment: *list of standards*]

FPT_TCT.1/CK TSF data confidentiality transfer protection - Cryptographic keys

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]
FPT_TCT.1.1/CK	The TSF shall enforce the *Key Management SFP*[113] by providing the ability to *transmit and receive*[114] **a cryptographic key** ~~TSF data~~ in a manner protected from unauthorised disclosure **according to FCS_COP.1/KW and FCS_COP.1/KU**.

FPT_TIT.1/CK TSF data integrity transfer protection - Cryptographic keys

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]
FPT_TIT.1.1/CK	The TSF shall enforce the *Key Management SFP*[115] by providing the ability to *transmit and receive*[116] **cryptographic keys** ~~TSF data~~ in a manner protected from *modification and insertion*[117] errors **according to FCS_COP.1/KW**.
FPT_TIT.1.2/CK	The TSF shall be able to determine on receipt of **cryptographic keys** ~~TSF data~~, whether *modification and insertion*[118] has occurred **according to FCS_COP.1/KU**.

[113] [assignment: *access control SFP, information flow control SFP*]
[114] [selection: *transmit, receive, transmit and receive*]
[115] [assignment: *access control SFP, information flow control SFP*]
[116] [selection: *transmit, receive, transmit and receive*]
[117] [selection: *modification, deletion, insertion, replay*]
[118] [selection: *modification, deletion, insertion, replay*]

FPT_ISA.1/CK Import of TSF data with security attributes - Cryptographic keys

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FMT_MTD.1 Management of TSF data or
FMT_MTD.3 Secure TSF data]
[FMT_MSA.1 Management of security attributes, or
FMT_MSA.4 Security attribute value inheritance]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_ISA.1.1/CK The TSF shall enforce the *Key Management SFP*[119] when importing **cryptographic keys** ~~TSF data~~, controlled under the SFP, from outside of the TOE.

FPT_ISA.1.2/CK The TSF shall use the security attributes associated with the imported **cryptographic keys** ~~TSF data~~

FPT_ISA.1.3/CK The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the **cryptographic key** ~~TSF data~~ received.

FPT_ISA.1.4/CK The TSF shall ensure that interpretation of the security attributes of the imported **cryptographic key** ~~TSF data~~ is as intended by the source of the **cryptographic key** ~~TSF data~~.

FPT_ISA.1.5/CK The TSF shall enforce the following rules when import **cryptographic key** ~~TSF data~~. controlled under the SFP from outside the TOE:

(1) *The TSF imports the TSF data in certificates only after successful verification of the validity of the certificate including the verification of the digital signature of the issuer and the validity time period.*

(2) *no additional importation control rules*[120]

Application note 15: The operational environment is obligated to use trust centre services for secure key management, cf. OE.SecManag.

[119] [assignment: *access control SFP, information flow control SFP*]
[120] [assignment (by ST author): *additional importation control rules*]

FPT_TDC.1/CK Inter-TSF basic TSF data consistency - Key import

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1/CK The TSF shall provide the capability to consistently interpret *security attributes of the imported cryptographic keys*[121] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2/CK The TSF shall use the **following rules**

(1) *the TOE reports about conflicts between the Key identity of stored cryptographic keys and cryptographic keys to be imported,*

(2) *the TOE does not change the security attributes Key identity, Key type, Key usage type and Key validity time period of the key being imported*[122]

when interpreting **the imported key data object** ~~TSF data from another trusted IT product~~.

[121] [assignment: *list of TSF data types*]

[122] [assignment: *list of interpretation rules to be applied by the TSF*]

FPT_ESA.1/CK Export of TSF data with security attributes - Cryptographic keys

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data] [FMT_MSA.1 Management of security attributes, or FMT_MSA.4 Security attribute value inheritance] FPT_TDC.1 Inter-TSF basic TSF data consistency
FPT_ESA.1.1/CK	The TSF shall enforce the *Key Management SFP*[123] when importing **cryptographic keys** ~~TSF data~~, controlled under the SFP(s), from outside of the TOE.
FPT_ESA.1.2/CK	The TSF shall export the **cryptographic key** ~~TSF data~~ with the **cryptographic key's** ~~TSF data~~ associated security attributes.
FPT_ESA.1.3/CK	The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported **cryptographic key** ~~TSF data~~.
FPT_ESA.1.4/CK	The TSF shall enforce the following rules when **cryptographic key** ~~TSF data~~ is exported from the TOE: *For keys with the security attribute "Key Usage Counter", the TSF must ensure that decreasing the counter importing an older version of the key is impossible. Additionally there are no other exportation control rules.*[124]

Application note 16: There are no fixed rules for presentation of security attributes defined. The element FPT_ESA.1.4/CK must define rules expected in FPT_TDC.1 Inter-TSF basic TSF data consistency if inter-TSF key exchange is intended.

W.r.t. to FPT_ESA.1.4/CK note the following naive attack: 1) A user exports a key having the attribute "Key Usage Counter". 2) The key is then re-imported and used several times. 3) The key is exported again and 4) the exported version of 1) instead of the one of 3.) is re-imported, thus effectively decreasing the attribute "Key Usage Counter". A straight-forward way to counter this is to prohibit keys with the attribute "Key Usage Counter" from being exported.

[123] [assignment: *access control SFP, information flow control SFP*]
[124] [assignment (by ST author): *additional exportation control rules*]

6.1.2 Data encryption

FCS_COP.1/ED Cryptographic operation - Data encryption and decryption

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/ED	The TSF shall perform *data encryption and decryption*[125] in accordance with a specified cryptographic algorithm *symmetric data encryption according to AES-128 and* AES-256[126] *in CBC and* no other[127] *mode* and cryptographic key size *128 bits,* 256 bits[128] that meet the following: *IST-SP800-38A [27], ISO 18033-3 [19], ISO 10116 [17]*[129].

Application note 17: Data encryption and decryption should be combined with data integrity mechanisms in Encrypt-then-MAC order, i. e. the MAC is calculated over the ciphertext and verified before decryption. The modes of operation should combine encryption with data integrity mechanisms into authenticated encryption, e. g. Cipher Block Chaining Mode (CBC, cf. NIST SP800-38A) should be combined with CMAC (cf. FCS_COP.1/MAC) or HMAC (cf. FCS_COP.1/HMAC). For combination of symmetric encryption, decryption and data integrity mechanisms by means of CCM or GCM refer to the next section 6.1.3.

***ST application note 8*: *CRT* (a typo in the PP) was corrected to *CTR*, short for *Counter Mode.* Please note that this Application Note does not contain any relevant information due to the operations performed in the previous SFR. It has been kept for completeness.**

[125] [assignment: *list of cryptographic operations*]
[126] [selection (by ST author): AES-256, no other algorithm]
[127] [selection (by ST author): CTR, OFB, CFB, no other]
[128] [selection (by ST author): 256 bits, no other key size]
[129] [assignment: *list of standards*]

6.1.3 Hybrid encryption with MAC for user data

FCS_COP.1/HEM Cryptographic operation - Hybrid data encryption and MAC calculation

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/HEM	The TSF shall perform *hybrid data encryption and MAC calculation*[130] in accordance with a specified cryptographic algorithm *asymmetric key encryption according to* FCS_CKM.5/ECDHE[131], *symmetric data encryption according to AES-128,* AES-256[132] *[23] in* CBC [27][133] *mode with* CMAC [28][134] *calculation* and cryptographic **symmetric** key sizes *128 bits,* 256 bits[135] that meet the following: *the referenced standards above according to the chosen selection*[136].

Application note 18: Hybrid data encryption and MAC calculation is a self-contained security service of the TOE. The generation and encryption of the seed, derivation of encryption and MAC keys as well as AES encryption and MAC calculation are only steps of this service. Hybrid encryption is combined with MACs as data integrity mechanisms for the cipher text, i. e. encrypt-then-MAC creation for CMAC. section 6.1.3.

[130] [assignment: *list of cryptographic operations*]
[131] [selection (by ST author): FCS_CKM.1/ECKA-EG, FCS_CKM.1/AES_RSA, FCS_CKM.5/ECDHE]
[132] [selection (by ST author): AES-256, none other]
[133] [selection (by ST author): CBC [27], CCM [24], GCM [29]]
[134] [selection (by ST author): CMAC [28], GMAC [29], HMAC [31]]
[135] [selection (by ST author): 256 bits, no other key size]
[136] [assignment: *list of standards*]

FCS_COP.1/HDM Cryptographic operation - Hybrid data decryption and MAC verification

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/HDM	The TSF shall perform *hybrid MAC verification and data decryption*[137] in accordance with a specified cryptographic algorithm *asymmetric key decryption according to* FCS_CKM.5/ECDHE[138], *verification of* CMAC [28][139] *and symmetric data decryption according to AES with* AES-128, AES-256[140] *[23] in mode* CBC [27][141] and cryptographic **symmetric** key sizes *128 bits,* 256 bits[142] that meet the following: *the referenced standards above according to the chosen selection*[143].

Application note 19: Hybrid data decryption and MAC verification is a self-contained security service of the TOE. The decryption of the seed and derivation of the encryption key and MAC key as well as the AES decryption and MAC verification are only steps of this service. The used symmetric key shall fit to the AES CMAC or GMAC and the AES algorithm for decryption of the cipher text for MAC, e. g. verification-thendecrypt for CMAC.

***ST application note 9*: In the selections for *verification* and *symmetric data decryption* in FCS_COP.1/HDM, the PP mixed up *GCM* and *GMAC*. Those were fixed in the ST's selections to their appropriate algorithm type.**

6.1.4 Data integrity mechanisms

Cryptographic data integrity mechanisms comprise two types of mechanisms - symmetric message authentication code mechanisms and asymmetric digital signature mechanisms. A message authentication code mechanism comprises the generation of a MAC for the original message, the verification of a given pair of a message and MAC, and management of the underlying symmetric key(s). The MAC may be applied to a plaintext without encryption, but when combined with encryption it should be applied to ciphertexts in Encrypt-then-MAC order.

[137] [assignment: *list of cryptographic operations*]
[138] [selection (by ST author): FCS_CKM.5/ECDHE, FCS_CKM.5/ECKA-EG, FCS_CKM.5/AES_RSA]
[139] [selection (by ST author): CMAC [28], GCM [29], HMAC [31]]
[140] [selection (by ST author): AES-128, AES-256]
[141] [selection (by ST author): CBC [27], CCM [24], GMAC [29]]
[142] [selection (by ST author): 256 bits, no other key size]
[143] [assignment: *list of standards*]

FCS_COP.1/MAC Cryptographic operation - MAC using AES

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 destructio
FCS_COP.1.1/MAC	The TSF shall perform *MAC generation and verification*[144] in accordance with a specified cryptographic algorithm *AES-128 and* AES-256[145] *[23], CMAC [28] and* no other[146] and cryptographic key sizes *128 bits,* 256 bits[147] that meet the following: *the referenced standards above according to the chosen selection*[148].

Application note 20: The MAC may be applied to plaintexts and cipher texts. The algorithm AES-128 CMAC is mandatory.

FCS_COP.1/HMAC Cryptographic operation - HMAC

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/HMAC	The TSF shall perform *HMAC generation and verification*[149] in accordance with a specified cryptographic algorithm *HMAC-SHA256 and* no other[150] and cryptographic key sizes *128 bits and above*[151] that meet the following: *RFC2104 [31] , ISO 9797-2 [20]*[152].

Application note 21: The cryptographic key is a random bit string generated by FCS_RNG.1 or a referenced internal secret. The cryptographic key sizes assigned in FCS_COP.1/HMAC must be at least 128 bits.

[144] [assignment: *list of cryptographic operations*]

[145] [selection (by ST author): AES-256, none other]

[146] [selection (by ST author): GMAC [29], no other]

[147] [selection (by ST author): 256 bits, no other key size]

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

[149] [assignment: *list of cryptographic operations*]

[150] [selection (by ST author): HMAC-SHA-1, HMACSHA384, no other]

[151] [assignment (by ST author): *cryptographic key sizes*]

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

FCS_COP.1/CDS-ECDSA Cryptographic operation - Creation of digital signatures ECDSA

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/CDS-ECDSA	The TSF shall perform *signature-creation*[153] in accordance with a specified cryptographic algorithm *ECDSA with* Curve P-256[154] and cryptographic key sizes 256 bits[155] that meet the following: FIPS PUB 186-4 B.4 and D.1.2.3[156].

***ST application note 10*: The selections in FCS_COP.1/CDS-ECDSA refer to Table 2 in the Base-PP [9] and to Table 6.1 in this ST document.**

FCS_COP.1/VDS-ECDSA Cryptographic operation - Verification of digital signatures ECDSA

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/VDS-ECDSA	The TSF shall perform *signature-verification*[157] in accordance with a specified cryptographic algorithm *ECDSA with* Curve P-256[158] and cryptographic key sizes 256 bits[159] that meet the following: FIPS PUB 186-4 B.4 and D.1.2.3[160].

***ST application note 11*: The selections in FCS_COP.1/VDS-ECDSA refer to Table 2 in the Base-PP [9] and to Table 6.1 in this ST document.**

[153] [assignment: *list of cryptographic operations*]
[154] [selection (by ST author): elliptic curves in table 2]
[155] [selection (by ST author): key size in table 2]
[156] [selection (by ST author): standards in table 2]
[157] [assignment: *list of cryptographic operations*]
[158] [selection (by ST author): elliptic curves in table 2]
[159] [selection (by ST author): key size in table 2]
[160] [selection (by ST author): standards in table 2]

FCS_COP.1/CDS-RSA Cryptographic operation - Creation of digital signatures RSA

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/CDS-RSA The TSF shall perform *signature-creation*[161] in accordance with a specified cryptographic algorithm *RSA and EMSA-PSS*[162] and cryptographic key sizes *4096 bits*[163] that meet the following: *ISO/IEC 14888-2 [18], PKCS #1, v2.2 [36]*[164].

FCS_COP.1/VDS-RSA Cryptographic operation - Verification of digital signatures RSA

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/VDS-RSA The TSF shall perform *signature-verification*[165] in accordance with a specified cryptographic algorithm *RSA and EMSA-PSS*[166] and cryptographic key sizes *4096 bits*[167] that meet the following: *ISO/IEC 14888-2 [18], PKCS #1, v2.2 [36]*[168].

[161] [assignment: *list of cryptographic operations*]
[162] [assignment: *cryptographic algorithm*]
[163] [assignment (by ST author): *cryptographic key sizes*]
[164] [assignment: *list of standards*]
[165] [assignment: *list of cryptographic operations*]
[166] [assignment: *cryptographic algorithm*]
[167] [assignment (by ST author): *cryptographic key sizes*]
[168] [assignment: *list of standards*]

FDP_DAU.2/Sig Data Authentication with Identity of Guarantor - Signature

Hierarchical to:	FDP_DAU.1 Basic Data Authentication
Dependencies:	FIA_UID.1 Timing of identification
FDP_DAU.2.1/Sig	The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of *user data*[169] **imported according to FDP_ITC.2/UD by means of** FCS_COP.1/CDS-ECDSA[170] **and keys holding the security attribute Key identity assigned to the guarantor and Key usage type "digitalSignature".**
FDP_DAU.2.2/Sig	The TSF shall provide *external entities*[171] with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence.

Application note 22: The TSF according to FDP_DAU.2/Sig is intended for a signature service for user data. The user data source shall select the security attributes *Key owner* of the guarantor and *Key usage type "digitalSignature"* of the cryptographic key for the signature service in the security attributes provided with the user data. The user data source subject shall meet the *Key access control attributes* for the signature creation operation. The verification of the evidence requires a certificate showing the identity of the key owner.

[169] [assignment: *list of objects or information types*]
[170] [selection (by ST author): FCS_COP.1/CDS-RSA, FCS_COP.1/CDS-ECDSA]
[171] [assignment: *list of subjects*]

6.1.5 Time Stamp

FDP_DAU.2/TS Data Authentication with Identity of Guarantor - Signature with time stamp and optional key usage counter

Hierarchical to: FDP_DAU.1 Basic Data Authentication

Dependencies FIA_UID.1 Timing of identification

FDP_DAU.2.1/TS The TSF shall provide a capability to generate evidence that can be used as a guarantee of the **existence at certain point in time, sequence and** validity of

(a) *user data imported according to FDP_ITC.2/UD*

(b) *exported audit records according to FMT_MTD.1/Audit clause (1) and FAU_STG.3 clause (1)*[172]

with

(1) **time stamp of the evidence generation according to FPT_STM.1,**

(2) **and optionally the key usage counter of the signature key by means of digital signature generated according to** FCS_COP.1/CDS-ECDSA[173] **and keys holding the dedicated values of the security attributes Key identity that indicate key ownership of the TOE sample and Key usage type "Time stamp service".**

FDP_DAU.2.2/TS The TSF shall provide *external entities*[174] with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence.

Application note 1 of [8]: The TSF according to FDP_DAU.2/TS is intended for time stamp service of the TOE for any provided user data and exported audit records. The user data source shall select the security attribute Key usage type "TimeStamp" of the signature key of the time stamp service. The signature key of exported audit records shall be defined according to FMT_MOF.1.1/TSA clause (5). The Key usage counter allows to verify the sequence of signed data e. g. in an audit trail. The verification of the evidence requires a certificate showing the identity of the TOE sample and the key usage type of time stamp service. The format of input data and output data shall meet the BSI TR-03151 [12] .

[172] [assignment: *list of objects or information types*]

[173] [selection (by ST author): FCS_COP.1/CDS-ECDSA, FCS_COP.1/CDS-RSA]

[174] [assignment (by ST author): *list of subjects*]

6.1.6 Authentication and attestation of the TOE, trusted channel

FIA_API.1/PACE Authentication Proof of Identity - PACE authentication to Application component

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1/PACE The TSF shall provide *PACE in ICC role*[175] to prove the identity of the *TOE*[176] to an external entity **and to establish a trusted channel according to FTP_ITC.1 case 1 or 2.**

FIA_API.1/CA Authentication Proof of Identity - Chip authentication to user

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1/CA The TSF shall provide *Chip Authentication Version 2 according to [4] section 3.4*[177] to prove the identity of the *TOE*[178] to an external entity **and to establish a trusted channel according to FTP_ITC.1 case 3.**

[175] [assignment: *authentication mechanism*]
[176] [assignment: *object, authorized user or role*]
[177] [assignment: *authentication mechanism*]
[178] [assignment: *object, authorized user or role*]

FDP_DAU.2/Att Data Authentication with Identity of Guarantor - Attestation

Hierarchical to:	FDP_DAU.1 Basic Data Authentication
Dependencies:	FIA_UID.1 Timing of identification
FDP_DAU.2.1/Att	The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of *attestation data*[179] **by means of** FCS_COP.1/CDS-ECDSA[180] ***according to*** [35][181], *no further cryptographic authentication mechanisms*[182] **and keys holding the security attributes Key identity assigned to the TOE sample, and Key usage type "contentCommitment".**
FDP_DAU.2.2/Att	The TSF shall provide *external entities*[183] with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence.

Application note 23: The attestation data shall represent the TOE sample as a genuine sample of the certified product. The attestation data may include the identifier of the certified product, the serial number of the device or a group of product samples, the hash value of the TSF implementation and some TSF data as result of a self-test, or other data. It may be generated internally or may include internally generated and externally provided data. The assigned cryptographic mechanisms shall be appropriate for attestation meeting OSP.SecCryM, e. g. a digital signature, a group signature or a direct anonymous attestation mechanism as e.g. used for Trusted Platform Modules *[35]* or FIDO U2F Authenticators *[14]*.

[179] [assignment: *list of objects or information types*]
[180] [selection (by ST author): FCS_COP.1/CDS-RSA, FCS_COP.1/CDS-ECDSA, ECDAA]
[181] [selection (by ST author): [35, 14]]
[182] [assignment (by ST author): *other cryptographic authentication mechanisms*]
[183] [assignment: list of subjects]

Case	**Authentication of TOE and remote entity**	**Key agreement**	**Protection of communication data**	**Cryptographic operation**
1	FIA_API.1/PACE, FIA_UAU.5.1 (2)	FCS_CKM.1/PACE	modification	FCS_COP.1/TCM
2	FIA_API.1/PACE, FIA_UAU.5.1 (2)	FCS_CKM.1/PACE	modification disclosure	FCS_COP.1/TCM FCS_COP.1/TCE
3	FIA_API.1/CA, FIA_UAU.5.1 (4) or (5), and (6)	FCS_CKM.1/TCAP	modification disclosure	FCS_COP.1/TCM FCS_COP.1/TCE

Table 6.3: Operation in SFR for trusted channel (Table 4 in Base-PP [9])

FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 The TSF shall provide a communication channel between TSF and another trusted IT product that is ~~logically distinct from other communication channels~~ logically separated from other communication channels[184] and provides assured identification of its end points by authentication of the TOE and remote entity according to the case 1 and 2 in Table 6.3[185] and protection of the channel data from modification or disclosure *according to the case 1 and 2 in Table 6.3*[186] **as required by** cryptographic operation according to the case 1 and 2 in Table 6.3[187].

FTP_ITC.1.2 The TSF shall permit the remote trusted IT product[188] **determined according to FMT_MOF.1.1 clause (3)** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for *communication with entities defined according to FMT_MOF.1 clause (4)*[189].

***ST application note 12*: The selections/assignments in FTP_ITC.1 refer to Table 4 in the Base-PP [9] and to Table 6.3 in this ST document.**

[184] [selection (by ST author): logically separated from other communication channels, using physical separated ports]

[185] [selection (by ST author): Authentication of the TOE and remote entity according to the case in table 4]

[186] [assignment (by ST author): *according to the case in table 4*]

[187] [selection (by ST author): cryptographic operation according to the case in table 4]

[188] [selection: *the TSF, the remote trusted IT product*]

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

FCS_CKM.1/PACE Cryptographic key generation - Key agreement for trusted channel PACE

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/PACE The TSF shall generate cryptographic keys ***for MAC with for FCS_COP.1/TCM and if selected encryption keys for FCS_COP.1/TCE*** in accordance with a specified cryptographic key ~~generation~~ **agreement** algorithm *PACE with* Curve P-256[190] *and Generic Mapping in ICC role*[191] and specified cryptographic key sizes 128 bits, 256 bits[192] that meet the following: *ICAO Doc9303, Part 11, section 4.4 [16]*[193].

Application note 24: PACE is used to authenticate the TOE and the application component, or TOE and human user using a terminal. It establishes a trusted channel with MAC integrity protection and - if selected - also encryption.

ST application note 13**: The selections in FCS_CKM.1/PACE refer to Table 2 in the Base-PP [9] and to Table 6.1 in this ST document.**

FCS_CKM.1/TCAP Cryptographic key generation - Key agreement by Terminal and Chip authentication protocols

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/TCAP The TSF shall generate cryptographic keys ***for encryption according to FCS_COP.1/ TCE and MAC according to FCS_COP.1/TCM*** in accordance with a specified cryptographic key ~~generation~~ **agreement** algorithm *Terminal Authentication version 2 and Chip Authentication Version 2*[194] and specified cryptographic key sizes 128 bits, 256 bits[195] that meet the following: *BSI TR-03110 [4], section 3.3 and 3.4*[196].

ST application note 14**: The selections in FCS_CKM.1/TCAP refer to Table 2 in the Base-PP [9] and to Table 6.1 in this ST document.**

[190] [selection (by ST author): elliptic curves in table 2]
[191] [assignment: *cryptographic algorithm*]
[192] [selection (by ST author): 128 bits, 192 bits, 256 bits]
[193] [assignment: *list of standards*]
[194] [assignment: *cryptographic algorithm*]
[195] [selection (by ST author): 128 bits, 192 bits, 256 bits]
[196] [assignment: *list of standards*]

FCS_COP.1/TCE Cryptographic operation - Encryption for trusted channel

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/TCE The TSF shall perform *encryption and decryption*[197] in accordance with a specified cryptographic algorithm *AES in* CBC [27][198] *mode*[199] and specified cryptographic key sizes 128 bits, 256 bits[200] that meet the following: [23][201].

FCS_COP.1/TCM Cryptographic operation - MAC for trusted channel

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/TCM The TSF shall perform *MAC calculation and MAC verification*[202] in accordance with a specified cryptographic algorithm *AES* CMAC [28][203] and cryptographic key sizes 128 bits, 256 bits[204] that meet the following: [23][205].

[197] [assignment: *list of cryptographic operations*]
[198] [selection (by ST author): CBC [27], CCM [24], GCM [29]]
[199] [assignment: *cryptographic algorithm*]
[200] [selection (by ST author): 128 bits, 192 bits, 256 bits]
[201] [assignment: *list of standards*]
[202] [assignment: *list of cryptographic operations*]
[203] [selection (by ST author): CMAC [28], GMAC [29]]
[204] [selection (by ST author): 128 bits, 192 bits, 256 bits]
[205] [assignment: *list of standards*]

6.1.7 User identification and authentication

FIA_ATD.1 User attribute definition - Identity based authentication

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users:

(1) *Identity,*

(2) *Authentication reference data,*

(3) *Role.*

FMT_MTD.1/RAD Management of TSF data - Authentication reference data and Authentication Data Records

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/RAD The TSF shall restrict the ability to

(1) *create*[206] *the initial Authentication reference data of all authorized users*[207] *to* User Administrator[208]

(2) ***delete***[209] ***the Authentication reference data of an authorized user***[210] ***to*** User Administrator[211] ,

(3) ***modify***[212] ***the Authentication reference data***[213] ***to the corresponding authorized user***[214]***.***

(4) ***create***[215] ***the permanently stored session key of a trusted channel as Authentication reference data***[216] ***to*** User Administrator[217].

(5) ***define***[218] ***the time in range*** *[0; ∞]*[219] ***after which the user security attribute Role of the authentication data record is reset according to FMT_SAE.1***[220] ***to*** User Administrator[221],

(6) ***define***[222] ***the value*** Unidentified user[223] ***to which the security attribute Role of the authentication data record shall be reset according to FMT_SAE.1***[224] ***to*** User Administrator[225].

ST application note 15*: FMT_MTD.1/RAD (6) defines the role to which the role of an user or entity is reset once it expires. As the only selected option is *Unidentified

[206] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[207] [assignment: *list of TSF data*]
[208] [selection (by ST author): Administrator, User Administrator]
[209] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[210] [assignment: *list of TSF data*]
[211] [selection (by ST author): Administrator, User Administrator]
[212] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[213] [assignment: *list of TSF data*]
[214] [assignment: *the authorised identified roles*]
[215] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[216] [assignment: *list of TSF data*]
[217] [selection (by ST author): Administrator, User Administrator]
[218] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[219] [assignment (by ST author): *time frame*]
[220] [assignment: *list of TSF data*]
[221] [selection (by ST author): Administrator, User Administrator]
[222] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[223] [selection (by ST author): Unidentified user, Unauthenticated user]
[224] [assignment: *list of TSF data*]
[225] [selection (by ST author): Administrator, User Administrator]

user, the role is always reset to _Unidentified user_.

Application note 25: The Administrator is responsible for user management. The Administrator creates and revokes a user as a known authorized user of the TSF by creating resp. deleting authentication data records and additionally authentication reference data for the user identities in these records, as defined in clause (1). The Administrator may define additional authentication reference data as described in clause (3), i. e. the trusted channel combines initial authentication of communication endpoints (cf. FIA_UAU.5.1 clause (3) and (4)) with an agreement of session keys used for authentication of exchanged messages (cf. FIA_UAU.5.1 clause (5)). The session keys may be permanently stored for trusted communication with the known authorized entity. The user manages its own authentication reference data to prevent impersonation based of known authentication data (e.g. as addressed by FMT_MTD.3). The bullets (2) to (6) are refinements in order to avoid an iteration of component and therefore printed in bold.

FMT_MTD.3 Secure TSF data

Hierarchical to: No other components.

Dependencies: FMT_MTD.1 Management of TSF data

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for *passwords*[226] **by enforcing a change of initial passwords to a different operational password on the first successful authentication of the 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 User Administrator configurable positive integer within *[1; 10]*[227] unsuccessful authentication attempts occur related to *user authentication*[228].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met[229], the TSF shall *block the corresponding user authentication using a User Administrator configurable time span*[230].

[226] [assignment: *list of TSF data*]

[227] [selection (by ST author): [assignment: *positive integer number*], an ~~administrator~~ **[selection: Administrator, User Administrator]** configurable positive integer within [assignment: range of acceptable values]]

[228] [assignment (by ST author): *list of authentication events*]

[229] [selection (by ST author): met, surpassed]

[230] [assignment (by ST author): *list of actions*]

FIA_USB.1 User-subject binding

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition

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

(1) *Identity,*

(2) *Role*[231]*.*

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: *the initial role of the user is Unidentified user*[232].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

(1) *after successful identification of the user, the attribute Role of the subject shall be changed from Unidentified user to Unauthenticated user;*

(2) *after successful authentication of the user for a selected role, the attribute Role of the subject shall be changed from Unauthenticated User to that role;*

(3) *after successful re-authentication of the user for a selected role, the attribute Role of the subject shall be changed to that role*[233]*.*

FMT_SAE.1 Time-limited authorisation

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FPT_STM.1 Reliable time stamps

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for *a Role*[234] *to* User Administrator[235].

FMT_SAE.1.2 For each of these security attributes, the TSF shall be able to *reset the Role to the value assigned according to FMT_MTD.1/RAD, clause (6)*[236], after the expiration time for the indicated security attribute has passed.

Application note 26: The TSF shall implement means to handle an expiration time for the roles

[231] [assignment: *list of user security attributes*]
[232] [assignment: *rules for the initial association of attributes*]
[233] [assignment: *rules for the changing of attributes*]
[234] [assignment: *list of security attributes for which expiration is to be supported*]
[235] [selection (by ST author): Administrator, User Administrator]
[236] [assignment: *list of actions to be taken for each security attribute*]

within a session (i.e. between power-up and power-down of the TOE) which may not necessarily meet the requirements for a reliable time stamp as required by FPT_STM.1. If the security target requires FPT_STM.1 (e.g. if the PP-module "Time Stamp and Audit" claimed), this time stamp shall be used to meet FMT_SAE.1.

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.1,*

(2) *identification of the TOE to the user,*

(3) *no further actions*[237]

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

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of ~~that user~~ **the Unauthenticated 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.1,*

(2) *authentication of the TOE to the user after authentication of the user to the TOE,*

(3) *identification of the user to the TOE and selection of* a set of role[238] *for authentication* **while ensuring that the Auditor role is mutually exclusive with other roles**,

(4) *attestation according to FDP_DAU.2 and key agreement between master and slave node in cluster*[239]

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

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.

[237] [assignment (by ST author): *list of other TSF-mediated actions [assignment: list of TSF mediated actions]*]
[238] [selection (by ST author): a role, a set of role]
[239] [assignment (by ST author): *list of other TSF-mediated actions*]

Application note 27: Clause (2) and (3) in FIA_UAU.1.1 allows mutual identification for mutual authentication, e. g. by exchange of certificates.

FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1 The TSF shall provide

(1) *password authentication,*

(2) *PACE with Generic Mapping with the TOE in ICC and the user in PCD context with the establishment of trusted channel according to FTP_ITC.1,*

(3) *certificate based Terminal Authentication Version 2 according to section 3.3 in [4] with the TOE in ICC and the user in PCD context,*

(4) *Terminal Authentication Version 2 with the TOE in ICC context and user in PCD context modified by omitting the verification of the certificate chain (simplified TA2),*

(5) *Chip Authentication Version 2 with establishment of a trusted channel according to FTP_ITC.1,*

(6) *message authentication by MAC verification of received messages*[240]

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the **rules**

(1) *password authentication shall be used for authentication of human users if enabled according to FMT_MOF.1.1, clause (1),*

(2) *PACE shall be used for authentication of human users using terminals with the establishment of a trusted channel according to FTP_ITC.1,*

(3) *PACE may be used for authentication of IT entities with the establishment of a trusted channel according to FTP_ITC.1,*

(4) *certificate based Terminal Authentication Version 2 may be used for authentication of users whose certificate is imported as TSF data,*

(5) *the simplified version of Terminal Authentication Version 2 may be used for authentication of identified users associated with a known user's public key,*

(6) *message authentication by MAC verification of received messages shall be used after initial authentication of a remote entity according to clauses (2) or (3) for a trusted channel according to FTP_ITC.1,*

(7) *no further rules*[241].

[240] [assignment: *list of multiple authentication mechanisms*]

[241] [assignment (by ST author): *additional rules*]

FIA_UAU.6 Re-authenticating

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

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions

(1) *changing to a role not selected for the current valid authentication session,*

(2) *power on or reset,*

(3) *every message received from entities after establishing trusted channel according to FIA_UAU.5.1, clause (2), (3) or (6),*

(4) *no other conditions under which re-authentication is required*[242]

[242] [assignment (by ST author): *list of other conditions under which re-authentication is required*]

6.1.8 Access control

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

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1/UD The TSF shall enforce the Cryptographic Operation SFP[243] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/UD The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/UD The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

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

FDP_ITC.2.5/UD The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:

(1) *user data imported for encryption according to FCS_COP.1/ED shall be imported with the attribute Key identity of the key and the identification of the requested cryptographic operation,*

(2) *user data imported for encryption according to FCS_COP.1/HEM shall be imported with the attribute Key identity of the public key encryption key or key agreement method,*

(3) *user data imported for decryption according to FCS_COP.1/HDM shall be imported with the attribute Key identity of the asymmetric decryption key, encrypted seed and data integrity check sum,*

(4) *user data imported for digital signature creation shall be imported with the attribute Key identity of the private signature key,*

(5) *user data imported for digital signature verification shall be imported with digital signature and Key identity of the public signature key*[244].

Application note 28: Keys to be used for the cryptographic operation of the imported user data are identified by security attribute Key identity.

[243] [assignment: *access control SFP, information flow control SFP*]
[244] [assignment: *additional importation control rules*]

FDP_ITC.2/TS Import of user data with security attributes - User data for time stamping

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1/TS The TSF shall enforce the Cryptographic Operation SFP[245] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/TS The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/TS The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

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

FDP_ITC.2.5/TS The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:

(1) *user data imported for time stamp generation to FDP_DAU.2/TS shall be imported with security attributes Key identity of the signature key and Key usage type TimeStamp, and the identification of the requested cryptographic operation*[246].

Application note 2 of [8]: Keys to be used for the cryptographic operation of the imported user data are identified by security attribute Key identity.

[245] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[246] [assignment: *additional importation control rules*]

FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1 The TSF shall enforce the *Cryptographic Operation SFP*[247] when exporting user data, controlled under the SFP(s), outside of the TOE.

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

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

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE:

(1) *user data exported as ciphertext according to FCS_COP.1/HEM shall be exported with reference to the key decryption key, encrypted data encryption key and data integrity check sum,*

(2) *user data exported as plaintext according to FCS_COP.1/HDM shall be exported only if the MAC verification confirmed the integrity of the ciphertext,*

(3) *user data exported as signed data according to FCS_COP.1/CDS-ECDSA or FCS_COP.1/CDS-RSA shall be exported with a digital signature and Key identity of the used signature-creation key*[248].

Application note 29: In case of internally generated data exported as signed data, the Key identity of the used key should be exported as well in order to identify the corresponding signature-verification key. Note that the TOE may implement more than one signature-creation key for signing internally generated data.

[247] [assignment: *access control SFP, information flow control SFP*]

[248] [assignment: *additional exportation control rules*]

FDP_ETC.2/TS Export of user data with security attributes - User data with time stamp

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1/TS The TSF shall enforce the Cryptographic Operation SFP[249] when exporting user data, controlled under the SFP(s), outside of the TOE.

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

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

FDP_ETC.2.4/TS The TSF shall enforce the following rules when user data is exported from the TOE:

(1) *user data exported as time stamped data according to FDP_DAU.2/TS shall be exported with digital signature and Key identity of the used signature-creation key*[250]

Application note 3 of [8]: In case of internally generated data (e.g. audit records) the exported signed data shall be attributed with the *Key identity* of the used signature-creation key. Note that the TOE may implement more than one signature-creation key for signing internally generated data.

FDP_ETC.1 Export of user data without security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.1.1 The TSF shall enforce the *Cryptographic Operation SFP*[251] when exporting user data **as plaintext according to FCS_COP.1/HDM,** controlled under the SFP(s), outside of the TOE.

FDP_ETC.1.2 The TSF shall export the ~~user data~~ **successfully MAC verified and decrypted ciphertext as plaintext according to FCS_COP.1/HDM** without the user data's associated security attributes.

[249] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[250] [assignment: *additional exportation control rules*]
[251] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

FDP_ACC.1/Oper Subset access control - Cryptographic operation

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/Oper	The TSF shall enforce the Cryptographic Operation SFP[252] on

(1) *subjects:* Crypto-Officer[253], *Key Owner, no other roles*[254]*;*

(2) *objects: operational cryptographic keys, user data;*

(3) *operations: cryptographic operation*[255]

[252] [assignment: *access control SFP*]

[253] [selection (by ST author): Administrator, Crypto-Officer]

[254] [assignment (by ST author): *other roles*]

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

FDP_ACF.1/Oper Security attribute based access control - Cryptographic operations

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

FDP_ACF.1.1/Oper The TSF shall enforce the Cryptographic Operation SFP[256] to objects based on the following:

(1) *subjects: subjects with security attribute Role* Crypto-Officer[257], *Key Owner, no other roles*[258]*;*

(2) *objects:*

(a) *cryptographic keys with security attributes: Identity of the key, Key owner, Key type, Key usage type, Key access control attributes, Key validity time period;*

(b) *user data*[259]

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

(1) *A Subject in* Crypto-Officer[260] *role is allowed to perform cryptographic operations on cryptographic keys in accordance with their security attributes.*

(2) *The Subject Key Owner is allowed to perform cryptographic operations on user data with cryptographic keys in accordance with the security attribute Key owner, Key type, Key usage type, Key access control attributes and Key validity time period;*

(3) *no other rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*[261]

[256] [assignment: *access control SFP*]

[257] [selection (by ST author): Administrator, Crypto-Officer]

[258] [assignment (by ST author): *other roles*]

[259] [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*]

[260] [selection (by ST author): Administrator, Crypto-Officer]

[261] [assignment (by ST author): *other rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

FDP_ACF.1.3/Oper The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

(1) *subjects with the security attribute Role are allowed to perform cryptographic operations on user data and cryptographic keys with security attributes as shown in the rows of table 5.*

(2) *no additional rules, based on security attributes, that explicitly authorise access of subjects to objects*[262]

FDP_ACF.1.4/Oper The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

(1) *No subject is allowed to use cryptographic keys by cryptographic operation other than those identified in the security attributes Key usage type and the Key access control attributes;*

(2) *No subject is allowed to decrypt ciphertext according to FCS_COP.1/HDM if MAC verification fails.*

(3) *no additional rules, based on security attributes, that explicitly deny access of subjects to objects*[263]

***ST application note 16*: FDP_ACF.1.3/Oper (1) refers to Table 5 in the Base-PP [9] and to Table 6.4 in this ST document.**

Access control rules for cryptographic operation:

[262] [assignment (by ST author): *additional rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[263] [assignment (by ST author): *additional rules, based on security attributes, that explicitly deny access of subjects to objects*]

Security attribute Role of the subject	**Security attribute of the cryptographic key**	**Cryptographic operation referenced by SFR allowed for the subject on user data with the cryptographic key**
Crypto-Officer selection (by ST author): [Administrator, Crypto-Officer, Key Owner]	*Key type: symmetric* *Key usage type: Key wrap* *Key validity time period:*	*FCS_COP.1/KW*
Crypto-Officer selection (by ST author): [Administrator, Crypto-Officer, Key Owner]	*Key type: symmetric* *Key usage type: Key unwrap* *Key validity time period:*	*FCS_COP.1/KU*
(any authenticated user)	*Key type: public* *Key usage type: ECKA-EG* *Key validity time period: as in certificate*	*FCS_COP.1/HEM* *FCS_CKM.1/ECKA-EG*
Key owner	*Key type: private* *Key usage type: ECKA-EG* *Key validity time period:*	*FCS_COP.1/HDM* *FCS_CKM.5/ECKA-EG*
(any authenticated user)	*Key type: public* *Key usage type: RSA_ENC* *Key validity time period: as in certificate*	*FCS_COP.1/HEM* *FCS_CKM.1/AES_RSA*
Key owner	*Key type: private* *Key usage type: RSA_ENC* *Key validity time period: as in certificate*	*FCS_COP.1/HDM* *FCS_CKM.5/AES_RSA*
Key owner	*Key type: private* *Key usage type: DS-ECDSA* *Key validity time period:*	*FCS_COP.1/CDS-ECDSA*
(any authenticated user)	*Key type: public* *Key usage type: DS-ECDSA* *Key validity time period:*	*FCS_COP.1/VDS-ECDSA*
Key owner	*Key type: private* *Key usage type: DS-RSA* *Key validity time period:*	*FCS_COP.1/CDS-RSA*
(any authenticated user)	*Key type: public* *Key usage type: DS-RSA* *Key validity time period:*	*FCS_COP.1/VDS-RSA*

Table 6.4: Security attributes and access control (Table 5 in Base-PP [9])

FDP_ACF.1/TS Security attribute based access control - Cryptographic operations

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/TS The TSF shall enforce the *Cryptographic Operation SFP*[264] to objects based on the following:

(1) *subjects: subjects with security attribute Role Application Component, no other role*[265]*;*

(2) *objects: user data*[266]

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

(1) *Application Component, Auditor*[267] *is allowed to perform cryptographic operation according to FDP_DAU.2/TS on user data with cryptographic keys with Key usage type TimeStamp.*

(2) *no further rules*[268].

FDP_ACF.1.3/TS The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *no additional rules, based on security attributes, that explicitly authorise access of subjects to objects*[269].

FDP_ACF.1.4/TS The TSF shall explicitly deny access of subjects to objects based on the

(1) *No subject is allowed to use cryptographic keys by cryptographic operation other than those identified in the security attributes Key usage type and the Key access control attributes;*

(2) *no further rules*[270].

[264] [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*]

[265] [assignment (by ST author): *other roles*]

[266] [assignment: *access control SFP*]

[267] [assignment (by ST author): *other roles*]

[268] [assignment (by ST author): *other rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[269] [assignment (by ST author): *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[270] [assignment (by ST author): *additional rules, based on security attributes, that explicitly deny access of subjects to objects*]

6.1.9 Security Management

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) *management of security functions behaviour (FMT_MOF.1),*

(2) *management of Authentication reference data (FMT_MTD.1/RAD),*

(3) *management of security attributes of cryptographic keys (FMT_MSA.1/KM, FMT_MSA.2, FMT_MSA.3/KM,*

(4) *no further functions*[271].

FMT_SMF.1/TSA Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

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

(1) *management of security functions behaviour FMT_MOF.1/TSA*[272]

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: *Unidentified User, Unauthenticated User, Key Owner, Application component,* Crypto-Officer, User Administrator, Update Agent[273] no other roles[274].

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

[271] [assignment (by ST author): *additional list of security management functions to be provided by the TSF*]
[272] [assignment: *list of management functions to be provided by the TSF*]
[273] [selection (by ST author): Administrator, Crypto-Officer, User Administrator, Update Agent]
[274] [selection (by ST author): [assignment: other roles], no other roles]]

Application note 30: The ST may select the general role Administrator or more detailed administrator roles as supported by the TOE.

FMT_SMR.1/TSA Security roles

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.1.1/TSA	The TSF shall maintain the roles **additional to those required by FMT_SMR.1 in the Base-PP**: Auditor, Timekeeper[275]
FMT_SMR.1.2/TSA	The TSF shall be able to associate users with roles.

Application note 4 of [8]: The ST may select the general role *Administrator* or more detailed Administrator roles as supported by the TOE. The ST may select

- *Auditor* role in FMT_SMR.1/TSA separated from Administrator roles selected in the SFR FMT_SMR.1 according to the Base-PP and, or
- *Timekeeper* role in FMT_SMR.1/TSA separated from Administrator roles selected in the SFR FMT_SMR.1 according to the Base-PP and, or
- *no other roles* in FMT_SMR.1/TSA and assign the management of audit TSF in FMT_MTD.1/Audit to a selected Administrator role in the SFR FMT_SMR.1 according to the Base-PP.

The assignment of security management of audit and other functions must not result in a conflict of duties

[275] [selection (by ST author): Auditor, Timekeeper, no other roles]

FMT_MSA.2 Secure security attributes

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

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes:

(1) *Key identity,*

(2) *Key type,*

(3) *Key usage type,*

(4) *no additional security attributes*[276].

The cryptographic keys shall have

(1) **a Key identity uniquely identifying the key among all keys implemented in the TOE,**

(2) **the Key type defined as exactly one of secret key, private key, or public key,**

(3) **a Key usage type identifying at least one cryptographic mechanism the key can be used for.**

[276] [assignment (by ST author): *additional security attributes*]

FMT_MOF.1 Management of security functions behaviour

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1 The TSF shall restrict the ability to

(1) *enable*[277] *the functions password authentication according to FIA_UAU.5.1, clause (1)*[278] *to* User Administrator[279].

(2) ***disable***[280] ***the functions password authentication according to FIA_UAU.5.1, clause (1)***[281] ***to*** User Administrator[282]*,*

(3) ***determine the behavior of***[283] **the functions *trusted channel according to FDP_ITC.1.2***[284] **by defining the remote trusted IT products permitted to initiate communication via the trusted channel to** User Administrator[285],

(4) ***determine the behavior of***[286] **the functions *trusted channel according to FDP_ITC.1.3***[287] **by defining the entities for which the TSF shall enforce communication via the trusted channel to** User Administrator[288],

(5) **The TSF shall restrict the ability to determine the behavior of[289] the functions FIA_AFL.1[290] to User Administrator[291].**

Application note 31: The refinements of FMT_MOF.1.1 in bullets (2) to (4) are made in order to avoid iteration of the component. In case of the client-server architecture, the applications using the TOE and supporting the cryptographically protected trusted channel belong to the entities for which the TSF shall enforce a trusted channel according to FDP_ITC.1, cf. FMT_MOF.1.1 in bullet (4).

***ST application note 17*: FDP_ITC should be FTP_ITC in FMT_MOF.1 and Application note 31. The ST author decided to mention this typo in the PP by an ST application note instead of making refinements to the PP.**

[277] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[278] [assignment: *list of functions*]
[279] [selection (by ST author): Administrator, User Administrator]
[280] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[281] [assignment: *list of functions*]
[282] [selection (by ST author): Administrator, User Administrator]
[283] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[284] [assignment: *list of functions*]
[285] [selection (by ST author): Administrator, User Administrator]
[286] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[287] [assignment: *list of functions*]
[288] [selection (by ST author): Administrator, User Administrator]
[289] [selection: determine the behaviour of, disable, enable, modify the behaviour of]
[290] [assignment (by ST author): list of functions]
[291] [assignment (by ST author): the authorised identified roles]

FMT_MOF.1/TSA Management of security functions behaviour

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MOF.1.1/TSA	The TSF shall restrict the ability to

(1) *modify the behaviour of*[292] *the functions adjustment of the internal clock according to FPT_STM.1 clause (1)*[293] *to* Timekeeper[294],

(2) ***modify the behaviour of***[295] ***the functions adjustment of the internal clock according to FPT_STM.1 clause (2)***[296] ***to*** Timekeeper[297],

(3) ***determine the behaviour of and modify the behaviour of***[298] ***the functions select the auditable events according to FAU_GEN.1***[299] ***to*** Auditor[300]

(4) ***determine the behaviour of and modify the behaviour of***[301] ***the functions automatic export of audit trails according to FAU_STG.3.1 clause (1)***[302] ***to*** Auditor[303]

(5) ***determine the behaviour of and modify the behaviour of***[304] ***the functions FDP_DAU.2/TS by selection of signature key used to sign exported audit trails to*** Auditor[305]***.***

Application note 5 of [8]: The SFR defines additional management of security functions behaviour for new SFR with respect to the Base-PP. The refinements of FMT_MOF.1.1/TSA in bullets (2) to (5) are made in order to avoid further iterations of the component.

[292] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[293] [assignment: *list of functions*]
[294] [selection (by ST author): Administrator, Timekeeper]
[295] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[296] [assignment: *list of functions*]
[297] [selection (by ST author): Administrator, Timekeeper]
[298] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[299] [assignment: *list of functions*]
[300] [selection (by ST author): Administrator, Auditor]
[301] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[302] [assignment: *list of functions*]
[303] [selection (by ST author): Administrator, Auditor]
[304] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[305] [selection (by ST author): Administrator, Auditor]

6.1.10 Security Audit

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[306] level of audit; and

(c) *Discrete adjustment of the real time clock*

(1) *by automatic adjustment of the clock according to FPT_STM.1.1 clause (2) if selected as auditable event,*

(2) *by* **Timekeeper as a refinement of** *Administrator according to FPT_STM.1.1 clause (1) or(2),*

(3) *failure of adjustment according to FPT_STM.1.1*

(d) other auditable events

(1) *Start-up after power-up*

(2) *Import of UCP (FDP_ITC.2/UCP),*

(3) *Authentication failure handling (FIA_AFL.1): the reaching of the threshold for the unsuccessful authentication attempts with claimed Identity of the user,*

(4) Generation of (selected types of) signature key pairs (all FCS_CKM.1 instantiations for generation of permanent stored keys), (6) Cryptographic key destruction (FCS_CKM.4) of permanent stored keys [307]

(11) *custom audit events created by entities in administrative roles, (un)blocking of users by the User Administrator if configured via FMT_MTD.1/Audit*[308].

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, *further information regarding the event if applicable*[309].

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

[307] [selection (by ST author): (4) *Generation of (selected types of) signature key pairs (all FCS_CKM.1*

Application note 6 of [8]: The SFR FDP_ITC.2/UCP, FIA_AFL.1, FCS_CKM.1, FCS_COP.1, FCS_CKM.4, FPT_FLS.1 and FMT_MOF.1 are defined in the Base-PP. The SFR FPT_STM.1, FMT_MOF.1/TSA and FMT_MTD.1/Audit are defined in this PP-Module.

***ST application note 18*: "selected types of" in FAU_GEN.1.1(d)(4) refers to permanently stored keys.**

FAU_GEN.1/CL Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1/CL 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[310] level of audit; and

(c) *other auditable events*

(1) *Generation of cluster keys for the secure channel according to FMT_MTD.1/CL and FCS_CKM.5/CLDH,*

(2) *Export of Authentication Data Records and cryptographic keys from the MasterCSPLight according to FPT_ESA.1.3/CL,Management of Authentication Data Records (FMT_MTD.1/RAD): creation and deletion of Authentication Data Record*

(3) *Import according to FPT_ISA.1/CL of Authentication Data Records and cryptographic keys into Slave-CSPLights.*[311]

FAU_GEN.1.2/CL 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, *further information regarding the event if applicable*[312].

instantiations for generation of permanent stored keys) (5) *Execution of (selected types of) cryptographic operation (all FCS_COP.1 instantiations),* (6) *Cryptographic key destruction (FCS_CKM.4) of permanent stored keys* (7) *Failure with preservation of secure state (FPT_FLS.1): entering and exiting secure state* (8) *Management of security functions (FMT_MOF.1, FMT_MOF.1/TSA),* (9) *Management of TSF data (FMT_MTD.1/AUDIT): Export, clear and selection of events causing audit data* (10) *No other event,*]

[308] [assignment (by ST author): *additional specifically defined auditable events*]

[309] [assignment (by ST author): *other audit relevant information*]

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

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

[312] [assignment (by ST author): *other audit relevant information*]

Application note 5 of [10]: The SFR FAU_GEN.1/CL adds auditable events to FAU_GEN.1 required by PPM-TS-Au. The SFR FPT_STM.1 is required by PPM-TS-Au.

Application note 6 of [10]: FMT_MTD.1/RAD is defined in the Base-PP.

FMT_MTD.1/Audit 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/Audit The TSF shall restrict the ability to

(1) *manual export*

(2) *clear after manual export,*

(3) *select audited events in FAU_GEN.1* **and FAU_GEN.1/CL***,*

(4) *define the number of audit records causing automatic export and clearing of exported audit records according to FAU_STG.3.1 clause (1),*

(5) *define the percentage of storage capacity of audit records if actions are assigned in FAU_STG.3.1 clause (2)*[313]

the audit records[314] to Auditor[315].

Application note 7 of [8]: The selection of auditable events according to FMT_MTD.1.1/Audit, clause (3) enables or disables or specifies the generation of audit records as defined in FAU_GEN.1. The role Administrator may be selected only if it is selected in FMT_SMR.1 in the Base-PP and any conflict of duties is prevented (cf. application note to FMT_SMR.1/TSA).

***ST application note 19*: It should be noted that the TOE enforces the limit that has been set in FAU_STG.3.1 with an accuracy of +/- 1 MB. Due to the sheer size of the available disk space (of at least 500 GB, this falls into the area of a rounding error.)**

[313] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]
[314] [assignment: *list of TSF data*]
[315] [selection (by ST author): Administrator, Auditor]

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to *prevent*[316] unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.3 Action in Case of Possible Audit Data Loss

Hierarchical to: No other components.

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.3.1 The TSF shall

(1) *automatically export audit trails and clear automatically exported audit records*[317] *if the audit trail exceeds an* Auditor[318] *defined number of audit records within* *[*10^{100} – 10^{100}*]*[319]

(2) *enter an error state that will not allow operations causing audit events except those that are required to export the audit log or adjust the setting for the threshold*[320] **if the audit trail exceeds an** Auditor[321] ***settable percentage of storage capacity***[322]**.**

Application note 8 of [8]: The ST writer shall perform the open operations in FAU_STG.3.1 element. If the number of audit records in clause (1) is set to 1 then the TSF export each audit record automatically. If the number of number of audit records in clause (1) is set higher than maximum number of audit records in the audit trail then the TSF does not export audit records automatically. The assignment of clause (2) may be “no actions” if an appropriate number of audit records is assigned in clause (1).

ST application note 20**: The first assignment in FAU_STG.3 has been set to a googol to make clear that the TOE does not support automatic exports. It should be noted that the log messages that are provided to a SMAERS component as a result of a call of the API of the TOE are neither considered an automatic nor a manual export.**

[316] [selection: *choose one of: prevent, detect*]
[317] [assignment: *actions to be taken in case of possible audit storage failure*]
[318] [selection (by ST author): Administrator, Auditor]
[319] [assignment (by ST author): *pre-defined range*]
[320] [assignment (by ST author): *actions to be taken in case of possible audit storage failure*]
[321] [selection (by ST author): Administrator, Auditor]
[322] [assignment: *pre-defined limit*]

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 **by means of** internal clock with accuracy *5 seconds per day* with automatic adjustment of the clock by an externally trustable source in a cryptographically verifiable manner (e.g. by signed Network Time Protocol) and the ability of adjustment of the clock by the Timekeeper.[323]

Application note 9 of [8]: The external trustable source (e.g. signed Network Time Protocol) provides a reliable time source for adjustment of the internal clock. The time intervals of adjustments in clause (2) may be configured by the Administrator. Any adjustment or failure of adjustment of the internal clock is an auditable event according to FAU_GEN.1.1.The refinement with selection defines different cases for internal clocks and are therefore printed in bold.

Note that it is not expected that the internal clock continues to operate when the TOE is switched off. An implementation that e.g. counts CPU ticks with sufficient accuracy while switched on would suffice to fulfil the requirements, provided that all auditable events are logged properly.

FPT_TIT.1/Audit TSF data integrity transfer protection - Audit functionality

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]
FPT_TIT.1.1/Audit	The TSF shall enforce the *Update SFP*, Key Management SFP[324] to *transmit*[325] TSF data **audit records** in a manner protected from *modification, deletion, insertion and replay*[326] errors.
FPT_TIT.1.2/Audit	The TSF shall be able to determine on receipt of TSF data **time**, whether *modification*[327] has occurred.

Application note 10 of [8]: The Update SFP is enforced by the export of audit records about

[323] [selection (by ST author):

(1) ***internal clock with accuracy [assignment: approximate deviation] with the ability of adjustment of the clock by the [selection: Administrator, Timekeeper],***

(2) ***internal clock with accuracy [assignment: approximate deviation] with automatic adjustment of the clock by an externally trustable source in a cryptographically verifiable manner (e.g. by signed Network Time Protocol) and the ability of adjustment of the clock by the [selection: Administrator, Timekeeper].***

]

[324] [selection (by ST author): Key Management SFP, Cryptographic Operation SFP]

[325] [selection: *transmit, receive, transmit and receive*]

[326] [selection: *modification, deletion, insertion, replay*]

[327] [selection: *modification, deletion, insertion, replay*]

import of UCP, cf. FAU_GEN.1.1 clause d) (2). The selection of the Key Management SFP or Cryptographic Operation SFP depends on the selection of auditable events of key management, cryptographic operations and adjustment of the internal clock (e. g. used for verification of validity time period) in FAU_GEN.1.1 clause c). The TSF transmits audit records and receives time as TSF data for security audit. The TSF protects the audit records by means of digital signature against modification and by means of time stamps and key usage counter of the signature key as part of the signature against deletion, insertion and replay as required in FPT_TIT.1.1.

6.1.11 Protection of the TSF

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 fails*

(2) *none*[328].

Refinement: When the TOE is in a secure error mode the TSF shall not perform any cryptographic operations and all data output interfaces shall be inhibited by the TSF.

FPT_TST.1 TSF testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self tests *during initial start-up and after power-on*[329] to demonstrate the correct operation of *the random number generator, the cryptographic functionality and the access control functionality*[330].

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of *TSF data*[331].

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of *TSF* ***implementation***[332].

[328] [assignment (by ST author): *list of types of additional failures*]

[329] [selection: *during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions[assignment: conditions under which self test should occur]*]

[330] [assignment (by ST author): *parts of TSF*]

[331] [selection: *[assignment: parts of TSF data], the TSF data*]

[332] [selection: *[assignment: parts of TSF], TSF*]

6.1.12 Import and verification of Update Code Package

The TOE imports Update Code Package as user data objects with security attributes according to FDP_ITC.2/ UCP, verifies the authenticity of the received Update Code Package according to FCS_COP.1/VDSUCP, and decrypts authentic Update Code Package according to FCS_COP.1/DecUCP.

FDP_ITC.2/UCP Import of user data with security attributes - Update Code Package

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency
FDP_ITC.2.1/UCP	The TSF shall enforce the *Update SFP*[333] when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2/UCP	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3/UCP	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.
FDP_ITC.2.4/UCP	The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.
FDP_ITC.2.5/UCP	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: (1) *encrypted Update Code Package are stored only after successful verification of authenticity according to FCS_COP.1/VDSUCP,* (2) *authentic Update Code Package are decrypted according to FCS_COP.1/DecUCP*[334].

[333] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[334] [assignment: *additional importation control rules*]

FPT_TDC.1/UCP Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1/UCP The TSF shall provide the capability to consistently interpret *security attributes Issuer and Version Number*[335] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2/UCP The TSF shall use **the following rules:**

(1) *the Issuer must be identified and known,*

(2) *the Version Number must be identified*

when interpreting the TSF data from another trusted IT product.

FCS_COP.1/VDSUCP Cryptographic operation - Verification of digital signature of the Issuer

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/VDSUCP The TSF shall perform *verification of the digital signature of the authorized Issuer*[336] in accordance with a specified cryptographic algorithm *Curve P-256*[337] and cryptographic key sizes *256 bits*[338] that meet the following: *[3]* [339].

Application note 32: The authorized *Issuer* is identified in the security attribute of the received Update Code Package and the public key of the authorized *Issuer* shall be known as TSF data before receiving the Update Code Package. Only the public key of the authorized *Issuer* shall be used for the verification of the digital signature of the Update Code Package.

[335] [assignment: *list of TSF data types*]
[336] [assignment: *list of cryptographic operations*]
[337] [assignment (by ST author): *cryptographic algorithm*]
[338] [assignment (by ST author): *cryptographic key sizes*]
[339] [assignment (by ST author): *list of standards*]

FCS_COP.1/DecUCP Cryptographic operation - Decryption of authentic Update Code Package

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/DecUCP	The TSF shall perform *decryption of authentic encrypted Update Code Package*[340] in accordance with a specified cryptographic algorithm *AES-CBC*[341] and cryptographic key sizes *256 bits*[342] that meet the following: *[27]* [343].

FDP_ACC.1/UCP Subset access control - Update code Package

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/UCP	The TSF shall enforce the Update SFP[344] on

(1) *subjects:* Update Agent[345]*;*

(2) *objects: Update Code Package;*

(3) *operations: import, store*[346]

[340] [assignment: *list of cryptographic operations*]
[341] [assignment (by ST author): *cryptographic algorithm*]
[342] [assignment (by ST author): *cryptographic key sizes*]
[343] [assignment (by ST author): *list of standards*]
[344] [assignment: *access control SFP*]
[345] [selection (by ST author): Administrator, Update Agent]
[346] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

FDP_ACF.1/UCP Security attribute based access control - Import Update Code Package

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/UCP The TSF shall enforce the Update SFP[347] to objects based on the following:

(1) *subjects:* Update Agent[348]*;*

(2) *objects: Update Code Package with security attributes Issuer and Version Number*[349]*.*

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

(1) Update Agent[350] *is allowed to import Update Code Package according to FDP_ITC.2/UCP.*

(2) Update Agent[351] *is allowed to store a Update Code Package if*

(a) *authenticity is successfully verified according to FCS_COP.1/VDSUCP and the Update Code Package is decrypted according to FCS_COP.1/DecUCP*

(b) *the Version Number of the Update Code Package is equal or higher than the Version Number of the TSF.*[352]

FDP_ACF.1.3/UCP The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *no further rules*[353].

FDP_ACF.1.4/UCP The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *no further rules*[354].

[347] [assignment: *access control SFP*]
[348] [selection (by ST author): Administrator, Update Agent]
[349] [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*]
[350] [selection (by ST author): Administrator, Update Agent]
[351] [selection (by ST author): Administrator, Update Agent]
[352] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[353] [assignment (by ST author): *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[354] [assignment (by ST author): *rules, based on security attributes, that explicitly deny access of subjects to objects*]

FDP_RIP.1/UCP Subset residual information protection

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_RIP.1.1/UCP	The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource* ***after unsuccessful verification of the digital signature of the Issuer according to FCS_COP.1/VDSUCP***[355] the following objects: *received Update Code Package*[356].

6.1.13 Clustering

The cluster of TOE samples is set up by the Administrator as Cluster-CSPLight(s) by

- selecting one TOE sample of the cluster as Master-CSPLight, all other TOE samples of the cluster are Slave-CSPLight(s),
- initialization of secure channels between the Master-CSPLight and the Slave-CSPLight(s),
- transfer of TSF data as security attributes of known users and cryptographic keys with security attributes between Master-CSPLight and Slave-CSPLight(s) using the application.

FDP_ACC.1/CL Subset access control - Clustering

Hierarchical to:	No other components.
Dependencies	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/CL	The TSF shall enforce the Clustering SFP[357] on (1) *subjects:* **Crypto-Officer** (2) *objects: cluster keys, Authentication Data Records, cryptographic keys;* (3) *operations: generation, export, import*[358]

[355] [selection: *allocation of the resource to, deallocation of the resource from*]
[356] [assignment: *list of objects*]
[357] [assignment: *access control SFP*]
[358] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

FMT_MTD.1/CL Management of TSF data - Authentication Data Records and cryptographic keys

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management

FMT_MTD.1.1/CL The TSF shall restrict the ability to

(1) *generate according to FCS_CKM.5/CLDH*[359] *the cluster keys*[360] *to* **Crypto Officer** ~~*Administrator*~~[361]

(2) ***export from the Master-CSPLight according to FPT_ESA.1/CL, FPT_TCT.1/CL and FPT_TIT.1/CL***[362] ***the Authentication Data Records***[363] ***to* Crypto Officer*,***

(3) ***import into Slave-CSPLights according to FPT_ISA.1/CL, FPT_TCT.1/CL and FPT_TIT.1/CL***[364] ***the Authentication Data Records***[365] ***to* Crypto Officer**

(4) ***export from the Master-CSPLight according to FPT_ESA.1/CL, FPT_TCT.1/CL and FPT_TIT.1/CL***[366] ***the cryptographic keys***[367] ***to*** Crypto-Officer[368],

(5) ***import into Slave-CSPLights according to FPT_ISA.1/CL, FPT_TCT.1/CL and FPT_TIT.1/CL***[369] ***the cryptographic keys***[370] ***to*** Crypto-Officer[371].

Application note 1 of [10]: Authentication Data Records and cryptographic keys are TSF data. The selection in FMT_MTD.1/CL allows for a more detailed separation of duties between the roles if supported by the TOE. The bullets (2) to (5) are refinements to avoid further iterations of the component FMT_MTD.1.1/CL and therefore printed in bold.

[359] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[360] [assignment: *list of TSF data*]
[361] [assignment: *the authorised identified roles*]
[362] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[363] [assignment: *list of TSF data*]
[364] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[365] [assignment: *list of TSF data*]
[366] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[367] [assignment: *list of TSF data*]
[368] [selection (by ST author): Application Component, Administrator, Crypto-Officer]
[369] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[370] [assignment: *list of TSF data*]
[371] [selection (by ST author): Application Component, Administrator, Crypto-Officer]

FCS_CKM.5/CLDH Cryptographic key derivation - Cluster keys

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.5.1/CLDH	The TSF shall derive cryptographic *cluster keys*[372] from *an agreed shared secret*[373] in accordance with a specified cryptographic key derivation algorithm *anonymous Diffie-Hellman Key Agreement for ECC key pair generation with* Curve P-256[374] and specified cryptographic key sizes 256 bits[375] that meet the following: FIPS PUB 186-4 B.4 and D.1.2.3[376]

Application note 2 of [10]: The cryptographic cluster keys shall be used for encryption according to FCS_COP.1/ED (cf. Base-PP) and FPT_TCT.1/CL and MAC protection according to FCS_COP.1/MAC (cf. Base-PP) and FPT_TIT.1/CL during transfer of Authentication Data Records and the cryptographic keys between MasterCSPLight and Slave-CSPLight. The tables 2 and 3 are defined in the Base-PP [9].

***ST application note 21*: The selections in FCS_CKM.5/CLDH refer to Table 2 / Table 3 in the Base-PP [9] and to Table 6.1 / Table 6.2 in this ST document.**

FPT_TCT.1/CL TSF data confidentiality transfer protection - Cluster

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset informationflow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]
FPT_TCT.1.1/CL	The TSF shall enforce the *Clustering SFP*[377] by providing the ability to *transmit and receive*[378] **Authentication Data Records and cryptographic keys** ~~TSF data~~ in a manner protected from unauthorised disclosure **according to FCS_COP.1/ED.**

Application note 3 of [10]: FCS_COP.1/ED is defined in the Base-PP.

[372] [assignment: *key type*]
[373] [assignment: *input parameters*]
[374] [selection (by ST author): elliptic curves in the table 2 [9]]
[375] [selection (by ST author): key size in the table 2 [9]]
[376] [selection (by ST author): standards in the tables 2 and 3 [[9], [3]]]
[377] [assignment: *access control SFP, information flow control SFP*]
[378] [selection: *transmit, receive, transmit and receive*]

FPT_TIT.1/CL TSF data integrity transfer protection - Cluster

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset informationflow control]
[FMT_MTD.1 Management of TSF data or
FMT_MTD.3 Secure TSF data]

FPT_TIT.1.1/CL The TSF shall enforce the *Clustering SFP*[379] to *transmit and receive*[380] **Authentication Data Records and cryptographic keys** ~~TSF data~~ in a manner protected from *modification*[381] errors **according to FCS_COP.1/MAC.**

FPT_TIT.1.2/CL The TSF **in role Slave-CSPLight** shall be able to determine on receipt of **Authentication Data Records and cryptographic keys** ~~TSF data~~, whether *modification*[382] has occurred **according to FCS_COP.1/MAC.**

Application note 4 of [10]: FCS_COP.1/MAC is defined in the Base-PP.

[379] [assignment: *access control SFP, information flow control SFP*]
[380] [selection: *transmit, receive, transmit and receive*]
[381] [selection: *modification, deletion, insertion, replay*]
[382] [selection: *modification, deletion, insertion, replay*]

FPT_ISA.1/CL Import of TSF data with security attributes - Cluster

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset informationflow control]
[FMT_MTD.1 Management of TSF data, or
FMT_MTD.3 Secure TSF data]
[FMT_MSA.1 Management of security attributes, or
FMT_MSA.4 Security attribute value inheritance]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_ISA.1.1/CL The TSF **in role Slave-CSPLight** shall enforce the *Clustering SFP*[383] when importing **Authentication Data Records and cryptographic keys** ~~TSF data~~, controlled under the SFP, from ~~outside of the TOE~~ **Master-CSPLight**.

FPT_ISA.1.2/CL The TSF **in role Slave-CSPLight** shall use the security attributes associated with the imported **Authentication Data Records and cryptographic keys** ~~TSF data~~.

FPT_ISA.1.3/CL The TSF **in role Slave-CSPLight** shall ensure that the protocol used provides for the unambiguous association between the security attributes and the **Authentication Data Records and cryptographic keys** ~~TSF data~~ received.

FPT_ISA.1.4/CL The TSF **in role Slave-CSPLight** shall ensure that interpretation of the security attributes of the imported **Authentication Data Records and cryptographic keys** ~~TSF data~~ is as intended by the source of the **Authentication Data Records and cryptographic keys** ~~TSF data~~.

FPT_ISA.1.5/CL The TSF **in role Slave-CSPLight** shall enforce the following rules when importing **Authentication Data Records and cryptographic keys** ~~TSF data~~ controlled under the SFP from ~~outside of the TOE~~ **Master-CSPLight:**

(1) *TSF in role Slave-CSPLight always imports Authentication Data Records with security attributes from Master-CSPLight*

(2) *TSF in role Slave-CSPLight imports cryptographic keys with security attributes from Master-CSPLight only if the security attribute Clustering of the key allows transfer*[384].

[383] [assignment: *access control SFP, information flow control SFP*]
[384] [assignment: *additional importation control rules*]

FPT_ESA.1/CL Export of TSF data with security attributes - Cluster

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset informationflow control]
[FMT_MTD.1 Management of TSF data, or
FMT_MTD.3 Secure TSF data]
[FMT_MSA.1 Management of security attributes, or
FMT_MSA.4 Security attribute value inheritance]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_ESA.1.1/CL The TSF **in role Master-CSPLight** shall enforce the *Clustering SFP*[385] when exporting **Authentication Data Records and cryptographic keys** ~~TSF data~~, controlled under the SFP(s), ~~outside of the TOE~~ **to Slave-CSPLight**.

FPT_ESA.1.2/CL The TSF **in role Master-CSPLight** shall export the **Authentication Data Records and cryptographic keys** ~~TSF data~~ with the TSF data's associated security attributes.

FPT_ESA.1.3/CL The TSF **in role Master-CSPLight** shall ensure that the security attributes, when exported ~~outside the TOE~~ **to Slave-CSPLight**, are unambiguously associated with the exported **Authentication Data Records and cryptographic keys** ~~TSF data~~.

FPT_ESA.1.4/CL The TSF **in role Master-CSPLight** shall enforce the following rules when **Authentication Data Records and cryptographic keys** ~~TSF data~~ is exported ~~from the TOE~~ **to Slave-CSPLight:**

(1) *TSF in role Master-CSPLight exports Authentication Data Records with security attributes to any Slave-CSPLight*

(2) *TSF in role Master-CSPLight exports cryptographic keys with security attributes to Slave-CSPLight only if the security attribute Clustering of the key allows transfer*[386].

[385] [assignment: *access control SFP, information flow control SFP*]
[386] [assignment: *additional exportation control rules*]

FPT_TDC.1/CL Inter-TSF basic TSF data consistency - Clustering

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1/CL The TSF shall provide the capability to consistently interpret *Authentication Data Records and cryptographic keys with their security attributes*[387] when shared between the TSF and **TOE sample in the cluster** ~~another trusted IT product.~~

FPT_TDC.1.2/CL The TSF shall use *the following* rules:

(1) *the TSF in Slave-CSPLight role shall interpret the imported Authentication Data Records with their security attributes in the same way as it interprets the Authentication Data Records when it exports them in Master-CSPLight role,*

(2) *the TSF in Slave-CSPLight role shall interpret the imported cryptographic keys with their security attributes in the same way as it interprets the Authentication Data Records when it exports them in Master-CSPLight role,*[388]

when interpreting the **Authentication Data Records and cryptographic keys** ~~TSF data~~ from **Master-CSPLight** ~~another trusted IT product.~~

6.2 Security Assurance Requirements

The PP requires the TOE to be evaluated according to EAL2 augmented with ALC_CMS3 (Implementation representation CM coverage) and ALC_LCD.1 (Developer-Defined Lifecycle Model), and with specific refinements on ALC_CMS.3, ADV_ARC.1 and ATE_IND.2.

6.2.1 Assurance Refinements

Refinement on ALC_CMS.3.1C:

The implementation representation listed shall comprise the implementation representation of the TOE defining the TSF to a level of detail such that the compliance of the TOE and TSF to the requirements imposed by the platform guidances on which the TOE is designed to run on, can be verified by that evidence.

Refinement on ADV_ARC.1.3D:

The security guidance documentation of each platform (hardware platform and operating system) on which the TOE is designed to run shall be provided in addition.

[387] [assignment: *list of TSF data types*]
[388] [assignment: *list of interpretation rules to be applied by the TSF*]

Refinement on ADV_ARC.1.1C to 1.5C:

The security architecture description shall include an assessment how each single security requirement imposed by the platform documentation (guidance documentation and if available evaluation or certification results) has been followed in the TOE design and implementation concept.

Examples for such security requirements could include but are not limited to:

- **Dedicated library calls: Dedicated calls protecting against attacks may be provided by the platform for cryptographic operation. For example, dedicated calls implement operations that are hardened against timing side channel attacks, while others execute faster, but are not hardened. The platform guidance may require such library calls to be used.**
- **Key usage limitations: Key usage above a certain limit may reveal side channel information which can then be exploited. The implementation must ensure that the key usage limit is adhered to.**
- **Dedicated calls to ensure a correct program flow are provided (i.e. for boolean verification calls) to ensure protection against attacks that disturb the execution flow. Such library calls must be made use of in critical operations.**
- **Dedicated library calls are provided for the secure generation of cryptographic random numbers. Other random number generation functionality is present, but is not suitable to generate cryptographic random numbers. It must be ensured that correct random number generation library calls are used.**

Refinement on ADV_ARC.1.1E:

The evaluators task includes to check consistency of the requirements considered in the architectural description against those outlined in the platform documentation.

Refinement on ATE_IND.2.1D:

Providing the TOE for testing shall include in addition the implementation representation of the TOE as defined by ALC_CMS.3.

Refinement of ATE_IND.2.2C:

The resources provided shall include additionally appropriate tools or access to the TOE development environment in order to enable the evaluator to perform source code review most efficiently.

Refinement of ATE_IND.2.3E:

The evaluators test activities shall include a verification of the TOE implementation representation provided in order to confirm code compliance of the TOE implementation representation to the security guidance of the hardware platform and operating system and libraries which the TOE/TSF is intended to be run on. Therefore, the evaluator shall assess and verify that all platform guidance requirements are met and indicate possible vulnerabilities to the AVA evaluation activity for the TOE for further consideration.

6.3 Security Requirements Rationale

6.3.1 Dependency rationale

This chapter demonstrates that each dependency of the security requirements is either satisfied, or justifies the dependency not being satisfied.

Note, the column SFR components showing the concrete SFR satisfying the dependencies are typical use cases. It does not exclude that the SFR in the first column may solve dependencies of other SFR as well. E. g. the SFR FCS_CKM.1 defines requirements for ECC key generation, and a generated ECC key pair may not only be directly used for ECDSA digital signatures according to FCS_COP.1/CDS-RSA and FCS_COP.1/VDSRSA, but also for encryption and decryption of the AES key in FCS_COP.1/HEM and FCS_COP.1/HDM.

SFR	Dependencies of the SFR	SFR components
FCS_CKM.1/AES	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/ED FCS_CKM.4
FCS_CKM.1/AES_RSA	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/HEM with FCS_CKM.1/AES_RSA FCS_CKM.4
FCS_CKM.1/ECC	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/CDS-ECDSA FCS_COP.1/VDS-ECDSA, FCS_CKM.4
FCS_CKM.1/ECKA-EG	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/HEM with FCS_CKM.1/ECKA-EG, FCS_CKM.4
FCS_CKM.1/PACE	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/TCE FCS_COP.1/TCM, FCS_CKM.4
FCS_CKM.1/RSA	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/CDS-RSA FCS_COP.1/VDS-RSA, FCS_CKM.4

FCS_CKM.1/TCAP	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/TCE FCS_COP.1/TCM, FCS_CKM.4
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]	FCS_CKM.1/ECC FCS_CKM.1/RSA, FCS_CKM.1/ECKA-EG, FCS_CKM.1/AES_RSA FCS_CKM.1/TCAP, FCS_CKM.1/PACE
FCS_CKM.5/AES	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/ED FCS_CKM.4
FCS_CKM.5/AES_RSA	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/HDM with FCS_CKM.5/AES_RSA FCS_CKM.4
FCS_CKM.5/ECC	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/CDS-ECDSA, FCS_CKM.5/VDS-ECDSA, FCS_CKM.4
FCS_CKM.5/ECDHE	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/HEM with FCS_CKM.5/ECDHE, FCS_CKM.4
FCS_CKM.5/ECKA-EG	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/HEM with FCS_CKM.5/ECKA-EG, FCS_CKM.4
FCS_COP.1/CDS-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
FCS_COP.1/CDS-RSA	[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

FCS_COP.1/DecUCP	[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	Import of UCP decryption key as TSF data with confidentiality protection FPT_TCT.1/CK and FCS_COP.1/KU, FCS_CKM.4
FCS_COP.1/ED	[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
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	Hash functions do not use keys
FCS_COP.1/HDM	[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.5/ECKA-EG, FCS_CKM.5/AES_RSA, FCS_CKM.5/ECDHE (note deterministic FCS_CKM.5 play the role of randomized FCS_CKM.1) FCS_CKM.4
FCS_COP.1/HEM	[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/ECKA-EG, FCS_CKM.1/AES_RSA, FCS_CKM.5/ECDHE FCS_CKM.1/AES_RSA FCS_CKM.4
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_RNG.1 generates random strings as HMAC keys FCS_CKM.4
FCS_COP.1/KU	[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
FCS_COP.1/KW	[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

FCS_COP.1/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 MT_MSA.2 Secure security attributes	FCS_CKM.1/AES FCS_CKM.4
FCS_COP.1/TCE	[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/TCAP FCS_CKM.1/PACE, FCS_CKM.4
FCS_COP.1/TCM	[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/TCAP FCS_CKM.1/PACE, FCS_CKM.4
FCS_COP.1/VDS-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	FPT_ISA.1/Cert (note keys are TSF data), FCS_CKM.4
FCS_COP.1/VDS-RSA	[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	FPT_ISA.1/Cert (note keys are TSF data), FCS_CKM.4
FCS_COP.1/VDSUCP	[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	Import of signature verification key of UCP Issuer as TSF data FPT_ISA.1/Cert, FPT_TIT.1/Cert, FCS_CKM.4
FCS_RNG.1	No dependencies	
FDP_ACC.1/KM	FDP_ACF.1 Security attribute based access control	Dependency on FDP_ACF.1 is not fulfilled. Access control to key management functions are specified by FMT_MTD.1/KM because cryptographic keys are TSF data.
FDP_ACC.1/Oper	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Oper

FDP_ACC.1/UCP	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/UCP
FDP_ACF.1/Oper	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/Oper, FMT_MSA.3/KM
FDP_ACF.1/UCP	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/UCP, FMT_MSA.3 is not included, because the security attributes of UCP are imported according to FDP_ITC.2/UCP without default values
FDP_DAU.2/Att	FIA_UID.1 Timing of identification	FIA_UID.1
FDP_DAU.2/Sig	FIA_UID.1 Timing of identification	FIA_UID.1
FDP_ETC.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.1/Oper
FDP_ETC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.1/Oper
FDP_ITC.2/UCP	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1/UCP trusted communication is provided by FCS_COP.1/VDSUCP and FCS_COP.1/DecUCP, FPT_TDC.1/UCP
FDP_ITC.2/UD	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1/Oper trusted communication is provided by FCS_COP.1/HDM and FCS_COP.1/VDS-*, FPT_TDC.1/CK because import of user data is intended for cryptographic operation with key
FDP_RIP.1/UCP	No dependencies	
FIA_AFL.1	FIA_UAU.1 Timing of authentication	FIA_UAU.1
FIA_API.1/CA	No dependencies	
FIA_API.1/PACE	No dependencies	
FIA_ATD.1	No dependencies	
FIA_UAU.1	FIA_UID.1 Timing of identification	FIA_UID.1
FIA_UAU.5	No dependencies	
FIA_UAU.6	No dependencies	

FIA_UID.1	No dependencies	
FIA_USB.1	FIA_ATD.1 User attribute definition	FIA_ATD.1
FMT_MOF.1	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMF.1, FMT_SMR.1
FMT_MSA.1/KM	[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/KM, FDP_ACC.1/Oper, FMT_SMF.1, FMT_SMR.1
FMT_MSA.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FDP_ACC.1/KM, FDP_ACC.1/Oper, FMT_MSA.1/KM, FMT_SMR.1
FMT_MSA.3/KM	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1/KM, FMT_SMR.1
FMT_MTD.1/KM	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMF.1, FMT_SMR.1
FMT_MTD.1/RAD	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMF.1, FMT_SMR.1
FMT_MTD.1/RK	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMF.1, FMT_SMR.1
FMT_MTD.3	FMT_MTD.1 Management of TSF data	FMT_MTD.1/RAD
FMT_SAE.1	FMT_SMR.1 Security roles, FPT_STM.1 Reliable time stamps	FMT_SMR.1, dependency on FPT_STM.1 is not fulfilled, cf. to the application note to FMT_SAE1
FMT_SMF.1	No dependencies	
FMT_SMR.1	FIA_UID.1 Timing of identification	FIA_UID.1

FPT_ESA.1/CK	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data] [FMT_MSA.1 Management of security attributes, or FMT_MSA.4 Security attribute value inheritance] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1/KM, FMT_MTD.1/KM FMT_MSA.1/KM FPT_TDC.1/CK
FMT_FPT.1	No dependencies	
FPT_ISA.1/Cert	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data] [FMT_MSA.1 Management of security attributes, or FMT_MSA.4 Security attribute value inheritance] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1/KM, FMT_MTD.1/RK FMT_MSA.1/KM FPT_TDC.1/Cert
FPT_ISA.1/CK	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data] [FMT_MSA.1 Management of security attributes, or FMT_MSA.4 Security attribute value inheritance] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1/KM, FMT_MTD.1/RK FMT_MTD.1/KM FMT_MSA.1/KM FPT_TDC.1/Cert
FPT_TCT.1/CK	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]	FDP_ACC.1/KM, FMT_MTD.1/RK FMT_MTD.1/KM
FPT_TDC.1/Cert	No dependencies	
FPT_TDC.1/CK	No dependencies	
FPT_TDC.1/UCP	No dependencies	

FPT_TIT.1/Cert	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]	FDP_ACC.1/KM, FMT_MTD.1/RK
FPT_TIT.1/CK	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]	FDP_ACC.1/KM, FMT_MTD.1/KM
FPT_TST.1	No dependencies	
FTP_ITC.1	No dependencies	
FAU_GEN.1	FPT_STM.1 Reliable time stamps	FPT_STM.1
FAU_STG.1	FAU_GEN.1 Audit data generation	FAU_GEN.1
FAU_STG.3	FAU_STG.1 Protected audit trail storage	FAU_STG.1
FDP_ACF.1/TS	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/Oper in Base-PP, FMT_MSA.3 in Base-PP
FDP_DAU.2/TS	FIA_UID.1 Timing of identification	FIA_UID.1 in Base-PP
FDP_ETC.2/TS	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.1/Oper in Base-PP
FDP_ITC.2/TS	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1/Oper, trusted communication is provided by FCS_COP.1/HDM and FCS_COP.1/VDS-*, FTP_ITC.1, FPT_TDC.1/CK because import of user data is intended for cryptographic operation with key with appropriate security attribute “TimeStamp”, all these SFR in Base-PP
FMT_MOF.1/TSA	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1/TSA and FMT_SMR.1 in Base-PP, FMT_SMF.1/TSA
FMT_MTD.1/Audit	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1/TSA and FMT_SMR.1 in Base-PP, FMT_SMF.1/TSA
FMT_SMF.1/TSA	No dependencies	
FMT_SMR.1/TSA	FIA_UID.1 Timing of identification	FIA_UID.1 in Base-PP

FMT_STM.1	No dependencies	
FPT_TIT.1/Audit	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]	FDP_ACC.1/UCP in Base-PP and FDP_ACC.1/KM and FDP_ACC.1/Oper if selected, FMT_MTD.1/Audit
FAU_GEN.1/CL	FPT_STM.1 Reliable time stamps	FPT_STM.1 required by PPMTS-Au
FCS_CKM.5/CLDH	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/ED, FCS_COP.1/MAC and FCS_CKM.4 required in the Base-PP
FDP_ACC.1/CL	FDP_ACF.1 Security attribute based access control	FAU_STG.1
FMT_MTD.1/CL	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMF.1, FMT_SMR.1 required in the Base-PP
FPT_ESA.1/CL	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data] [FMT_MSA.1 Management of security attributes, or FMT_MSA.4 Security attribute value inheritance] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1/CL FMT_MTD.1/CL, FMT_MTD.1/RAD and FMT_MTD.1/KM required in the Base-PP, FMT_MSA.1/KM applies for exported and imported keys and required in the Base-PP, FPT_TDC.1/CL
FPT_ISA.1/CL	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data] [FMT_MSA.1 Management of security attributes, or FMT_MSA.4 Security attribute value inheritance] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1/CL FMT_MTD.1/CL, FMT_MTD.1/RAD and FMT_MTD.1/KM required in the Base-PP, FMT_MSA.1/KM applies for exported and imported keys and required in the Base-PP, FPT_TDC.1/CL
FPT_TCT.1/CL	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]	FDP_ACC.1/CL FMT_MTD.1/CL

FPT_TDC.1/CL	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	
FPT_TIT.1/CL	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FMT_MTD.1 Management of TSF data or FMT_MTD.3 Secure TSF data]	FDP_ACC.1/CL FMT_MTD.1/CL

Table 6.5: Dependency rationale (Table 6 in Base-PP [9])

6.3.2 Security functional requirements rationale

Table 6.6 traces each SFR back to the security objectives for the TOE.

	O.I&A	O.AuthentTOE	O.Enc	O.DataAuth	O.RBGS	O. TChann	O.AccCtrl	O.SecMan	O.TST	O.SecUpCP	O.Audit	O.TimeService	O.Cluster
FCS_CKM.1/AES			×	×				×					
FCS_CKM.1/AES_RSA			×	×				×					
FCS_CKM.1/ECC		×	×	×				×					
FCS_CKM.1/ECKA-EG			×	×				×					
FCS_CKM.1/PACE		×				×		×					
FCS_CKM.1/RSA		×	×	×				×					
FCS_CKM.1/TCAP		×				×		×					
FCS_CKM.4			×	×				×					
FCS_CKM.5/AES			×	×				×					
FCS_CKM.5/AES_RSA			×	×				×					
FCS_CKM.5/ECC			×	×				×					
FCS_CKM.5/ECDHE			×	×				×					
FCS_CKM.5/ECKA-EG			×	×				×					
FCS_COP.1/CDS-ECDSA		×		×									
FCS_COP.1/CDS-RSA		×		×									
FCS_COP.1/DecUCP										×			
FCS_COP.1/ED			×					×					

FCS_COP.1/Hash				×				×					
FCS_COP.1/HDM			×	×									
FCS_COP.1/HEM			×	×									
FCS_COP.1/HMAC		×		×									
FCS_COP.1/KU								×					
FCS_COP.1/KW								×					
FCS_COP.1/MAC				×									
FCS_COP.1/TCE						×							
FCS_COP.1/TCM						×							
FCS_COP.1/VDS-ECDSA				×									
FCS_COP.1/VDS-RSA				×									
FCS_COP.1/VDSUCP										×			
FCS_RNG.1					×			×					
FCS_ACC.1/KM							×	×					
FCS_ACC.1/Oper							×						
FCS_ACC.1/UCP										×			
FCS_ACF.1/Oper							×						
FCS_ACF.1/UCP										×			
FCS_DAU.2/Att		×											
FCS_DAU.2/Sig				×									
FCS_ETC.1				×									
FCS_ETC.2			×	×									
FCS_ITC.2/UCP										×			
FCS_ITC.2/UD			×	×									
FCS_RIP.1/UCP										×			
FIA_AFL.1	×												
FIA_AFI.1/CA	×	×				×							
FIA_AFI.1/PACE	×	×				×							
FIA_ATD.1	×						×	×					
FIA_UAU.1	×												
FIA_UAU.5	×					×							
FIA_UAU.6	×												

FIA_UID.1	×												
FIA_USB.1	×												
FMT_MOF.1	×					×							
FMT_MSA.1/KM			×	×		×	×	×					
FMT_MSA.2							×	×					
FMT_MSA.3/KM							×	×		×			
FMT_MTD.1/KM								×					
FMT_MTD.1/RAD	×												
FMT_MTD.1/RK	×		×	×				×					
FMT_MTD.3	×												
FMT_SAE.1	×												
FMT_SMF.1								×					
FMT_SMR.1	×							×					
FPT_ESA.1/CK								×					
FPT_FLS.1									×				
FPT_ISA.1/Cert	×			×				×		×			
FMT_ISA.1/CK								×					
FMT_TCT.1/CK								×		×			
FPT_TDC.1/CK			×	×				×					
FPT_TDC.1/Cert	×		×	×				×					
FPT_TDC.1/UCP										×			
FMT_TIT.1/Cert	×			×				×		×			
FMT_TIT.1/CK								×					
FPT_TST.1									×				
FMT_ITC.1						×							
FAU_GEN.1												×	
FAU_STG.1												×	
FAU_STG.3												×	
FDP_ACF.1/TS													×
FDP_DAU.2/TS												×	×
FDP_ETC.2/TS													×
FDP_ITC.2/TS													×

FMT_MOF.1/TSA													×
FMT_MTD.1/Audit												×	
FMT_SMF.1/TSA												×	×
FMT_SMR.1/TSA												×	×
FPT_STM.1												×	×
FPT_TIT.1/Audit												×	
FAU_GEN.1/CL												×	
FCS_CKM.5/CLDH													×
FDP_ACC.1/CL													×
FMT_MTD.1/CL													×
FPT_ESA.1/CL													×
FPT_ISA.1/CL													×
FPT_TCT.1/CL													×
FPT_TDC.1/CL													×
FPT_TIT.1/CL													×

Table 6.6: Security functional requirement rationale (Table 7 in Base-PP [9])

The following part of the chapter demonstrates that the SFRs meet all security objectives for the TOE. The security objective for the TOE O.I&A "Identification and authentication of users" is met by the following SFR:

- The SFR FIA_ATD.1 lists the security attributes *Identity, Authentication reference data and Role* belonging to individual users and the SFR FMT_SMR.1 defines the security roles maintained by TSF.
- The SFR FIA_USB.1 requires the TSF to associate the user security attributes *Identity and Role* with subjects acting on the behalf of that user.
- The SFR FIA_UID.1 defines the TSF-mediated actions allowed on behalf of Unidentified User.
- The SFR FIA_UAU.1 defines the TSF-mediated actions allowed on behalf of Unauthenticated User.
- The SFR FIA_UAU.5 requires the TSF lists the authentication mechanisms and the rules for their application.
- The SFR FIA_API.1/CA and FIA_API.1/PACE require the TSF to authenticate external entities using Chip Authentication and PACE to communication endpoints of trusted channels.

- The SFR FIA_UAU.6 requires the TSF to request re-authentication of users under the listed conditions.
- The SFR FMT_MOF.1 requires the TSF to enable and disable of human user authentication.
- The SFR FMT_MTD.1/RAD and The SFR FMT_MTD.1/RK defines the management function of and the access limitation to authentication mechanisms and their TSF data including the root public keys.
- The SFR FMT_MTD.3 enforce secure values for password mechanisms.
- The SFR FMT_SAE.1 requires the TSF to limit the validity of user authentication and reset the security attribute Role to a values defined by an administrator according to FMT_MTD.1/RAD.
- The SFR FIA_AFL.1 requires the TSF to detect and react on failed authentication attempts.
- The SFR FPT_ISA.1/Cert and FPT_TIT.1/Cert require the TSF to import certificates integrity protected and with their security attributes including those for entity authentication.
- The SFR FPT_TDC.1/Cert requires the TSF to interpret the certificates correctly.

The security objective for the TOE O.AuthentTOE "Authentication of the TOE to external entities" is met by the following SFR:

- The SFR FCS_CKM.1/ECC, FCS_CKM.1/RSA require the TSF to generate TOE authentication keys and SFR FCS_CKM.1/PACE and FCS_CKM.1/TCAP require the TSF to agree keys for authentication of the TOE to external entities.
- The SFR FCS_COP.1/CDS-ECDSA and FCS_COP.1/CDS-RSA require the TSF to generate digital signatures for authentication of the TOE to external entities.
- SFR FCS_COP.1/HMAC requires the TSF to generate HMAC for authentication of the TOE to external entities.
- The SFR FIA_API.1/CA, and FIA_API.1/PACE require the TSF to authenticate themselves using Chip Authentication, and PACE to communication endpoints of trusted channels.
- The SFR FDP_DAU.2/Att requires the TSF to generate evidence that can be used as a guarantee of the validity of attestation data to external entities.

The security objective for the TOE O.Enc "Confidentiality of user data by means of encryption and decryption" is met by the following SFR:

- The SFR FCS_CKM.1/ECC and FCS_CKM.1/RSA require (long term) key generation for the encryption and decryption security service of the TSF.

- The SFR FCS_CKM.1/AES, FCS_CKM.1/AES_RSA, FCS_CKM.5/ECDHE, and FCS_CKM.1/ECKA-EG, require key generation and FCS_CKM.5/AES, FCS_CKM.5/AES_RSA, FCS_CKM.5/ECKA-EG and FCS_CKM.5/ECC require key derivation for encryption and decryption security service of the TSF. Note the keys must be generated or agreed with the appropriate key type for encryption respectively for decryption or in case of symmetric cryptographic mechanisms for both according to FMT_MSA.1/KM.

- The FCS_COP.1/ED requires encryption and decryption as cryptographic operations for the encryption and decryption security service of the TSF.

- The FCS_COP.1/HDM requires hybrid decryption and the SFR FCS_COP.1/HEM requires hybrid encryption and decryption as cryptographic operations for the encryption and decryption security service of the TSF.

- The SFR FDP_ETC.2 require the TSF to export encrypted user data with reference to the key and data integrity checksums for decryption and FDP_ITC.2/UD require import of encrypted user data with reference to decryption key and data integrity checksums for decryption.

- The SFR FCS_CKM.4 requires the TSF to implement secure key destruction.

- The SFR FMT_MTD.1/RK requires the TSF management of root keys for key hierarchy known to the TSF if used for encryption.

- The SFR FPT_TDC.1/Cert requires the TSF to interpret consistently the security attributes of certificates (including those used for encryption and decryption).

- The SFR FPT_TDC.1/CK requires the TSF to interpret consistently the security attributes of keys (including those used for encryption and decryption).

The security objective for the TOE O.DataAuth "Data authentication by cryptographic mechanisms" is met by the following SFR:

- The SFR FCS_CKM.1/ECC and FCS_CKM.1/RSA require (long term) key generation for the signature security service of the TSF. The SFR FCS_CKM.1/AES, FCS_CKM.1/ECKA-EG, FCS_CKM.1/AES_RSA require key generation and FCS_CKM.5/AES_RSA, FCS_CKM.5/ECDHE, FCS_CKM.5/ECC, FCS_CKM.5/ECKA-EG key derivation for MAC generation and verification. Note the keys must be generated or agreed with the appropriate key type for signature-creation, signature-verification or, in case of symmetric cryptographic mechanisms for data authentication according to FMT_MSA.1/KM.

- The SFR FDP_ETC.2 require the TSF to export signed data with and signature and public key reference for signature verification and FDP_ITC.2/UD import of signed data with signature and public key reference for signature verification. The SFR FDP_ETC.1 require the TSF to export successfully MAC verified and decrypted ciphertext as plaintext according to FCS_COP.1/HDM without the user data's associated security attributes:

- The SFR FCS_COP.1/Hash requires the TSF to implement cryptographic primitive hash function used for HMAC, cf. FCS_COP.1/HMAC, digital signature creation, cf. FCS_COP.1/CDS-*and digital signature verification, cf. FCS_COP.1/VDS-*.

- The FCS_COP.1/CDS-ECDSA and FCS_COP.1/CDS-RSA require asymmetric cryptographic mechanisms for signature-creation.
- The SFR FCS_COP.1/VDS-ECDSA and FCS_VDS/RSA require asymmetric cryptographic mechanisms for signature-verification.
- The SFR for keyed hash FCS_COP.1/HMAC and block cipher based MAC FCS_COP.1/MAC require the TSF to provide symmetric data integrity mechanisms.
- The SFR FCS_COP.1/HEM requires hybrid MAC calculation and FCS_COP.1/HDM requires hybrid MAC verification for the ciphertext as security service of the TSF.
- The SFR FPT_ISA.1/Cert requires import of certificates with security attributes and integrity protection according to FPT_TIT.1/Cert.
- The SFR FCS_CKM.4 requires the TSF to implement secure key destruction.
- The SFR FPT_TDC.1/Cert requires the TSF to interpret consistently the security attributes in certificates (including those used for data authentication).
- The SFR FPT_TDC.1/CK requires the TSF to interpret consistently the security attributes keys (including those used for data authentication).

The security objective for the TOE O.TChann "Trusted channel" is met by the following SFR:

- The SFR FTP_ITC.1 requires different types of trusted channel depending on the capability of the other endpoint. The cases are defined in Table 6.3. The remote entity and the TOE may use mutual authentication and key agreement by means of PACE according to FCS_CKM.1/PACE, shall provide integrity protection according to FCS_COP.1/TCM and may support confidentiality of the communication data according to FCS_COP.1/TCE. The cases 3 requires support of trusted channel with mutual authentication by FIA_API.1/CA, FIA_UAU.5, key agreement TCAP according to FCS_CKM.1/TCAP, encryption and MAC data authentication.
- The TOE authenticate themselves according to FIA_API.1/PACE in case of PACE. It authenticates themselves according to FIA_API.1/CA in case of TCAP as Proximity Integrated Circuit Card (PICC).
- The SFR FMT_MOF.1 limits the configuration of the trusted channel according to FTP_ITC.1.3 to an administrator.
- The SFR FMT_MSA.1/KM describe the requirements for management of key security attributes for these mechanisms.

The security objective for the TOE O.AccCtrl "Access control" is met by the following SFR:

- The SFR FIA_ATD.1 defines the security attributes of individual users including Role which is used for access control according to FDP_ACF.1/Oper.
- The SFR FDP_ACC.1/Oper describes the subset access control for the Cryptographic Operation SFP.
- The SFR FDP_ACF.1/Oper defines the access control rules of the Cryptographic Operation SFP.

- The Cryptographic Operation SFP is defined by means of security attributes managed according to the SFR FMT_MSA.1/KM, FMT_MSA.2 and FMT_MSA.3/KM.

The security objective for the TOE O.SecMan "Security management" is met by the following SFR:

- The SFR FIA_ATD.1 defines the security attributes of individual users including Role which is used to enforce the Key Management SFP.
- The SFR FDP_ACC.1/KM defines subjects, objects and operations of the Key Management SFP.
- The SFR FMT_SMF.1 lists the security management functions provided by the TSF.
- The SFR FMT_SMR.1 lists the security role supported by the TOE especially the administrator and - if supported - Crypto-Officer responsible for key management.
- The SFR FCS_CKM.1/AES, FCS_CKM.1/ECC, FCS_CKM.1/ECKA-EG. FCS_CKM.1/PACE, FCS_CKM.1/RSA, FCS_CKM.1/AES_RSA, FCS_CKM.1/TCAP require the TSF to implement key generation function according to the assigned standards.
- The SFR FCS_CKM.5/ECDHE require the TSF to implement key agreement function according to the assigned standards.
- The SFR FCS_CKM.5/AES and FCS_CKM.5/ECKA-EG require the TSF to implement key derivation function according to the assigned standards.
- The SFR FCS_CKM.1/AES_RSA and FCS_CKM.5/AES_RSA require the TSF to implement AES session key generation function with RSA key encryption respective RSA key decryption and AES key derivation according to the assigned standards.
- The SFR FCS_RNG.1 requires the TSF to implement a random number generator for key generation, key agreement functions and cryptographic operations.
- The SFR FCS_COP.1/ED requires the TSF to provide encryption and decryption according to AES which may be used for key management.
- The SFR FCS_COP.1/Hash requires the TSF to implement cryptographic primitive hash function for key derivation, cf. FCS_CKM.5.
- The SFR FPT_ISA.1/CK requires import and FPT_ESA.1/CK the export of cryptographic keys with security attributes and protection of confidentiality according to SFR FPT_TCT.1/CK and integrity protection according to FPT_TIT.1/CK.
- The SFR FPT_ISA.1/Cert requires import of certificates with security attributes and integrity protection according to FPT_TIT.1/Cert.
- The SFR FPT_TDC.1/Cert requires consistent interpretation of certificate's content. The SFR FPT_TDC.1/ CK requires consistent interpretation of security attributes imported with the key.
- The SFR FCS_COP.1/KW and FCS_COP.1/KU require the TSF key wrapping and unwrapping for key management.
- The SFR FCS_CKM.4 requires the TSF to implement secure key destruction.

- The SFR FMT_MSA.1/KM and FMT_MSA3/KM limit the setting of default values and specification of alternative initial values for security attributes of cryptographic keys to administrators. The SFR FMT_MSA.1/KM prevents modification or deletion of security attributes of keys.
- FMT_MSA.2 enforce secure values for security attributes.
- The SFR FMT_MTD.1/KM and FMT_MTD.1/RK restricts the management of cryptographic keys espacially the import of root public keys to specifically authorized users.

TOE O.TST "Self-test" is directly met by the SFR FPT_TST.1 and FPT_FLS.1. The TSF shall preserve a secure state if self test fails.

The security objective for the TOE O.SecUpCP "Secure import of Update Code Package" is met by the following SFR:

- The SFR FDP_ACC.1/UCP and FDP_ACF.1/UCP requires the TSF to provide access control to enforce SFP *Update*. Note the verification of the authenticity of UCP and decryption of authentic UCP are performed under control of the TSF.
- The SFR FCS_COP.1/VDSUCP requires the verification of digital signature of the Issuer and FCS_COP.1/ DecUCP requires decryption of authentic of UCP.
- The SFR FDP_ITC.2/UCP requires the TSF to import UCP as user data with security attributes if the authenticity of UCP is successful verified.
- The SFR FPT_TDC.1/UCP requires the TSF to import consistently the security attributes of the UCP.
- The SFR FMT_MSA.3 requires to provide restrictive initial security attributes to enforce the SFP Update.
- The SFR FDP_RIP.1/UCP requires the TSF to remove the received UCP after unsuccessful verification of its authenticity.
- The UCP signature verification key may be updated according to FPT_ISA.1/Cert with integrity protection according to FPT_TIT.1/Cert.
- The UCP decryption key may be updated with confidentiality protection according to FPT_TCT.1/CK with FCS_COP.1/KU.

The security objective for the TOE O.TimeService “Time services” is met by the following SFR:

- The SFR FPT_STM.1 requires the TSF to provide time stamps for the real time service.
- The SFR FDP_DAU.2/TS requires the TSF to provide cryptographic protected time stamps for time stamp service supported by FCS_COP.1/CDS-ECDSA resp. FCS_COP.1/CDS-RSA for signature creation defined in the Base-PP.
- The SFR FDP_ACF.1/TS defines access control on time stamp service to enforce the Cryptographic Operation SFP defined in the Base-PP.
- The SFR FDP_ITC.2/TS for user data import with security attributes indicating the signature key for time stamps.

- The SFR FDP_ETC.2/TS requires the TSF to export user data with time stamps.
- The SFR FMT_SMF.1/TSA defines the management functions and FMT_SMR.1/TSA the roles for the time service and the time stamp service additional to those defined in the Base-PP.
- The SFR FMT_MOF.1/TSA defines the management of the time service and the time service TSF.

The security objective for the TOE O.Audit "Audit" is met by the following SFR:

- The SFR FAU_GEN.1 requires the TSF to generate the audit records of auditable events.
- The SFR FAU_STG.1 and FAU_STG.3 requires the TSF to protect and to prevent loss of audit records.
- The SFR FMT_MTD.1/Audit restricts the ability to export and to delete exported audit records to an Administrator. It prevents undetected deletion of audit records by generation of an audit record about deletion. The export, clear and selection of events causing audit data as management TSF data is an auditable event, cf. FAU_GEN.1, clause (11).
- The SFR FPT_TIT.1/Audit requires the TSF to protect audit records when transmitted and time when imported.
- The SFR FMT_SMF.1/TSA defines the management functions and FMT_SMR.1/TSA the roles for the audit TSF additional to those defined in the Base-PP.
- The SFR FMT_MOF.1/TSA requires the TSF to provide the capability to define the auditable events in clause (3) and the behaviour of automatic export of audit records in clause (4).
- The SFR FDP_DAU.2/TS requires the TSF to provide the capability to export audit trails signed and time stamped.
- The SFR FPT_TIT.1/Audit defines the TSF data integrity transfer protection for the audit functionality
- The SFR FPT_STM.1 requires the TSF to provide time stamps being part of the audit records

The security objective for the TOE O.Audit "Audit" is met by the SFR FAU_GEN.1 in PPM-TS-AU and additionally by SFR FAU_GEN.1/CL to generate the audit records of auditable events for clustering.

The security objective for the TOE O.Cluster "Cluster" is met by the following SFR:

- The SFR FDP_ACC.1/CL defines subjects, objects and operations of the Clustering SFP.
- The SFR FMT_MTD.1/CL restricts the management of TSF data Authentication Data Records and cryptographic key by initiating the cluster to an administrator, and export and import of TSF data to an authorised identified role.
- The SFRs FPT_ESA.1/CL and FPT_ISA.1/CL require that export and import of TSF data is performed with security attributes.

- The SFR FPT_TCT.1/CL requires protection of confidentiality and the SFR FPT_TIT.1/CL the protection of integrity of the TSF data when transferred between Master-CSPLight and Slave-CSPLight.
- The SFR FCS_CKM.5/CLDH requires the TSF to agree on cryptographic keys. Note, the Base-PP defines the SFRs FCS_COP.1/ED and FCS_COP.1/MAC for encryption and MAC of the transferred TSF data.
- The SFR FPT_TDC.1/CL requires the TSF interpret consistently the TSF exchanged between TOE samples of the cluster.

6.3.3 Security assurance requirements rationale

Developers and users require for the TOE a low to moderate level of independently assured security in the absence of ready availability of the complete development record.

The EAL2 was chosen because it provides assurance by a full security target and an analysis of the SFRs in that ST, using a functional and interface specification, guidance documentation and a basic description of the architecture of the TOE, to understand the security behaviour. The analysis is supported by independent testing of the TSF, evidence of developer testing based on the functional specification, selective independent confirmation of the developer test results, and a vulnerability analysis (based upon the functional specification, TOE design, security architecture description and guidance evidence provided) demonstrating resistance to penetration attackers with a basic attack potential. EAL2 also provides assurance through use of a configuration management system and evidence of secure delivery procedures.

ALC_CMS.3 has been augmented to include the implementation representation as needed for ADV_ARC and ATE_IND refinements and to get evidence that the implementation representation provided is the one of the TOE. This means that the implementation representation is part of the configuration list.

CSPLight usually requires an initial configuration and/or the installation of key material and trust certificates. Hence, ALC_LCD.1 has been augmented such that the lifecycle of the TOE is defined by the developer and thus made explicit.

For getting confidence that the platform of the TOE (operational environment) is used by the TOE in a way that the requirements on security as outlined in the platform documentation (guidance documentation and if available evaluation or certification results) have been followed in the TOE design and implementation, refinements of ADV_ARC, ATE_IND and ALC_CMS have been defined.

The refinement of ADV_ARC ensures that the developer outlines how he has considered the requirements from the platform within his TOE security architecture and design concept. The evaluators task is to check consistency of the requirements considered against those outlined in the platform documentation.

As a second step of verification that the relevant platform requirements have been considered correctly, the independent evaluator activity at ATE_IND has been refined. The evaluator has to perform a specific "source code review", by means of cross check of the requirements from the platform to the implementation representation of the TOE by examine the implementation representation of the TOE using appropriate tools and the evidence from ADV_ARC.

Chapter 7

TOE Summary Specification

The following section describes how the TOE meets each SFR. For that reason, the SFRs are assigned to Security Functions (SF) provided by the TOE.

7.1 SF_KeyManagement

The TOE provides key management functionality including

- a deterministic random number generator,
- management of a key's security attributes,
- access to hash based functions,
- management of certificates,
- functions related to a key's life-cycle (key generation, key derivation and key destruction),
- import and export functionality.

Random Number Generation

The TOE implements a deterministic random number generator that is compliant with the requirements of class DRG.3 of [2]. The implementation follows the example of the iterated hash RNG in *Example 39* of [2] which is further defined in [21]. The RNG is seeded by obtaining random data from its hardware platform (which is the PrimeKey SEE that has a certified TRNG)

Key Management

According to FDP_ACC.1/KM, FMT_MSA.1/KM, FMT_MSA.3/KM and FMT_MTD.1/KM the TOE must implement several management functions with regards to cryptographic keys, and it must enforce access control security functional policies on the cryptographic keys. The TOE provides the functionality by operations for key creation, key derivation, key deletion, key property modification, key import and key export that are exposed by an external interface. The program logic enforces the required access control and access restrictions.

This SF supports several other SF, namely

- SF_DataEncryption (cf. Section 7.2),
- SF_HybridEncryptionWithMAC (cf. Section 7.3),
- SF_DataIntegrityMechanisms (cf. Section 7.4),
- SF_TOEAuthenticationAttestationTrustedChannel (cf. Section 7.5),
- SF_UserIdentificationAuthentication (cf. Section 7.6),
- SF_UpdateCodePackage (cf. Section 7.10),
- SF_Timestamping (cf. Section 7.11),
- SF_Clustering (cf. Section 7.12).

Key Lifecycle

Key generation is only allowed to users acting in an appropriate role. The only role allowed to generate new keys is that of the *Crypto-Officer*. After key generation, the *Crypto-Officer* can delegate the ownership of the new key to a *Key Owner* for usage. The generated keys are stored in the database of the TOE.

The TOE supports key generation and derivation using

- AES,
- Elliptic Curve Cryptography, and
- RSA.

Key generation always uses the DRG.3 random number generator provided by the TOE.

While the TOE protects all key material by using an encrypted storage mechanism and a tamper protected hardware platform, an explicit key destruction function exists, which allows the zeroisation of a key. The same role limitation as with *Key Generation* applies to *Key Destruction*. If that operation is triggered by an authenticated *Crypto-Officer*, the following steps will be executed:

(1) Overwrite the key with zeroes
(2) Delete the associated entry from the database

Restoration of a deleted key is not possible.

Also, whenever a key is stored temporarily in memory during usage, it overwritten with zero-bytes before its memory location is being released.

Certificate management

The certificate management capabilities of the TOE provide:

- Management of root public key of a PKI
- Verification of the integrity of certificates
- Import of TSF data from valid certificates

Whenever a certificate is imported into the TOE, its validity is verified:

- Check revocation status
- Check if it chains to the PKI root public key

Certificates manages by the TOE are periodically validated for their revocation status by the TOE.

Key export

The TOE supports key wrapping for secure key export from the TOE and key import to the TOE in line with FCS_COP.1/KW and FCS_COP.1/KU. The TOE implements this using a hardened cryptographic library. First, a wrapping key has to be generated which will in turn be used to encrypt some other key before it is exported from the TOE. Exporting a key without encrypting it first is not supported by the TOE. Both of these steps necessary for exporting a key are only executable by an authenticated user acting in the *Crypto-Office* role. The TOE maintains all key-wrap keys. TOE samples that are part of the same cluster may share the same (set of) key-wrap key(s).

Key agreement

The TOE supports mutual key agreement based on

- Elliptic Curve Diffie-Hellman, and
- ECKA-EG.

Key agreement is necessary for trusted channel establishment, clustering (cf. Section 7.12) and authentication (cf. Section 7.6). It also depends on random number generation. The TOE implements this using a hardened cryptographic library.

Auxiliary functions

The TOE provides a random number generation service of class DRG.3 for security module applications and for internal use (for key generation and key agreement).

The TOE supports hashing of data (FCS_COP.1/Hash) using

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

Hashing is a cryptographic primitive used by the TOE internally for HMAC (cf. FCS_COP.1/HMAC), and digital signature creation and verification (cf. FCS_COP.1/CDS-* and FCS_COP.1/VDS-*), and the hashing functionality is provided as a service (e.g. for use by application components). The TOE implements this using a hardened cryptographic library.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.1.

7.2 SF_DataEncryption

The TOE supports symmetric data encryption and decryption using AES in CBC mode with cryptographic key lengths of 128 and 256 bits in accordance with FCS_COP.1/ED. The TOE implements this using a hardened cryptographic library. This functionality is used by the TOE internally for data transfer confidentiality protection in a CSPLight cluster setup (cf. FPT_TCT.1/CL). This functionality is also provided by the TOE to SMAs by an external interface for encryption and decryption. The *Initialisation Vector (IV)* required to perform the encryption function will always be generated from fresh random data.

If hybrid encryption using an asymmetric key is possible, instead of this TSF, the *hybrid encryption/decryption* function (see section 7.3) should be used since it implements *authenticated encryption* directly.

This SF is dependent on the prior generation of a key as provided by SF_KeyManagement (cf. Section 7.1). Key generation requires the role of *Crypto-Officer*. Access to the secret key required for encryption and decryption requires either the *Crypto-Officer* or the *Key Owner* role.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.2.

7.3 SF_HybridEncryptionWithMAC

The TOE provides hybrid data encryption where a randomly generated secret key is generated and encrypted using an asymmetric keypair. The TOE implements this using a hardened cryp-

Security Function Group	Mapped Security Functional Requirements
SF_KeyManagement - Management of security attributes	FDP_ACC.1/KM (Subset access control - Cryptographic operation) FMT_MSA.1/KM (Management of security attributes - Key security attributes) FMT_MSA.3/KM (Static attribute initialisation - Key management) FMT_MTD.1/KM (Management of TSF data - Key management)
SF_KeyManagement - Hash Based Functions	FCS_COP.1/Hash (Cryptographic operation - Hash)
SF_KeyManagement - Management of Certificates	FMT_MTD.1/RK (Management of TSF data - Root key) FPT_TIT.1/Cert (TSF data integrity transfer protection - Certificates) FPT_ISA.1/Cert (Import of TSF data with security attributes - Certificates) FPT_TDC.1/Cert (Inter-TSF basic TSF data consistency - Certificate)
SF_KeyManagement - Key generation, agreement and destruction	FCS_RNG.1 Random number generation) FCS_CKM.1/AES (Cryptographic key generation - AES key) FCS_CKM.5/AES (Cryptographic key derivation - AES key derivation) FCS_CKM.1/ECC (Cryptographic key generation - Elliptic curve key pair ECC) FCS_CKM.5/ECC (Cryptographic key derivation - ECC key pair derivation) FCS_CKM.1/RSA (Cryptographic key generation - RSA key pair) FCS_CKM.5/ECDHE (Cryptographic key derivation - Elliptic Curve Diffie-Hellman ephemeral key agreement) FCS_CKM.1/ECKA-EG (Cryptographic key generation - ECKA-EG key generation with ECC encryption) FCS_CKM.5/ECKA-EG (Cryptographic key derivation - ECKA-EG key derivation) FCS_CKM.1/AES_RSA (Cryptographic key generation - Key generation and RSA encryption) FCS_CKM.5/AES_RSA (Cryptographic key derivation - RSA key derivation and decryption) FCS_CKM.4 (Cryptographic key destruction)
SF_KeyManagement - Key import and export	FCS_COP.1/KW (Cryptographic operation - Key wrapped) FCS_COP.1/KU (Cryptographic operation - Key unwrap) FPT_TCT.1/CK (TSF data confidentiality transfer protection - Cryptographic key) FPT_TIT.1/CK (TSF data integrity transfer protection - Cryptographic keys) FPT_ISA.1/CK (Import of TSF data with security attributes - Cryptographic keys) FPT_TDC.1/CK (Inter-TSF basic TSF data consistency - Key import) FPT_ESA.1/CK (Export of TSF data with security attributes - Cryptographic keys)

Table 7.1: Mapping of *Security Function SF_KeyManagement* to *Security Functional Requirements (SFR)*

Security Function Group	Mapped Security Functional Requirements
SF_DataEncryption	FCS_COP.1/ED (Cryptographic operation - Data encryption and decryption)

Table 7.2: Mapping of *Security Function SF_DataEncryption* to *Security Functional Requirements (SFR)*

tographic library. This combines the usability advantages of asymmetric encryption with the performance benefits of symmetric cryptography.

The symmetric encryption of the actual data is always done by using an *Authenticated Encryption* scheme using the *Encrypt-then-MAC* semantic as required by FCS_COP.1/HEM and FCS_COP.1/HDM. Either the integrated GCM mode can be used or a combination of CBC encryption with CMAC authentication.

The external interface provides functions for encrypting and decrypting data. Access to the asymmetric key required for decryption requires either the *Crypto-Officer* or the *Key Owner* role.

This SF is dependent on the prior generation of a key as provided by SF_KeyManagement (cf. Section 7.1). Key generation requires the role of *Crypto-Officer* while the usage of those keys can be done by a user either in the *Key Owner* or *Crypto-Officer* role.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.3.

Security Function Group	Mapped Security Functional Requirements
SF_HybridEncryption-WithMAC	FCS_COP.1/HEM (Cryptographic operation - Hybrid data encryption and MAC calculation) FCS_COP.1/HDM (Cryptographic operation - Hybrid data decryption and MAC verification)

Table 7.3: Mapping of *Security Function SF_HybridDataEncryption-WithMAC* to *Security Functional Requirements (SFR)*

7.4 SF_DataIntegrityMechanisms

The TOE supports data integrity protection via symmetric as well as via asymmetric cryptography. The TOE implements this using a hardened cryptographic library.

Both the methods for creating and verifying the integrity verification data can be accessed via the function of the external interface of the TOE.

This SF is dependent on the prior generation of keys as provided by SF_KeyManagement (cf. Section 7.1). Key generation requires the role of *Crypto-Officer* while the usage of those keys (creation or verification) can be done by a user either in the *Key Owner* or *Crypto-Officer* role. Signature creation and verification with timestamping keys may only be performed by entities in the *Application Component* role.

Message Authentication Codes

The symmetric *Message Authentication Code (MAC)* algorithms supported (FCS_COP.1/MAC and FCS_COP.1/HMAC) are

- HMAC
- CMAC

All of those are requiring a key of at least 128 bits in size.

HMAC is used exclusively in combination with the *Hash Function* SHA-256 yielding a tag length of 32 bytes.

The output length of CMAC, which uses the block cipher AES, is always 16 bytes.

Digital Signatures

The TOE supports the creation and verification of *Digital Signatures* via asymmetric cryptographic keys (FCS_COP.1/CDS-RSA, FCS_COP.1/VDS-RSA, FCS_COP.1/CDS-ECDSA, FCS_COP.1/VDS-ECDSA, FDP_DAU.2/Sig).

The following algorithms are supported:

- the Elliptic Curve Digital Signature Algorithm (ECDSA),
- RSA.

The usable key sizes are fixed for both algorithms: A 256 bit elliptic curve key (SECP256r1) for ECDSA and a 4096 bit modulus for RSA. All of those key lengths are deemed appropriate for long-term security by [13].

RSA signatures always use PSS Padding (PKCS#1 v2.1).

The nonce required to create the ECDSA signature is created using the internal random number generator (FCS_RNG.1).

The functionality *(Data Authentication with Identity of Guarantor (FDP_DAU.2/Sig)* is only available for keys which have a digital certificate obtained via the PKI associated.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.4.

Security Function Group	Mapped Security Functional Requirements
SF_DataIntegrity-Mechanisms	FCS_COP.1/MAC (Cryptographic operation - MAC using AES) FCS_COP.1/HMAC (Cryptographic operation - HMAC) FCS_COP.1/CDS-ECDSA (Cryptographic operation - Creation of digital signatures ECDSA) FCS_COP.1/VDS-ECDSA (Cryptographic operation - Verification of digital signatures ECDSA) FCS_COP.1/CDS-RSA (Cryptographic operation - Creation of digital signatures RSA) FCS_COP.1/VDS-RSA (Cryptographic operation - Verification of digital signatures RSA) FDP_DAU.2/Sig (Data Authentication with Identity of Guarantor - Signature)

Table 7.4: Mapping of *Security Function SF_DataIntegrityMechanisms* to *Security Functional Requirements (SFR)*

7.5 SF_TOEAuthenticationAttestationTrustedChannel

The TOE supports trusted channel establishment with

- PACE,
- Terminal Authentication 2, and
- Chip Authentication 2.

The trusted channel supports

- authenticity,
- integrity, and
- (optionally) confidentiality.

The TOE implements trusted channel establishment according to FTP_ITC.1, FIA_API.1/PACE and FIA_API.1/CA, using key agreement according to FCS_CKM.1/PACE and FCS_CKM.1/TCAP which is exposed by an external interface. PACE is established with Curve P-256 with cryptographic key sizes of 128 bits or 256 bits. Trusted channel establishment is supported by the random number generator (cf. Section 7.1) and key agreement (cf. Section 7.1) as provided by SF_KeyManagement (cf. Section 7.1).

Requests to an external interface of the TOE that require integrity and confidentiality protection undergo a MAC verification according to FCS_COP.1/TCM first. If the MAC verification is successful, the data is decrypted according to FCS_COP.1/TCE, and the requested operation is carried out. The corresponding responses of the TOE are encrypted and integrity protected according to FCS_COP.1/TCE and FCS_COP.1/TCM, respectively.

The TOE supports attestation via an external interface according to FDP_DAU.2/Att using ECDSA. This feature makes it possible for external entities (e.g., the PKI) to find out whether the TOE sample is genuine (e.g., in association with certificate signing requests sent to the PKI to attest that a genuine TOE sample has made the request). This functionality is based on the creation of digital signatures using ECDSA in accordance with FCS_COP.1/CDS-ECDSA as provided by SF_DataIntegrityMechanisms (cf. Section 7.4).

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.5.

Security Function Group	Mapped Security Functional Requirements
SF_TOEAuthentication-AttestationTrusted-Channel	FIA_API.1/PACE (Authentication Proof of Identity - PACE authentication to Application component) FIA_API.1/CA (Authentication Proof of Identity - Chip authentication to user) FDP_DAU.2/Att (Data Authentication with Identity of Guarantor - Attestation) FTP_ITC.1 (Inter-TSF trusted channel) FCS_CKM.1/PACE (Cryptographic key generation - Key agreement for trusted channel PACE) FCS_CKM.1/TCAP (Cryptographic key generation - Key agreement by Terminal and Chip authentication protocols) FCS_COP.1/TCE (Cryptographic operation - Encryption for trusted channel) FCS_COP.1/TCM (Cryptographic operation - MAC for trusted channel)

Table 7.5: Mapping of *Security Function SF_TOEAuthenticationAttestationTrustedChannel* to *Security Functional Requirements (SFR)*

7.6 SF_UserIdentificationAuthentication

The TOE stores the authentication reference data and roles of an entity or user in a persistent storage (cf. FIA_ATD.1).

The User Administrator may create, delete, modify authentication reference data and permanent session key for users and entities in accordance with FMT_MTD.1/RAD (1), FMT_MTD.1/RAD (2), FMT_MTD.1/RAD (3), FMT_MTD.1/RAD (4).

The User Administrator may define time limits for roles in accordance with FMT_MTD.1/RAD (5). A session is terminated once one of its roles expires (conceptually this is a reset to role Unidentified User in accordance with FMT_MTD.1/RAD (6)). The TOE persistently stores timestamps (cf. SF_Timestamping in Section 7.11) to realize this behavior.

At the first successful authentication, the password must be changed to a different secure operational password (cf. FMT_MTD.3). This is enforces by marking the password in the authentication data record as initial password by a Boolean flag. Also the minimal length of the password must be 12 characters.

The TOE supports blocking of users after 1 to 10 unsuccessful authentication attempts. The actual number is definable be the User Administrator and persisted in a settings table in the persistent storage. The User Administrator also defines for how long the user or entity shall be blocked by the TOE if the number of unsuccessful authentication attempts is met (the allowed duration must be between 1 and 60 minutes). This setting is also persistently stored by the TOE. If the number of unsuccessful authentication attempts is met, the TOE stores a timestamp of this event (cf. SF_Timestamping in Section 7.11). The TOE checks at each re-authentication attempt if the difference of the current timestamp and the recorded timestamp exceeds the blocking timespan, and only than is a re-authentication attempt allowed by the TOE.

The TOE manages the binding of roles to a user/entity in an active session context in accordance with FIA_USB.1. Initially the role is defined as Unidentified User. Once the identity (also called entity ID) is provided to the TOE and the entityID is in the set of persistently stored entityIDs, the user/entity is associated with the role Unauthenticated User. Once the correct credentials are provided to the TOE (i.e. the TOE checks the provided against the expected credentials in the authentication data record) and the every role in the set of claimed roles is also associated with the user/entity in the persistent database of the TOE, the role(s) of the user are changed to to the claimed role(s).

The TOE supports time-limited authorization (cf. FMT_SAE.1) by terminating active sessions after the expiration time (defined per role by the User Administrator) is exceeded. For that purpose, the TOE uses its timestamping capabilities (cf. ST_Timestamping in Section 7.11).

The TOE enforces that no other actions than the self test (cf. SF_TSFProtection in Section 7.9) and the identification of the TOE to the user may happen before user identification (cf. FIA_UID.1). This is the case if no session context exists in the TOE for an incoming request.

The TOE enforces that no other actions than the self test (cf. SF_TSFProtection in Section 7.9) and the identification of the user to the TOE and a selection of a set of roles for authentication may happen before authentication. This is the case if the session context for an incoming request is in the unauthenticated state. Only after successful authentication of the user to the TOE, an authentication of the TOE to the user may take place (cf. SF_TOEAuthenticationAttestationTrustedChannel in Section 7.5).

The TOE supports multiple authentication mechanisms (cf. FIA_UAU.5) as follows. The TOE supports password authentication for physically present users via keyboard inputs and changing the initial password to a secure operation password after the first successful authentication. Other actions are carried out via the HTTP interface provided by the TOE after successful trusted channel establishment. The TOE implements PACE, Terminal Authentication 2, and Chip Authentication 2 for trusted channel establishment and attestation (cf. SF_TOEAuthenticationAttestationTrustedChannel in Section 7.5). The TOE performs MAC verification for established trusted channels (SF_DataIntegrityMechanisms in Section 7.4).

The TOE holds active sessions in volatile memory. Therefore, power on or reset terminates active sessions (cf. FIA_UAU.6.1 (2)). The TOE does not support changing the role of an active session. Selecting the roles of a session is only possible during session establishment. Thus, FIA_UAU.6.1 (1) is met. All messages received via a trusted channel are authenticated by successful MAC verification (cf. FIA_UAU.6.1 (3)).

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.6.

Security Function Group	Mapped Security Functional Requirements
SF_UserIdentification-Authentication	FIA_ATD.1 (User attribute definition - Identity based authentication) FMT_MTD.1/RAD (Management of TSF data - Authentication reference data and Authentication Data Records) FMT_MTD.3 (Secure TSF data) FIA_AFL.1 (Authentication failure handling) FIA_USB.1 (User-subject binding) FMT_SAE.1 (Time-limited authorisation) FIA_UID.1 (Timing of identification) FIA_UAU.1 (Timing of authentication) FIA_UAU.5 (Multiple authentication mechanisms) FIA_UAU.6 (Re-authenticating)

Table 7.6: Mapping of *Security Function SF_UserIdentificationAuthentication* to *Security Functional Requirements (SFR)*

7.7 SF_AccessControl

The TOE validates external user data inputs associated with cryptographic operations (in accordance with FDP_ITC.2/UD, FDP_ITC.2/TS), e.g., to check whether the keyID is known to the TOE and the requested operation is possible with the key. In case of failed validation, the TOE will notify the requesting entity of the existing issue(s).

The TOE is implemented in a way to export data with security attributes (in accordance with FDP_ETC.2, FDP_ETC.2/TS). For example, enciphered user data will be exported with a reference to the encryption/decryption key and MAC.

The TOE checks whether the requesting entity is authorized to perform data exports (in accordance with FDP_ETC.1, FDP_ACC.1/Oper, FDP_ACF.1/Oper, FDP_ACF.1/TS). For example, only the Crypto-Officer and the Key Owner of a specific key are authorized to perform cryptographic operations in general, and only an Application Component is allowed to timestamp data. The TOE enforces this by checking whether the required role is in the set of roles in an active session. Access is denied if the user or entity is unauthorized.

After successful authentication, a user assumes one (or more) of the following roles:

- Key Owner,
- Application Component,
- Crypto-Officer,
- User Administrator,
- Update Agent,
- Auditor,
- Timekeeper.

A user may have the ability to assume multiple roles. The role determines the actions the user is allowed to execute. The Auditor role is mutually exclusive with other administrative roles. The role determines the actions the user is allowed to execute.

The role is requested as part of the authentication process. If a user or entity wants to claim a role that is not in the set of assigned roles the authentication fails.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.7.

Security Function Group	Mapped Security Functional Requirements
SF_AccessControl	FDP_ITC.2/UD (Import of user data with security attributes - User data) FDP_ITC.2/TS (Import of user data with security attributes - User data for time stamping) FDP_ETC.2 (Export of user data with security attributes) FDP_ETC.2/TS (Export of user data with security attributes - User data with time stamp) FDP_ETC.1 (Export of user data without security attributes) FDP_ACC.1/Oper (Subset access control - Cryptographic operation) FDP_ACF.1/Oper (Security attribute based access control - Cryptographic operations) FDP_ACF.1/TS Security attribute based access control - Cryptographic operations

Table 7.7: Mapping of *Security Function SF_AccessControl* to *Security Functional Requirements (SFR)*

7.8 SF_SecurityManagement

The TOE supports secure management of

- security functions,
- security roles,
- security attributes, and
- security functions behavior

as follows.

The TOE maintains the roles Key Owner, Application component, Crypto-Officer, User Administrator, Update Agent, Auditor, Timekeeper in association with users and entities in a persistently stored database. The roles Unidentified User, Unauthenticated User are assigned to users or entities during the authentication process and kept in volatile memory in accordance with FMT_SMR.1 and FMT_SMR.1/TSA.

The TOE ensures that only secure values are accepted for security attributes related with cryptographic keys by enforcing mandatory inputs and performing input validation for key generation requests in accordance with FMT_MSA.2. The TOE checks whether a keyID already exists before accepting a new keyID. The TOE provides a key generation interface that enforces that a key is either a secret key, private key, or public key. The input validation enforces that there is a least one Key usage type specified.

The TOE enforces FMT_MSA.3/KM by enforcing mandatory inputs and performing input validation. That is, the values are provided with each request, and the restrictiveness is given by validating the existence and contents of all associated security attributes in the provided input.

The TOE ensures that only authenticated users with the necessary role are allowed to execute corresponding functions and prevents unauthorized access by keeping a context for active sessions and comparing the required role of a function against the actual role of the authenticated user:

- Only User Administrators of the TOE may disable and enable password authentication for users in accordance with FMT_MOF.1.1 (1) and FMT_MOF.1.1 (2). The TOE persistently stores whether password authentication is enabled or disables on a per user basis.
- Only User Administrators of the TOE may set up PACE credentials including the ID of the entity and PACE PIN for trusted entities in accordance with FMT_MOF.1.1 (3) and FMT_MOF.1.1 (4). The TOE persistently stores these credentials.
- Only User Administrators of the TOE may unblock users that have been blocked after too many unsuccessful authentication attempts (cf. FMT_MOF.1.1 (5)).
- Only the Timekeeper may change the time of the TOE in accordance with FMT_MOF.1.1/TSA (1) and FMT_MOF.1.1/TSA (2).
- Only the Auditor may select auditable events (cf. FMT_MOF.1.1/TSA (3)).
- Only the Auditor may define the keyID used for signing audit logs according to FMT_MOF.1.1/TSA (5).
- Only the User Administrator may create, delete, modify authentication reference data and permanent session key for users and entities in accordance with FMT_MTD.1/RAD (1), FMT_MTD.1/RAD (2), FMT_MTD.1/RAD (3), FMT_MTD.1/RAD (4).
- Only the User Administrator may define time limits for roles in accordance with FMT_MTD.1/RAD (5). A session is terminated once one of its roles expires (conceptually this is a reset to role Unidentified User in accordance with FMT_MTD.1/RAD (6)). The TOE persistently stores timestamps (cf. SF_Timestamping in Section 7.11) to realize this behavior.
- Only the Crypto-Officer may initially set the keyID, the Key owner of a key, the Key type, the Key usage type, Key access control attributes (i.e., whether a key is clusterable or exportable), the Key validity time period in accordance with FMT_MSA.1/KM. The TOE validates the inputs and stores them persistently thereafter. The TOE does not provide any interface for the modification or deletion of the keyID, Key owner, Key type, Key usage type, Key validity time period, Key usage counter of en existing key in accordance with FMT_MSA.1/KM (2) and FMT_MSA.1/KM (3).
- Only the Crypto-Officer may change the Key access control attributes of existing key (i.e., whether a key is clusterable or exportable) in accordance with FMT_MSA.1/KM (4).

- Only the Crypto-Officer of the Key Owner may retrieve the Key type, Key usage type, Key access control attributes, Key validity time period and Key usage counter of a key in accordance with FMT_MSA.1/KM (5). While the Crypto-Officer may do this for any key known to the TOE, a user or entity not in the Crypto-Officer must be the assigned Key Owner of the specific queried key to be allowed to see these details.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.8.

Security Function Group	Mapped Security Functional Requirements
SF_SecurityManagement	FMT_SMF.1 (Specification of Management Functions) FMT_SMR.1 (Security roles) FMT_MSA.2 (Secure security attributes) FMT_MOF.1 (Management of security functions behaviour) FMT_MTD.1/RAD (Management of TSF data - Authentication reference data and Authentication Data Records) FMT_MSA.1/KM (Management of security attributes - Key security attributes) FMT_MSA.3/KM (Static attribute initialisation - Key management)

Table 7.8: Mapping of *Security Function SF_SecurityManagement* to *Security Functional Requirements (SFR)*

7.9 SF_TSFProtection

The TOE performs self-tests (cf. FPT_TST.1) to demonstrate the correct operation of the TSF and preserves its secure state in case of failure. The correct functioning of the secure hardware platform is part of this testing.

The self-test procedure is the first action that is executed by the TOE after it is started by the underlying *Operating System (OS)*. Successfully passing this self-test is an absolute requirement before the TOE will switch to an operational state and accept user communication.

In accordance with FPT_FLS.1, the TOE preserves a secure state in order to protect the user data in case of failures. To that end, if any part of the self-test returns an error result, this can lead to one of two consequences, depending on the gravity of the problem detected:

- The CSPL will pause until the problem is fixed or reboot itself
- The CSPL takes measures to make all key material stored by it inaccessible by erasing all necessary information to decrypt its storage

The first option will be chosen on a minor problem (e.g. a network timeout when communicating with an online timesource or an error during the gathering of the entropy for the random number generator). The latter one will be chosen on any severe incident being detected which might be part of an attack (e.g. a detected tampering of the hardware casing).

The TOE and its operational environment provide a tamper detection. If a tampering is detected, the TOE transitions to a *Secure Error State*. Changing to that state will make any key material permanently inaccessible by zeroising memory and deleting necessary decryption keys from the storage. The TOE will not be usable anymore afterwards.

Right after the boot of the TOE, a self test happens. This self-test does at least execute the following steps (FPT_TST.1):

- Check the availability of the timesource
- Check the entropy of the RNG
- Check for tampering attempts

If the self-test finds a tampering attempt, a hard failure of the self-test is triggered which causes the TOE to be transferred into the *Secure Error State*.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.9.

Security Function Group	Mapped Security Functional Requirements
SF_TSFProtection	FPT_FLS.1 (Failure with preservation of secure state) FPT_TST.1 (TSF testing)

Table 7.9: Mapping of *Security Function SF_TSFProtection* to *Security Functional Requirements (SFR)*

7.10 SF_UpdateCodePackage

The TOE supports the verification and decryption of Update Code Packages (UCPs). Authentic decrypted UCPs may be used for version updates of the TOE.

Updates are not done incrementally or in-place but via complete installations on a partition. For that end, the hardware platform of the TOE provides two system partitions: Partition *A* and Partition *B*. At any time either Partition *A* or Partition *B* is marked as “active”. The “active” partition is used for booting the TOE.

A UCP is a complete package containing everything that is required to run the TOE, namely:

- Linux kernel,
- Signatures of the kernel to enable a secure boot process via UEFI
- Full filesystem of the Operating System,
- Configuration files,
- the TOE.

A UCP is sent to the TOE via a dedicated external function accessible to an authenticated user in the *Update Agent* role. The UCP has both its confidentiality and its authenticity guaranteed via the cryptographic mechanisms of encryption and a digital signature. The concrete mechanisms are schemes already supported by the TOE as discussed in sections 7.2 and 7.4.

The verification of the UCP follows the *Encrypt-Then-Authenticate* paradigm, meaning that only after the signature of the ciphertext has been successfully verified, a decryption will be attempted. An unsuccessful verification results in the immediate deletion of the UCP from the storage of the TOE.

After the verification and decryption of an authentic UCP, the TOE (currently running on e.g. Partition A) may install it to the "inactive" (i.e. the partition that is not marked as "active") system partition (e.g. Partition B). Then, the TOE may conduct the version upgrade by updating the expected version number. The upgrade is finalized by marking the other system partition (e.g. Partition B) as "active" and rebooting the system.

Version downgrades are prevented by:

(1) rejection of UCPs with smaller version number than the currently installed TOE version number

(2) preventing operation of a TOE sample on data of the TOE with a higher version number

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.10.

Security Function Group	Mapped Security Functional Requirements
SF_UpdateCodePackage	FDP_ITC.2/UCP (Import of user data with security attributes - Update Code Package) FPT_TDC.1/UCP (Inter-TSF basic TSF data consistency) FCS_COP.1/VDSUCP (Cryptographic operation - Verification of digital signature of the Issuer) FCS_COP.1/DecUCP (Cryptographic operation - Decryption of authentic Update Code Package) FDP_ACC.1/UCP (Subset access control - Update code Package) FDP_ACF.1/UCP (Security attribute based access control - Import Update Code Package) FDP_RIP.1/UCP (Subset residual information protection)

Table 7.10: Mapping of *Security Function SF_UpdateCodePackage* to *Security Functional Requirements (SFR)*

7.11 SF_Timestamping

The TOE automatically synchronizes its local time periodically with a trusted time service via an authenticated *Network Time Protocol (NTP)* service provider. No standard NTP servers, which do not provide any cryptographic authentication, will ever be used by the TOE.

In addition to this automatic time synchronization, the TOE provides an external function allowing any user authenticated in the *Timekeeper* role to manually set the time of the TOE. Even if the time has been set manually, the automatic time synchronization process which relies on a trusted time source will continue to operate as before.

The TOE is able to add a verifiable timestamp to audit data or any other data exported from the TOE (cf. SF_SecurityAudit in Section 7.13, SF_DataIntegrityMechanisms in Section 7.4), which makes it possible to cryptographically guarantee, that the piece of data was created before the signature time of the timestamp. This prevents the backdating of data. This service of the TOE is used for the generation of signed log messages in the context of TR-03151[12].

Timestamping is also used in the context of time-limited authorization in order to have reliable time measurements for session termination (cf. SF_UserIdentificationAuthentication in Section 7.6).

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.11.

Security Function Group	Mapped Security Functional Requirements
SF_Timestamping - Time Stamp	FDP_DAU.2/TS (Data Authentication with Identity of Guarantor - Signature with time stamp and optional key usage counter)
SF_Timestamping - Access control on time stamp service	FDP_ITC.2/TS (Import of user data with security attributes - User data for time stamping) FDP_ETC.2/TS (Export of user data with security attributes - User data with time stamp) FDP_ACF.1/TS (Security attribute based access control - Cryptographic operations)
SF_Timestamping - Security Management	FMT_SMF.1/TSA (Specification of Management Functions) FMT_SMR.1/TSA (Security roles) FMT_MOF.1/TSA (Management of security functions behaviour)

Table 7.11: Mapping of *Security Function SF_TimeStamping* to *Security Functional Requirements (SFR)*

7.12 SF_Clustering

The TOE supports clustering of TOE samples with a single master node and one or multiple slave nodes. In general, clustering support is an optional feature, but some applications might require it. For example, the Security Module Application for Electronic Record-Keeping Systems (SMAERS) [11] requires clustering.

The Master-CSPLight communicates updates concerning its cryptographic keys and authentication data records to Slave-CSPLight(s) of the same cluster via a secure channel. The TOE supports the generation of audit logs for data imports and exports. The TOE supports the generation of an audit log once a TOE sample becomes master of cryptographic keys and authentication data records. A TOE-external reverse proxy routes requests to the Master-CSPLight.

Upon initialisation, the TOE can be configured to run in cluster mode. If that is desired, the following values have to be set:

- IP addresses or hostnames of other hosts in the cluster
- Shared secret between all cluster nodes (for establishing a secure connection)
- Whether the entity is supposed to be a master or a slave node

Only the database of the master node will allow write operations, all other nodes will be in a read-only state. Changes to the database of the master node will be propagated to all other nodes in the cluster periodically via a secured connection.

The role of a node in the cluster (master or slave) is fixed upon configuration and can only be altered manually by a user in the role *User Administrator*. A slave node will never change into master mode automatically. The most common case for assigning the Master-CSPLight role to a CSPL in the cluster is either at initial setup or after hardware failure. Non-clusterable keys will not be transferred. The transfer of key material and authentication reference data results in the generation of audit records (cf. SF_SecurityAudit in Section 7.13). All keys shared within a cluster will be transferred with all related attributes (like the key usage counter).

If a node is permanently removed from a cluster by the *User Administrator*, the cluster-secret and all synchronized cryptographic keys and authentication data records are deleted from it.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.12.

Security Function Group	Mapped Security Functional Requirements
SF_Clustering	FDP_ACC.1/CL (Subset access control - Clustering) FMT_MTD.1/CL (Management of TSF data - Authentication Data Records and cryptographic keys) FCS_CKM.5/CLDH (Cryptographic key derivation - Cluster keys) FPT_TCT.1/CL (TSF data confidentiality transfer protection - Cluster) FPT_TIT.1/CL (TSF data integrity transfer protection - Cluster) FPT_ISA.1/CL (Import of TSF data with security attributes - Cluster) FPT_ESA.1/CL (Export of TSF data with security attributes - Cluster) FPT_TDC.1/CL (Inter-TSF basic TSF data consistency - Clustering)

Table 7.12: Mapping of *Security Function SF_Clustering* to *Security Functional Requirements (SFR)*

7.13 SF_SecurityAudit

The TOE generates audit records of auditable events in accordance with FAU_GEN.1 and FAU_GEN.1/CL.

The auditable events are:

- Start-up and shutdown of the audit functions
- Discrete adjustment of the real time clock (cf. SF_Timestamping in Section 7.11)
- Start-up after power-up
- Import of UCP according to FDP_ITC.2/UCP (cf. SF_UpdateCodePackage in Section 7.10)
- Authentication failure handling (cf. SF_UserIdentificationAuthentication in Section 7.6)
- Generation of signature key pairs (cf. SF_KeyManagement in Section 7.1)
- Cryptographic key destruction (cf. SF_KeyManagement in Section 7.1)
- custom audit events created by entities in administrative roles, (un)successful authentications, (un)blocking of users by the User Administrator
- Generation of cluster keys for the secure channel according to FMT_MTD.1/CL and FCS_CKM.5/CLDH (cf. SF_Clustering in Section 7.12)
- Export of Authentication Data Records and cryptographic keys from the MasterCSP-Light according to FPT_ESA.1.3/CL, Management of Authentication Data Records (FMT_MTD.1/RAD): creation and deletion of Authentication Data Record (cf. SF_Clustering in Section 7.12)
- Import according to FPT_ISA.1/CL of Authentication Data Records and cryptographic keys into Slave-CSPLights (cf. SF_Clustering in Section 7.12)

In accordance with FAU_GEN.1.2, the TOE records the date and time, type, identity of the affected user or entity (if applicable), the result of the event (success or failure), and more details regarding the event (if applicable). Date and time is based on the current timestamp of the TOE. The type is based on a predefined set of values (defined as `EventType` in Listing 7.1).

The audit trail can be exported from the TOE as audit log messages. Audit log messages are always digitally signed and accompanied by a timestamp (cf. SF_Timestamping in Section 7.11) making all manipulation attempts futile. Audit log messages are encoded using DER (*Distinguished encoding rules*) according to the ASN.1 specification in [12]. In addition to the audit log structure defined in [12], Listing 7.1 defines the structure of the `seAuditData` field.

Listing 7.1: ASN.1 specification of audit trail exports for the field `seAuditData` of audit logs

```
FiskalyCSPLAuditingV1 { iso(1) identified-organization(3) dod(6) internet(1) private(4)
    enterprise(1) fiskaly(56096) applications(3) cspl(1) }
  DEFINITIONS
  IMPLICIT TAGS ::=

BEGIN

  id-fiskaly          OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6)
      internet(1) private(4) enterprise(1) 56096 }
  id-audit-trail-type OBJECT IDENTIFIER ::= { id-fiskaly applications(3) cspl(1) 1 }

  AuditTrail ::= SEQUENCE {
    version [0] IMPLICIT Version        DEFAULT v1,
    type    [1] IMPLICIT AuditTrailType DEFAULT id-audit-trail-type,
    records [2] IMPLICIT AuditRecords
  }

  AuditTrailType ::= OBJECT IDENTIFIER (id-audit-trail-type)

  AuditRecords ::= SEQUENCE OF auditRecord AuditRecord

  AuditRecord ::= SEQUENCE {
    timestamp [0] IMPLICIT UnixTimestamp,
    type      [1] IMPLICIT EventType,
    status    [2] IMPLICIT Status,
    subject   [3] IMPLICIT Subject OPTIONAL,
    details   [4] IMPLICIT Details OPTIONAL
  }

  EventType ::= ENUMERATED {
    unspecified (0),
    auditStart (1),
    auditStop (2),
    systemStart (3),
    timeAdjustment (4),
    ucpImport (5),
    authBlocked (6),
    authUnblocked (7),
    keyGeneration (8),
    keyDestruction (9),
    clusterExport (10),
    clusterImport (11),
    clusterKeyGeneration (12),
    custom (13)
  }

  Version ::= INTEGER {
    v1 (1) -- currently only v1 is defined
  }

  UnixTimestamp ::= INTEGER -- 64 bit / 8 Byte "1541688722"

  Subject ::= UUIDv4

  UUIDv4 ::= OCTET STRING (SIZE (16))

  Status ::= ENUMERATED {
    unspecified (0),
    success (1),
    failure (2)
  }

  Details ::= SEQUENCE {
    type  [0] IMPLICIT DetailsType OPTIONAL,
    value [1] IMPLICIT DetailsValue
  }

  DetailsType ::= OBJECT IDENTIFIER

  DetailsValue ::= OCTET STRING -- ANY DEFINED BY DetailsType

  -- If Details->type == id-audit-record-details-type, then Details->value will be an
       instance of AuditRecordDetails.
  -- Note that the field names correspond to the EventType enumeration fields:
  -- for instance, if AuditRecord->type == timeAdjustAuto, then the corresponding
       timeAdjustAuto field of AuditRecordDetails will be defined.
  AuditRecordDetails ::= CHOICE {
    timeAdjustment [1] IMPLICIT AuditRecordDetailsTimeAdjustment,
    ucpImport      [2] IMPLICIT AuditRecordDetailsUCPImport
  }

  AuditRecordDetailsTimeAdjustment ::= SEQUENCE {
    previousTime [0] IMPLICIT UnixTimestamp,
    currentTime  [1] IMPLICIT UnixTimestamp
  }

  AuditRecordDetailsUCPImport ::= SEQUENCE {
    previousVersion [0] IMPLICIT UCPVersion,
    currentVersion  [1] IMPLICIT UCPVersion
  }
```

```
  UCPVersion ::= PrintableString -- e.g. 1.0.1

END
```

The auditor can manually export audit records to free storage space (cf. FMT_MTD.1/Audit). In case the audit log reached a threshold of its overall capacity as configured by the Auditor, the TOE will enter a special error mode. In this error mode, all operations that cause audit events to be written will not longer be available but instead return an error message. An exception are those functions are required for the Auditor to authenticate and manually export and clear the log.

The detailed mapping of this *Security Function* to its corresponding SFRs can be found in Table 7.13.

Security Function Group	Mapped Security Functional Requirements
SF_SecurityAudit - Clustering	FAU_GEN.1/CL Audit data generation
SF_SecurityAudit - Timestamping	FAU_GEN.1 (Audit data generation) FAU_GEN.1/CL (Audit data generation) FMT_MTD.1/Audit (Management of TSF data) FAU_STG.1 (Protected audit trail storage) FAU_STG.3 (Action in Case of Possible Audit Data Loss) FPT_STM.1 (Reliable time stamps) FPT_TIT.1/Audit (TSF data integrity transfer protection - Audit functionality)

Table 7.13: Mapping of *Security Function SF_SecurityAudit* to *Security Functional Requirements (SFR)*

Bibliography

[1] ANSI-X9.63. Key Agreement and Key Transport Using Elliptic Curve Cryptography, 2011.

[2] BSI. A proposal for: Functionality classes for random number generators, Version 2.0.

[3] BSI. BSI, Elliptic Curve Cryptography, BSI Technical Guideline TR-03111, Version 2.1, 1.6.2018.

[4] BSI. Technical Guideline TR-03110 Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token – Part 2 - Protocols for electronic IDentification, Authentication and trust Services (eIDAS), Version 2.21, 2016 .

[5] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1, Revision 5, April 2017. https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf.

[6] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1, Revision 5, April 2017. https://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf.

[7] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, CCMB-2017-04-003, Version 3.1, Revision 5, April 2017. https://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf.

[8] Federal Office for Information Security. Protection Profile Cryptographic Service Provider Light - Time Stamp Service and Audit (PPC-CSPLight-TS-Au), Version 1.0. https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/aktuell/PP_0112.html.

[9] Federal Office for Information Security. Protection Profile Cryptographic Service Provider Light (CSPL) Version 1.0, BSI-CC-PP-0111-2019. https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/aktuell/PP_0111.html.

[10] Federal Office for Information Security. Protection Profile Cryptographic Service Provider Light Time Stamp Service and Audit - Clustering (PPC-CSPLight-TS-Au-Cl) Version 1.0, BSI-CC-PP-0113-2019. https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/aktuell/PP_0113.html.

[11] Federal Office for Information Security. Security Module Application for Electronic Recordkeeping Systems (SMAERS) Version 1.0, BSI-CC-PP-0105-V2-2020. https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/aktuell/PP_0105_0105_V2.html.

[12] Federal Office for Information Security. Technical Guideline BSI TR-03151 Secure Element API (SE API) Version 1.0.1. https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03151/index_htm.html.

[13] Federal Office for Information Security. Technical Guideline TR-02102-2Cryptographic Mechanisms: Recommendations and Key Lengths, Version 2020-01. `https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-2.pdf`.

[14] FIDO Alliance. Alliance Proposed Standard FIDO ECDAA Algorithm, 11 April 2017. `https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-ecdaa-algorithm-v1.2-ps20170411.html`.

[15] fiskaly GmbH. Preparative Procedures & Operational User Guidance Documentation fiskaly Cloud Crypto Service Provider Version 1.2.4, 2021.

[16] ICAO: Machine Readable Travel Documents. ICAO Doc9303, Part 11: Security Mechanisms for MRTDSs, seventh edition, 2015 .

[17] ISO/IEC 10116 Information Technology - Security techniques. Modes of operation for an n-bit block cipher, 2017.

[18] ISO/IEC 14888-2 Information technology – Security techniques. Digital signatures with appendix – Part 2: Integer factorization based mechanisms, 2008.

[19] ISO/IEC 18033-3 Information technology - Security techniques. Encryption algorithms - Part 3: Block ciphers, 2010.

[20] ISO/IEC 9797-2 Information Technology - Security techniques. Message Authentication Codes (MACs), Part 2: Mechanisms using a dedicated hash-function, 2011.

[21] NIST.] Recommendation for Random Number Generation Using Deterministic Random Bit Generators, NIST 800-90A Revision 1, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-90a.pdf.

[22] NIST. Digital Signature Standard (DSS), 2013.

[23] NIST. Federal Information Processing Standards Publication 197, Advanced Encryption Standard (AES), 2001.

[24] NIST. Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, May 2004 .

[25] NIST. Recommendation for Key Derivation through Extraction-then-Expansion, Special Publication SP800-56C, November 2011.

[26] NIST. Secure Hash, Standard (SHS), 2012.

[27] NIST. SP800-38A Recommendation for Block Cipher Modes of Operation: Methods and Techniques.

[28] NIST. SP800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005 .

[29] NIST. SP800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.

[30] NIST. SP800-38F Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, 2012.

[31] RFC2104. HMAC: Keyed-Hashing for Message Authentication.

[32] RFC5639. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, http://www.ietf.org/rfc/rfc5639.txt, 2010.

[33] RFC5903. Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2.

[34] RFC6954. Using the Elliptic Curve Cryptography (ECC) Brainpool Curves for the Internet Key Exchange Protocol Version 2 (IKEv2).

[35] Trusted Platform Module Library. Part 1: Architecture, Family "2.0", Level 00, Revision 01.38, September 29, 2016.

[36] PKCS #1 v2.2: RSA Cryptographic Standard. https://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11300-wp-pkcs-1v2-2-rsacryptography-standard.pdf, 27.10.2012.

Chapter 8

Keywords and Abbreviations

Term	**Description**
authentication reference data	data used by the TOE to verify the authentication attempt of a user
authentication verification data	data used by the user to authenticate themselves to the TOE
authenticity	the property that ensures that the identity of a subject or resource is the one claimed (cf. ISO/IEC 7498-2:1989)
cluster	a system of TOE samples initialized by an administrator and communication through trusted channels in order to manage known users and to share the cryptographic keys
cryptographic key	a variable parameter which is used in a cryptographic algorithm or protocol
data integrity	the property that data has not been altered or destroyed in an unauthorized manner (cf. ISO/IEC 7498-2:1989)
firmware	executable code that is stored in hardware and cannot be dynamically written or modified during execution while operating on a non-modifiable or limited execution platform, cf. ISO/IEC 19790
hardware	physical equipment or comprises the physical components used to process programs and data or to protect physically the processing components, cf. ISO/IEC 19790
Issuer of update code package	Trusted authority issuing an update code package (UCP) and holding the signature private key for signing the UCP and corresponding to the public key implemented in the TOE for verification of the UCP. The issuer is typically the TOE manufacturer. The issuer of an UCP is identified by the security attribute Issuer of the UCP.

Platform guidance	All documentation provided by the hardware manufacturer, or software platform manufacturer, that provides information on how to securely implement functionality.
private key	confidential key used for asymmetric cryptographic mechanisms like decryption of cipher text, signature-creation or authentication proof, where it is difficult for the adversary to derive the confidential private key from the known public key
public key	public known used for asymmetric cryptographic mechanisms like encryption of cipher text, signature-verification or authentication verification, where it is difficult for the adversary to derive the confidential private key from the known public key
secret key	key of symmetric cryptographic mechanisms, using two identical keys with the same secret value or two different values, where one may be easy calculated from the other one, for complementary operations like encryption / decryption, signature-creation / signature-verification, or authentication proof / authentication verification
secure channel	a trusted channel which is physically protected and logical separated communication channel between the TOE and the user, or is protected by means of cryptographic mechanisms
software	executable code that is stored on erasable media which can be dynamically written and modified during execution while operating on a modifiable execution platform, cf. ISO/IEC 19790
trusted channel	a means by which a TSF and another trusted IT product can communicate with necessary confidence (cf. CC part 1 [5], paragraph 97)
update code package	code if implemented changing the TOE implementation at the end of the TOE life time

Table 8.1: Glossary (Table 8 in Base-PP [9])

Keywords and Abbreviations

Acronym	**Term**
A.xxx	Assumption
CC	Common Criteria
CSP	Cryptographic Service Provider
CSPLight	Cryptographic Service Provider Light
ECC	Elliptic curve cryptography
HMAC	Keyed-Hash Message Authentication Code
KDF	Key derivation function
MAC	Message Authentication Code
n. a.	Not applicable
O.xxx	Security objective for the TOE
OE.xxx	Security objective for the TOE environment
OSP.xxx	Organisational security policy
PACE	Password Authenticated Connection Establishment
PKI	Public key infrastructure
PP	Protection profile
SAR	Security assurance requirements
SFR	Security functional requirement
T.xxx	Threat
TOE	Target of Evaluation
TSF	TOE security functionality
UCP	update code package

Table 8.2: Abbreviations (Table 9 in Base-PP [9])

Code and Document Signing

For verifying signed artifacts, the following public key shall be used:

```
RWRUupxsFLBrtFlb7g2ZWVFSQE23BvMEtPszqNu2E8Q32U4L99AKexpl
```

fiskaly GmbH uses an `Ed25519` key following OpenBSD's `signify` [1] approach for digitally signing all published artifacts (i.e., software and documents). In particular, fiskaly GmbH uses the `minisign` [2] tool that is fully compatible with the `signify` verification tool. As an example, the following command can be used for verification:

```
$ minisign -P {PUBKEY} -V -m {FILENAME}
Signature and comment signature verified
Trusted comment: timestamp:1603971010 file:{FILENAME}
```

[1] https://www.openbsd.org/papers/bsdcan-signify.html

[2] https://jedisct1.github.io/minisign/