swissbit®

Swissbit Cloud SMAERS - Common Criteria Security Target

Version 1.0.11 (96e2b1615d60f6e0f2ef4a1b4f3609af0357a97b), 2024-11-13

Table of Contents

1. ST Introduction

In order to combat tax-fraud, electronic record-keeping systems in Germany must be equipped with a 'Certified Technical Security System' (CTSS; 'Zertifizierte Technische Sicherheitseinrichtung') that consists of a storage medium, a security module, and a unified digital interface. The security module is subject to common criteria security certifications. W.r.t. to security requirements for the security module – defined by Bundesamt für Sicherheit in der Informationstechnik – the module consists of two components:

1. an application component that handles the business logic and functionality required to serve an electronic record-keeping system. This component is dubbed the *security module application for electronic record-keeping systems (SMAERS)*.
2. a generic and reusable cryptographic component that implements the core cryptographic functionality required. This component is dubbed *cryptographic service provider (CSP)*.

This Security Target defines a TOE for the SMAERS component based on **[PP-SMAERS]**. Depending on the configuration of the TOE architecture, different security requirements exist for a CSP. These are defined in two Protection Profiles and Protection Profile Configurations. For details on allowed architectures and required Protection Profiles and configurations, cf. Chapter 1.2, in particular *Section Non-TOE Hardware/ Software/ Firmware available to the TOE* of **[PP-SMAERS]** or the sections **Non-TOE Hardware/Software/Firmware available to the TOE** and **Architecture** within this document.

In the following, the abbreviation CSP-L is redundantly used for all allowed configurations mentioned.

1.1. ST Reference

This Security Target has the following reference parameters:

- ST Reference: Swissbit Cloud SMAERS - Common Criteria Security Target
- Sponsor: Swissbit AG
- ST Version: 1.0.11
- build from git commit: 96e2b1615d60f6e0f2ef4a1b4f3609af0357a97b
- ST Date: 2024-11-13
- CC Version: 3.1 Revision 5
- Assurance Level: EAL 2 augmented by ALC_LCD.1 and ALC_CMS.3
- Certification ID: BSI-DSZ-CC-1239

1.2. TOE Reference

Table 1. TOE Reference

TOE Identifier	TOE Version
Swissbit Cloud SMAERS	1.0.5

The TOE is delivered with the following additional documents:

Table 2. Delivery Items

Item	Version
Swissbit Cloud SMAERS - Guidance Manual **[SMAERS-AGD]**	1.0.5

1.3. TOE Overview

1.3.1. TOE Type

The Target of Evaluation (TOE) is named *Swissbit Cloud SMAERS* and is the SMAERS application of a Swissbit Cloud-TSE 2. As described in **[PP-SMAERS]**, the TOE is a software TOE. The TOE is going to be executed under the control of Swissbit such that the user of the TSE, i.e. the tax payer has no control over the TSE's execution platform. By this, also the Swissbit Cloud SMAERS is protected and separated from the tax payer.

The TOE is realized in client-server-architecture as defined in **[PP-SMAERS]**, so the TOE is communicating with a CSP-L via a trusted channel (referred to as client-server architecture). The CSP-L is always remotely connected, cf. **[PP-CSP][PP-CSPLight]**.

Realizing the TOE in client-server-architecture, this Security Target additionally uses the package *Trusted Channel between the TOE and the CSP* in chapter 7.

1.3.2. TOE Definition

The Swissbit Cloud SMAERS (or TOE) is a security module application (SMAERS) as part of the security module of a certified technical security system (CTSS) for electronic record-keeping systems (ERS) as described in **[PP-SMAERS]**. The TOE comprises multiple SMAES units and each unit is meant to be part of the Swissbit Cloud-TSE 2 of Swissbit AG. The TOE is a software TOE, coming as a java library to be directly integrated into the Swissbit Cloud-TSE 2. Here the CTSS interface component accesses the TOE. This is shown in **Figure 1**.

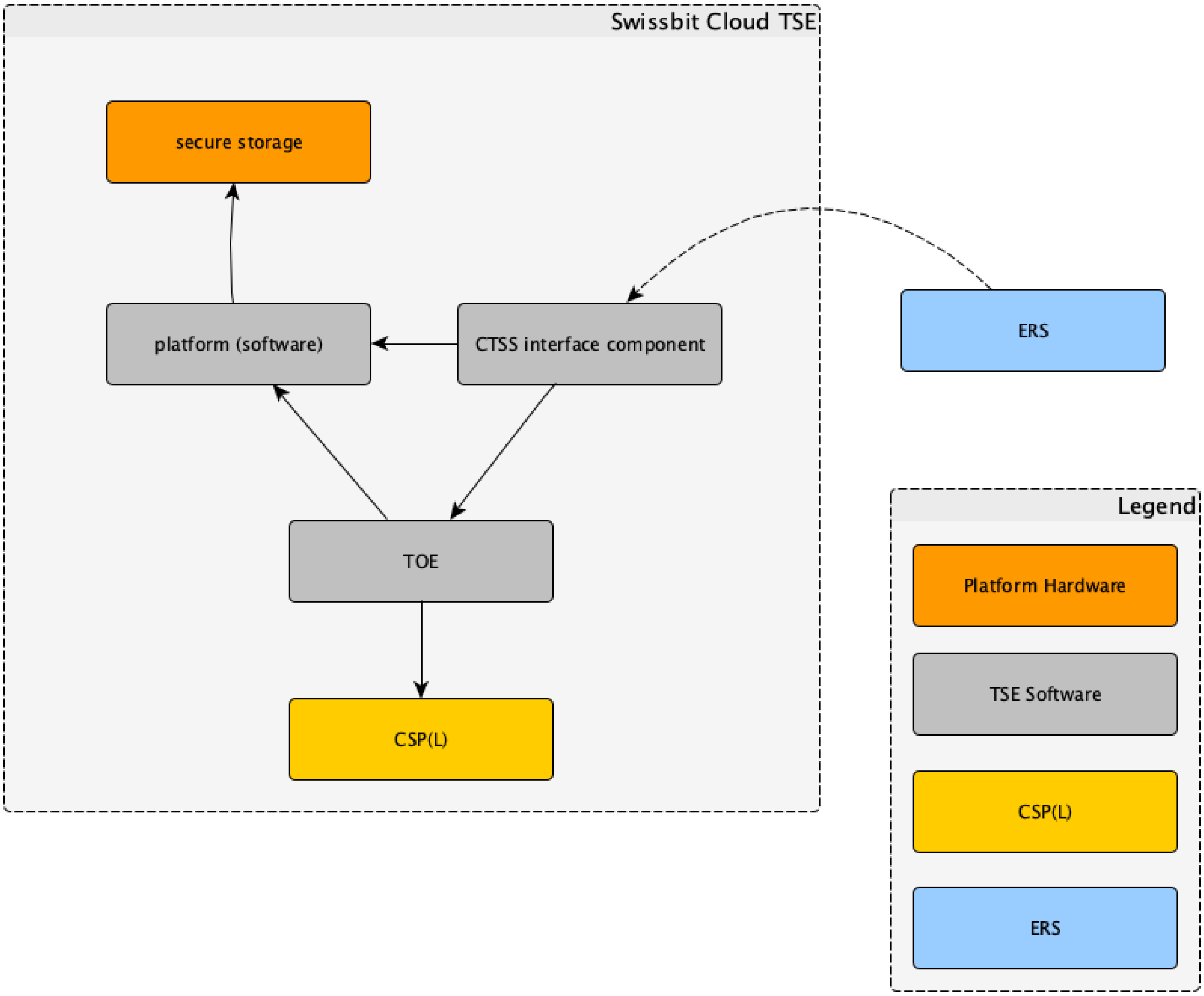

Figure 1. Overview of the components of Swissbit Cloud-TSE 2 and the TOE within

The Swissbit Cloud-TSE 2 containing the TOE meets the BSI Technical Guidance [BSI-TR-03153] and uses cryptographic services of the CSP-L compliant with BSI TR-03116-5 [BSI-TR-03116].

1.3.3. Method of Use

The TOE as part of its TSE protects accounts and records of one or more ERS. The TOE contains multiple SMAERS units as defined in [BSI-TR-03153], each of which has its own private signature key and certificate. Each SMAERS unit uses one signature key exclusively. So there is a 1:1 relation between SMAERS units and signature keys. Each of the signatures keys is stored in one CSP-L, so the SMAERS unit does not directly use it, but utilizes the CSP-L to perform the corresponding cryptographic operation. The timestamps in the resulting transaction logs are added by the same CSP-L or SMAERS, depending on the specification, cf [FCG], Section 146b.

As part of a cloud TSE, the TOE shall be configured so that it will only accept connections from ERS in the same cloud environment. More details can be found in **[SMAERS-AGD]**.

Note that each SMAERS unit establishes and maintains a private secure channel to the CSP-L, which holds the signature key, being used by the unit.

1.3.4. TOE Life Cycle

The TOE life cycle is part of the life cycle of the TSE. It consists of the following major steps:

- After development and certification, the TOE is integrated into Swissbit Cloud-TSE 2.
- When the development of the TSE is completed and all components acquired all required certifications, the TSE is delivered to Swissbit, the operator of the TSE.
- The TSE operator installs and setups the TSE (and the TOE within) according to the guidance and prepares it for operation. This might either be a new TSE, or an update of an existing TSE. An existing TSE might already contain active SMAERS units.
- Now, for each TSE being purchased by a Tax payer, the TSE operator creates one SMAERS unit within the TOE and gives access to this SMAERS unit to the tax payer
- The tax payer uses the TSE and by this the TOE within.
- The tax payer terminates the SMAERS unit, but this does not meet the end of the life cycle of the TOE.
- To end the TOE's life cycle, the TSE operator either installs a certified update of the TSE, or the TSE if finally terminated, which implicitly terminates all SMAERS units within.

The CSP-L is set up by the CSP-L operator independently.

The TOE life cycle ends either by installation of an *Update Code Package* upgrading the TOE to a future version or by terminating the TOE via its termination process.

Note that due to one TOE containing multiple SMAERS units, the life cycle of the TOE is not in 1:1 correspondence of the life cycle of a signature key in the CSP-L.

In case of a certified software update of the TSE, the platform installs the TSE and each unit detects the update on its own, when the unit becomes active for the first time after the installation. Here each unit acts on its own, updating its version number, creating the required messages and performing the required checks.

1.3.5. Non-TOE Hardware/Software/Firmware available to the TOE

The TOE is part of a TSE and requires a platform to be executed. The *Swissbit* as TSE operator sets up the platform, which is hosted in a cloud by a cloud service provider. In addition, the TOE requires a CSP-L, which has to be hosted according to the requirements of the CSP-L. The CSP-L shall export audit records in form of system logs meeting **[BSI-TR-03151]**. Last but not least, the TOE has to be integrated into Swissbit Cloud-TSE 2 to build a complete TSE. To operate the TOE according to the certification, the operator has to fullfil all requirements of the guidance documents, including especially the Umgebungschutzkonzept **[Umgebungsschutz]** of the Swissbit Cloud-TSE 2.

1.3.6. Usage and major security features

The Swissbit Cloud SMAERS as a java library offers a java-interface to its environment. This environment is the CTSS interface component of the Swissbit Cloud-TSE 2. It utilizes the TOE's functionality and offers a REST-Interface and a java-interface, wrapping the REST-Interface.

To use the TOE an ERS has to use the CTSS interface component via one of these interfaces.

The major TOE security feature is the generation of time stamped and signed *log messages*. To do so, the ERS triggers the TSE's CTSS Interface component accordingly. Then this component forwards the request to the TOE, which itself connects a remote CSP-L via a secure channel to trigger signature creation in the CSP-L. This also includes the creation and addition of the timestamp. Then the signed log message is returned to the TOE, which exports it to the CTSS Interface component, which then stores the log message in the Secure Storage of the TSE (cf. **[FCG]** section146b). Besides the generation of *log messages*, which originate from transactions of a ERS, the TOE also stores (and creates) *audit logs* and *system logs*. These messages resemble events in the TOE (or in the CSP-L), for example the execution of selftests, logins of users, etc.pp.

In addition, the TOE provides security management of the TSF for administrators. To do so, the TOE maintains a role *administrator* with PIN and PUK reference data for authentication. *Administrator* starts and stops the normal operation of the TOE for import of transaction data, generation and export of *log messages* and communication with the CSP. In addition, *administrator* prepares the configuration of the communication channels between the TOE and the CSP-L during production. For this communication, the TOE uses a trusted channel providing confidentiality and authenticity protection. To establish the trusted channel, the TOE implements the PACE protocol.

Note that the time of the Swissbit Cloud-TSE 2 is managed by the CSP-L and not via the SMAERS component, so the TOE does not manage or know a role for this purpose.

The TOE supports detection of *update code packages (UCP)*, which were installed by the TOE platform.

Last but not least, the TOE incorporates a selftest, which checks integrity of the TOE data and TOE implementation as well as identification of the external entities CSP-L and ERS.

1.4. TOE Description

The TOE is a security module application as part of the security module of a certified technical security system (CTSS or TSE) for electronic record-keeping systems (ERS). Figure 1 describes the interaction between TOE and non-TOE components. The CTSS consists of a security module, a storage medium, and a CTSS interface component providing the standardized digital interface (cf. [FCG], section 146a, paragraph 1, sentence 3) for the electronic record- keeping system and cash inspection (cf. [FCG], section 146b). The [KSV] section 2 requires the security module to provide

- the point in time when the transaction starts (cf. [KSV] section 2 sentence 2 number 1),
- the transaction number (cf. [KSV] section 2 sentence 2 number 2),
- the point in time when the transaction is completed or terminated (cf. SV section 2 sentence 2 number 6), and
- the check value (cf. [KSV] section 2 sentence 2 number 7).

The security module provides the logging of transactions and other audit-relevant processes in the form of log messages (cf. [BSI-TR-03153], Chapter 3.1). Log messages are created by the TOE using the CSP. Log messages consist of either certified data or audit data [BSI-TR-03151], as well as protocol data and a signature. There are three types of log messages, i.e. transaction logs, system logs and audit logs, cf. [TR SE] and Appendix: Log Message Structure and Data Dependency.

Transaction logs are created to protect the transaction data of the electronic record-keeping system as certified data. They are generated whenever a transaction is started, finished (i.e. completed or terminated), and may be generated when transaction data are updated. The protocol data of transaction logs contain the transaction number of the transaction and time stamps. All transaction logs with the same transaction number build together the required data of the fiscal transaction according to [KSV] Section 2, Sentence 2. System logs are generated to log the execution of system operations as described in [BSI-TR-03151] and TSF security events.

Audit logs are generated to document management or configuration operations of the CSP. The audit data of audit logs provide information for the interpretation of the transaction logs, e.g. providing information about setting or readjusting the time source that is used for time stamps. The TOE

- imports transaction data from the CTSS interface component and includes it as certified data in a transaction log,
- generates part of the protocol data for the transaction log including

- the transaction number generated by the TSF,
- the serial number included by the TSF for verification of the digital signature (keyID), + includes the timestamp, signature counter and digital signature created by the CSP over the certified data and the protocol data in the transaction log and system log,
- imports audit records from the CSP (cf. FAU_GEN.1) and exports them as audit log,
- generates a system log consisting of commands and TSF security events as certified data,
- exports all types of log messages to the CTSS interface component,
- provides identification and authentication of users, access control and security management of the TSF for authorized users by using cryptographic services of the CSP.

The signature counter enumerating the signatures created for log messages and the time stamps when the signature was created are generated by the CSP and are part of the protocol data. The main part of the protection profile in hand assumes the TOE being implemented as software running on a component that is physically separated from the CSP in a client-server architecture, cf. **[PP-CSP][PP-CSPLight]**). That means the security target shall claim the package trusted channel between the TOE and the CSP in Chapter 7. A trusted channel is necessary because the TOE and the CSP are implemented as separated components and must interact through a trusted channel in order to protect the integrity of the communication data, and to prevent misuse of the CSP w.r.t. signing and time stamping services provided for the TOE.

In case of the platform architecture, the TOE is running on a CSP where the CSP serves as a secure execution platform, cf. platform architecture **[PP-CSP]**. Then, the package trusted channel is not required. Note that the TOE must not be operated in the platform architecture in combination with CSPLight. The TOE must be compliant to BSI Technical Guideline TR-03153 **[BSI-TR-03153]**, and must use cryptographic services of the CSP compliant with BSI Technical Guideline TR-03116-5 **[BSI-TR-03116]**.

1.4.1. Method of Use

The TOE is part of the security module of the CTSS protecting accounts and records of one or more electronic record-keeping systems. If more than one electronic record-keeping system uses the TOE the serial number of ERS (clientID) sending input must be identifiable and known to the TOE for selecting the signature-creation key. The TOE generates time stamped and signed log messages using the CSP's cryptographic services in order to generate verifiable sequences of transaction data and log messages for cash register inspection, cf. **[FCG]**, Section 146b. The TOE provides security management features of the TSF for administrators. The security management features are used to configure the communication channels between the TOE with the CTSS interface component and the CSP. The TOE may support the security management functionality of the CSP by providing a communication interface to an administrator or other services, e.g. to a time server. The TOE requires the platform to support receiving and verifying the integrity of update code packages (UCPs) for installation of a new certified TOE.

1.4.2. Architecture

The TOE is a software TOE. It is developed in java, so naturally, its interface is a java one. It comes in form of jar file, which becomes part of the Swissbit Cloud-TSE 2. In Swissbit Cloud-TSE 2 the CTSS Interface component communicates with the TOE and invokes it, when the tax payer invokes a function of the TSE.

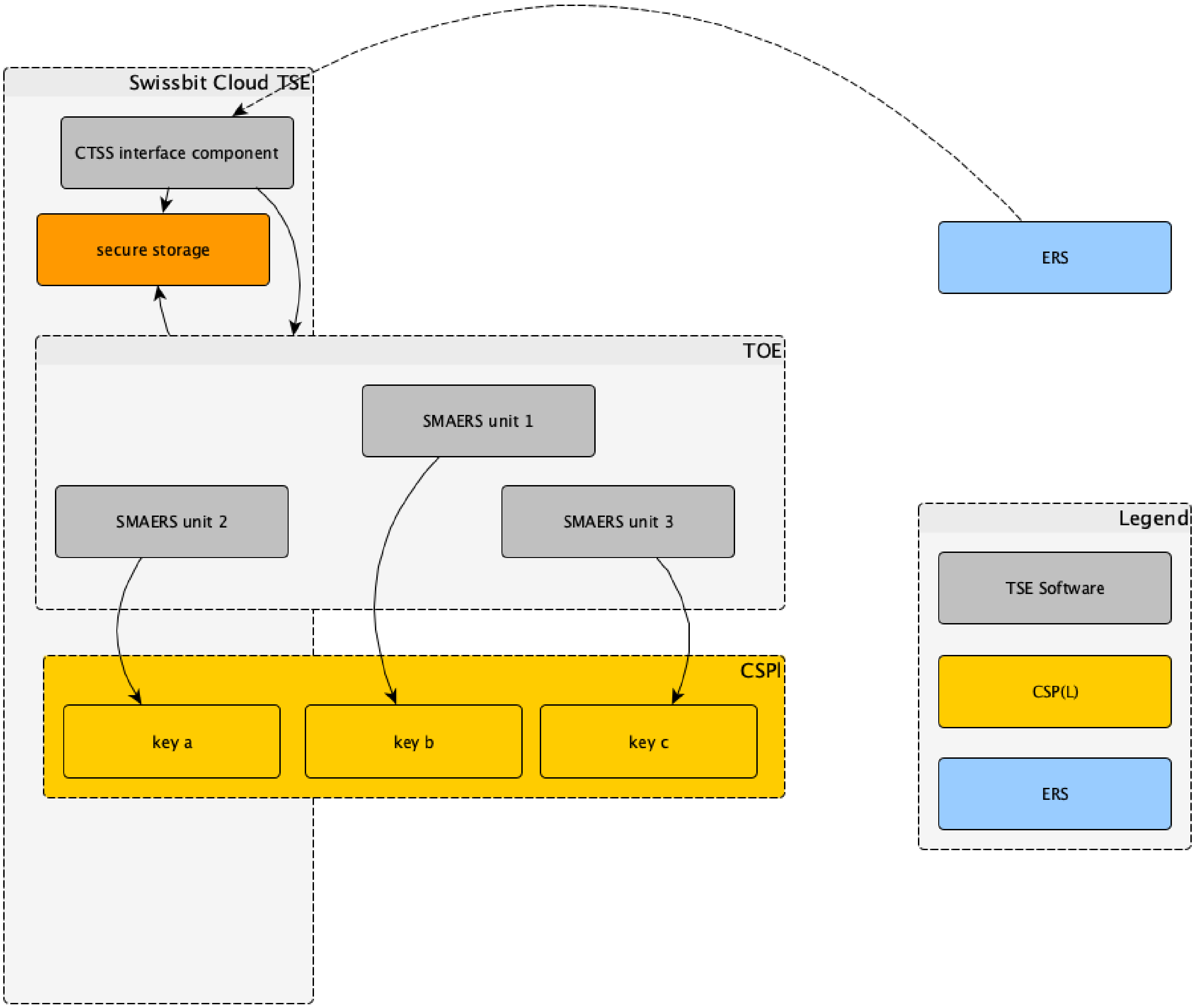

Figure 2. Overview of the TSE, the TOE, its SMAERS units and CSP-L relations.

Figure 2 shows the architecture of the Swissbit Cloud-TSE 2 with the Swissbit Cloud SMAERS within.

Due to the fact, the TOE hosts multiple SMAERS units, the TOE instantiates worker-threads in its process. When the tax payer invokes a function of the TSE and the TOE comes into play, one unused worker-thread from this thread-pool impersonates the SMAERS unit, loads the SMAERS units data and performs the function. After this is done, the worker-thread returns to the thread pool (after the SMAERS private data were cleared).

Each of these threads resemble one SMEARS unit at a time and each SMEARS unit is impersonated by at most one worker thread at a point in time. This allows to efficiently use the resources of the platform and instantiate SMAERS units when needed. Still, all the SMAERS units, share the same code. This is shown in Figure Figure 3.

Please note the the CSP-L and the CTSS Interface, as well as the secure storage are not part of the TOE, even though they are part of the TSE. Also, that all keys of all SMAERS units have to be present on the same CSP-L. Within the TOE, each SMAERS-Unit establishes one trusted channel to one CSP-L via PACE.

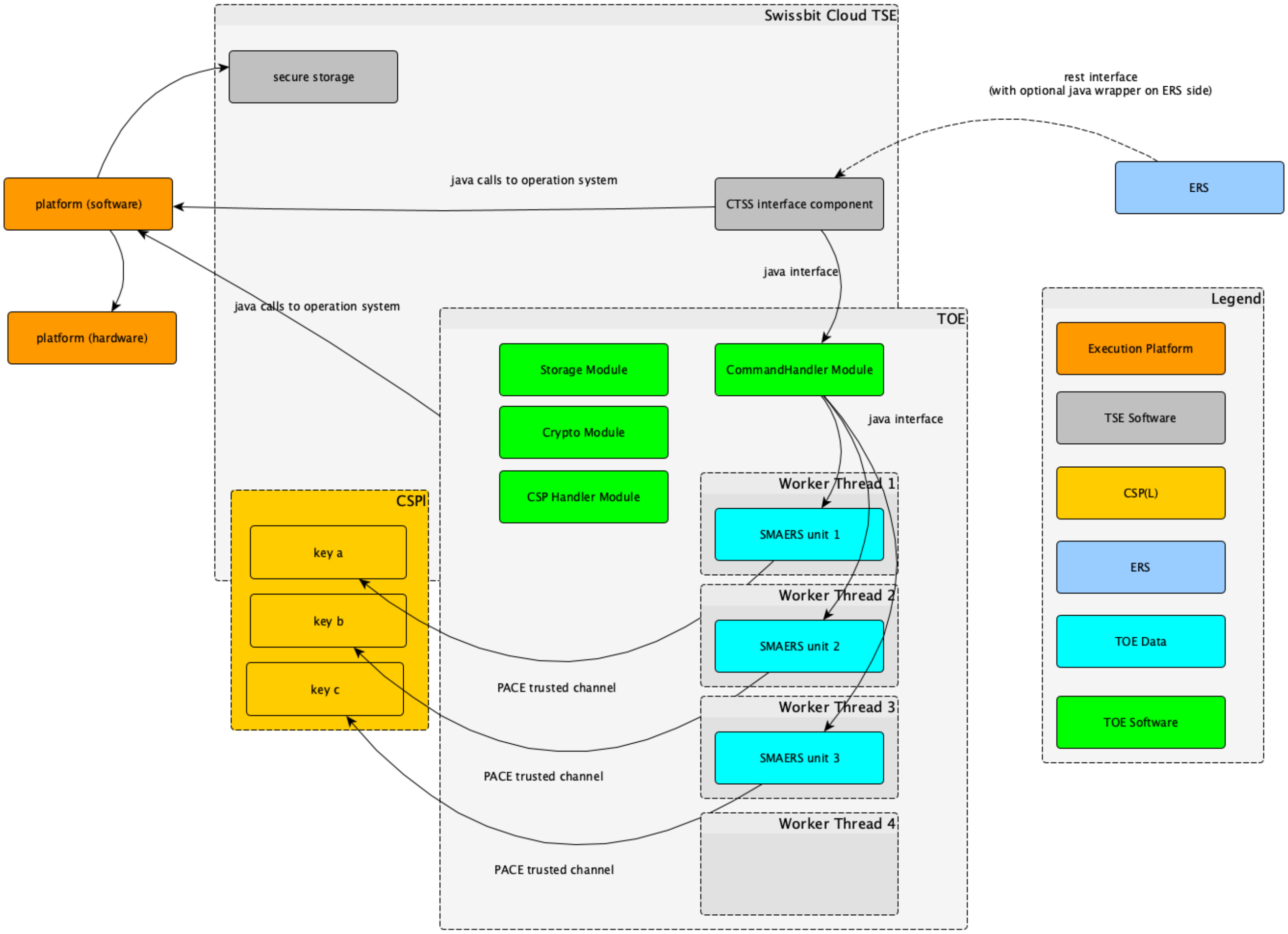

Figure 3. The worker threads and SMAERS units within the TOE. Note that each SMAERS unit uses each green module. Arrows were dismissed to not overload the image.

1.4.3. TOE boundaries

physical boundaries

As a software type TOE, the TOE itself has no physical boundary. The deliverables of the TOE are:

Table 3. TOE Delivery Items

Delivery Item	Version
Swissbit Cloud SMAERS-v1.0.5.jar	1.0.5
Swissbit Cloud SMAERS - Guidance Manual **[SMAERS-AGD]**	1.0.5

logical boundaries

The logical boundaries of the TOE are formed by the interface of the jar file, which contain the TOE. It exposes an java interface to the CTSS component of the TSE. Besides the TOE requires to store its data on the execution platform. Via a second interface, the TOE communicates with one or multiple CPSIs.

1.4.4. Integration of the TOE in the Environment

The TSE, which contains the TOE is installed by the TSE operator in a cloud of a cloud service provider. The TSE operator also has to provide and configure the secure storage and instantiate smaers units within the TOE. Last but not least, the operator has to give tax payers access to dedicated smaers units.

2. Conformance Claims

2.1. CC Conformance Claim

As defined by the references [CC1], [CC2] and [CC3], this Security Target:

- conforms to the requirements of Common Criteria v3.1, Revision 5 and
- is Part 2 extended and
- is Part 3 conformant.

2.2. PP Claim

This Security Target claims strict conformance to [PP-SMAERS], including the package Trusted Channel between TOE and CSP.

2.3. Package Claim

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

2.4. Conformance Rationale

The TOE as described in this ST is a product that allows to protect transaction data of Electronic Record Keeping Systems by using a certified cryptographic service provider (CSP).

It therewith falls directly into the classes of TOEs that are defined by [PP-SMAERS]. In chapter 1.2 [PP-SMAERS] states:

> The TOE is a security module application as part of the security module of a certified technical security system (CTSS) for electronic record-keeping systems (ERS). Figure 1 describes the interaction between TOE and non-TOE components.

[PP-SMAERS] requires strict conformance which is claimed by this Security Target.

3. Security Problem Definition

3.1. Introduction

The Security Problem Definition is identical to the one of [PP-SMAERS]. The changes as required due to the use of the functional package for the PACE channel as described in chapter 7 of [PP-SMAERS] have been made. No further changes were made here by the authors of this Security Target.

Assets

The assets of the TOE are

- the transaction data provided by the CTSS interface component, where authenticity and integrity including completeness of the transaction data shall be protected, i.e. verification of the transaction log messages shall determine whether the transaction data was received from the CTSS interface component, and modifications and gaps shall be detectable,
- the transaction number (as part of the transaction data) that enumerates transactions. The transaction number must be continuously increasing without gaps.
- the audit records imported from the CSP and exported as audit logs to the CTSS interface component, the system logs and transaction logs
- the update code package (UCP) and the UCP version number
- the PACE password to setup the trusted channel to the CSP (only in case the package 'Trusted Channel' is claimed).

The CSP protects and enumerates its audit records against undetected modification and gaps.

ST Application note 1: This TOE hosts multiple SMAERS units. Each of them has a private transaction number. Also transaction data provided by the CTSS interface and audit records imported from the CSP-L are associated with a dedicated SMAERS unit. Note that all SMAERS units use one CSP-L, but each has a private trusted channel to the CSP-L. So the PACE password is an asset of the unit.

Table 4. Assets to be protected by the TOE

Asset	Protection
transaction data	authenticity, integrity
transaction number	authenticity, integrity
audit logs/audit records, system logs and transaction logs	authenticity, integrity
update code package	authenticity

UCP version number	integrity

Users and subjects

The users and subjects defined below are distinct from the role model in [BSI-TR-03153]. Users and roles defined in the latter, including e.g. the taxpayer acting as (CTSS-)administrator, converge in the CTSS interface component.

The TOE knows users as external active entities communicating with the TOE as

- *electronic record-keeping system (ERS)*,
- *CTSS interface component*,
- *CSP*,
- *(SMAERS) administrator*.

ST Application note 2: Note that all roles apply to the SMAERS-Unit only and are not global roles. I.e. for each unit there is a separate administrator, CSP-L and also for each unit an ERS has to identify.

Roles

The TOE knows at least the following roles taken by a user or a subject acting on behalf of a user:

- role *unidentified user*: This role is associated with any user not (successfully) identified by the TOE. This role is assumed for subjects after start-up of the TOE and deactivated *CTSS interface component*. The TOE allows users in this role to run self-test of the TOE.
- role *administrator*: A user in this role is allowed to perform management functions. The administrator subject is acting on behalf of a human user after successful authentication as administrator until logout.
- role *CTSS interface*: A subject in this role is started automatically after start-up of the TOE if the CTSS interface role is activated and the *CTSS interface component* and the *CSP* are successfully tested according to FPT_TEE.1. The ERS uses the CTSS role. It is allowed to generate *system logs*.
- CSP role: A subject in this role is allowed to import audit records from CSP and to export *Audit logs* to the *CTSS interface component*. In addition the CSP role is allowed to start the update process. A subject in *CSP role* is started automatically after start-up of the TOE if the *CSP* is successfully tested according to FPT_TEE.1.
- Tr administrator role: A subject in this role is allowed to initialize and decommission the TOE. It also is allowed to register and deregister client Ids. It is allowed to generate *system logs*.
- Logger role: A subject in this role is allowed to import *Transaction Data* from *CTSS interface*

component, to generate *transaction logs* and *system logs*, and to export *transaction logs* and *system logs* to the *CTSS interface component*.

ST Application note 3: The TOE does not have a dedicated administrator model. The role (SMAERS) Administrator depends on the SMAERS unit and each has a dedicated one. The roles *Logger* and *Tr administrator* were required to be added, because **[BSI-TR-03151]** section 3.2.2.1 requires them to be managed by the SMAERS.

Objects

The TSF operates on the following types of user data objects

- *transaction data* (TD),
- *audit records*,
- *data-to-be-signed* (DTBS),
- *protocolData with signature* containing the time stamp, the signature counter and the digital signature as generated by the CSP (cf. **[BSI-TR-03153]** and **[BSI-TR-03151]**),
- *log messages* (LM) as *transaction log*, *system log*, or *audit log,*
- *update code package* (UCP)
- *commands* (type of operation).

The formats of *transaction data* and *log messages* meet the **[BSI-TR-03151]**.

The CTSS interface component provides *transaction data* as data to be certified by means of *transaction logs* (cf. below).

Audit records are data imported from the CSP.

The *data-to-be-signed* compiled by the TSF and sent to the CSP for signing and time stamping consists of

- certified data i.e.
  - in case of a *transaction log*: the *transaction data* with the type of the certified data *transaction log*, object identifier (id-SE-API-transaction-log): bsi-de (0.4.0.127.0.7) applications (3) sE-API (7) sE-API-dataformats(1) 1 (cf. **[BSI-TR-03153]**, chapter 2.3.1)
  - in case of a *system log*: the security related events with the type of the certified data *system log*, object identifier (id-SE-API-system-log): bsi-de (0.4.0.127.0.7) applications (3) sE-API (7) sE-API-dataformats(1) 2 (cf. **[BSI-TR-03153]**, chapter 2.3.2)
  - in case of an *audit log*: the *audit record* with the type of the certified data *audit log*, object

identifier (id-SE-API-audit-log): bsi-de (0.4.0.127.0.7) applications (3) sE-API (7) sE-API-dataformats(1) 3 (cf. **[BSI-TR-03153]**, chapter 2.3.3)

- protocol data generated by the TSF
- the *transaction number*,
- the *keyID* as a hash value of the signature-verification key,
- the *type of the operation* as name of the API function whose execution is recorded by the *log message*, i.e. *StartTransaction*, *UpdateTransaction* or *FinishTransaction*,
- the *optional protocol data* (may be empty).

The CSP adds to the *data-to-be-signed*

- the point in *time* when the *log message* was created,
- the *signature counter* that enumerates the signatures created with the signature-creation key.

Refer to **[BSI-TR-03153]** for details of the log messages format.

The Update Code Package (UCP) is a complete software package that is managed by the secure platform and its operating system that executes the SMAERS application. The operating system of the secure platform performs an update of the SMAERS application, it is required that the verification of the UCP is performed by the operating system prior to installation. Depending on the update procedure of the operating system either the new TOE alone or the old TOE and the new TOE together perform an *upgrade* by exporting and importing TSF data into the new TOE.

3.1.1. Security attributes

Administrators known to the TOE have the security attributes stored in an *Authentication Data Record*

- *user identity* (User-ID),
- *authentication reference data*,
- *role* with detailed access rights gained after successful authentication.

The *CTSS interface component* and the CSP known to the TOE, have at least the security attributes *identity*, cf. **FIA_ATD.1**

Passwords as *authentication reference data* have the security attributes

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

The *transaction data* (TD) have the security attributes

- *clientID* to determine the signature-creation key to be used for signing the *Transaction log* and the *keyID* to be included in the protocol data of the *Transaction log*,
- *type of the operation* to determine the actual transaction as *StartTransaction*, *UpdateTransaction* or *FinishTransaction*.
- *transaction number* to assign the TD to an ongoing transaction and enumerating the transactions continuously increasing without gaps.

ST Application note 4: In the course of this document, the *clientID* in use consists of an Identifier of the SMAERS unit and an identifier of the ERS. Whenever this ST speaks of a *cliendID* this data identifies the SMAERS unit as well as the ERS.

ST Application note 5: The TOE / CTSS supports the usage of multiple signatures keys. These are in a 1:1 relation to the SMAERS units of the TOE, i.e. each SMAERS unit uses exactly one key and each key is used by exactly one SMAERS unit. Therefore, each SMAERS unit maintains a transaction counter to assign transaction numbers to log messages.

The TOE accepts *transaction data* only if the clientID is known and mapped to a signature key in the CSP (*keyID*).

The TOE manages, for each known keyID, the last assigned transaction number and the transaction numbers of the ongoing transactions. If the *type of the operation* of imported transaction data is *StartTransaction*, then a new transaction is started and the TOE generates a new *transaction number* by addition of 1 to the last assigned *transaction number*, includes this value in the protocol data of the *transaction log* returned to the CTSS interface component, and add this value to the list of ongoing transaction. If the type of the operation is *UpdateTransaction* or *FinishTransaction* and meets the *transaction number* of an ongoing transaction, the *transaction number* in the transaction data is imported and assigned to the protocol data of the *transaction log*. If the *type of the operation* is *FinishTransaction* or the transaction is terminated by the TOE, the *transaction number* is removed from the list of ongoing transactions cf. **[BSI-TR-03153]**.

A *UCP* has the security attributes

- *issuer*: identifier of the authorized issuer of the UCP signing the UCP,
- *signature*: digital signature of the UCP generated by the authorized issuer,
- version number

Log messages

Log messages include at least the following security attributes and the signature used by the tax inspector of the cash register inspection

- *signature counter* enumerating the *log message* continuously increasing without gaps,
- *time stamp* as time when the *log message* was created,
- *keyID* to determine the certificate to be used for the verification of the digital signatures as a check value of the transaction data.

The following security attributes are conditional in log messages:

- Transaction logs contain the security attribute *transaction number* assigning the *log message* to the transaction of the electronic record-keeping system and the *type of operation*, i.e start, update or finish transaction.
- System logs contain the security attribute *event* assigning the *log message* to the security related event of the TSF.
- Audit logs contain the security attribute *audit record* assigning the log message to security related events of the CSP.

3.2. Threats

T.EvadTD: Evading *Transaction Data*

The attacker prevents sending to the TOE legally required *transaction data* in order to avoid generation of valid *Transaction logs*.

T.ManipTD: Manipulation of *Transaction Data*

The attacker manipulates *transaction data* sent by the electronic record-keeping system though the CTSS interface component to the TOE, or generates forged *transaction data* and sends them to the TOE in order to generate wrong *transaction logs*.

T.ManipDTBS: Manipulation of *Data-To-Be-Signed-And-Time-Stamped*

The attacker generates forged or manipulates *Data-To-Be-Signed* sent for signing and time stamping to the CSP. A forged *transaction log* may result in forged transaction data provided for cash inspection. A forged *audit log* or *system log* may result in faulty interpretation of the transaction data.

T.ManipLM: Manipulation of a *Log message*

The attacker manipulates without detection a *log message* exported to the CTSS interface component. This log message is then used for cash inspection.

T.ManipLMS: Manipulation of a *Log message sequence*

The attacker manipulates without detection the *log message sequence* exported to the CTSS interface component. This log message sequence is then used for cash inspection.

T.ManipTN: Manipulation of *Transaction Number*

The attacker manipulates the TOE's internal *transaction number* used in *log messages*.

T.FaUpD: Faulty *Update Code Package*

An attacker deploys an unauthorized manipulated *update code package* or restores a previous TSF implementation enabling attacks against integrity of TSF implementation, or confidentiality and integrity of user data or TSF data after installation of the manipulated *update code package*.

Application note 1: The taxpayer is the subject that owns and operates the ERS and CTSS (either directly or indirectly). The taxpayer is assumed to use an ERS equipped with a CTSS, to prevent misuse of the ERS by unauthorized persons, and to correctly tally all transactions with the ERS as required by law (c.f. **OSP.SecERS** and **OSP.ProtDev**). The TOE does not protect against threats that result from temporarily or permanently not using an ERS as required by law. The taxpayer is however also considered as potential attacker, who may use a manipulated CTSS or manipulates logs after they were produced by the CTSS.

Consideration of Application note 1: The TOE takes this application note into account and considers the dual role of the taxpayer (as required trustworthy supplier of transaction data), operator of the TOE, and as potential attacker appropriately.

3.3. Organizational security policies

OSP.SecERS: Secure use of the Electronic Record-Keeping System

The taxpayer shall use an electronic record-keeping system to generate accounts, records and receipts. The electronic record-keeping system shall record separately, correctly, completely, and in real time accounts and records of all transactions that are legally required; cf. **[FCG]**, Section 146a (1), Sentence 1. The receipt shall include besides the transaction data the points in time when the transaction is started, completed or terminated, and the transaction number provided by the certified security device; cf. **[KSV]**, Section 6, Sentence 1.

OSP.CerTSEcDev: Certified Security Device

The electronic record-keeping system and the accounts and records generated by the electronic record-keeping system shall be protected by a certified security device; cf. **[FCG]**, Section 146a (1), Sentence 2. The security module of the certified security device generates time stamps of the start, completion, and termination of a transaction, as well as a transaction number; cf. **[KSV]**, Section 2, Sentence 3.

OSP.ProtDev: Protection of Electronic Record-Keeping System and Certified Security Device

The taxpayer shall correctly operate the electronic record-keeping system (cf. **[FCG]**, Section 379 (1), Sentence 1, Number 4), and correctly protect the electronic record-keeping system and the certified security device; cf. [FCG], Section 379 (1), Sentence 1, Numbers 5.

OSP.ValidTrans: Validation of transactions

A sequence of transactions is valid if (1) all Log messages meet the requirements for content defined in [KSV] section 2, (2) their check values according to [KSV] section 2 sentence 2 number 7 are valid digital signatures, (3) the transaction numbers are consecutive increasing without gaps (cf. [KSV] section 2 sentence 4), and (4) the points in time when the transaction starts are monotonic increasing. The sequence of Log messages support detection of incomplete transactions and manipulations.

OSP.Update: Authorized Update Code Packages

Update Code Packages are delivered to the TOE from the platform and are signed by the authorized issuer. The platform verifies the authenticity of the received Update Code Package before installation.

Application note 2: The update is performed by the platform provided by the operational environment, c.f. OE.CSPPlatform for the platform architecture or **OE.SMAERSPlatform** for the client-server architecture.

Consideration of Application note 2: The TOE design takes this into account. Especially the TOE is operated by a trustworthy tse operator, who is guided by the guidance documents accordingly.

OSP.AdditionalCrypto: Additional Cryptographic Functionality of the TOE

The TOE implements a hash function to store authentication reference data in a secure way. In addition, the TOE implements hash last round on card to send a partial hash (and the hash functions internal state) to the CSP-L to create log messages.

ST Application note 6: The last OSP was added by the ST authors to enable a proper modelling of additional cryptographic functionality, that this TOE requires although it was not present in the Protection Profile.

3.4. Assumptions

A.SMAERSPlatform: Secure platform storage

The platform that executes the TOE provide mechanisms to preserve the confidentiality, integrity and to prevent rollback of stored sensitive objects, including the TOE software iTSElf.

A.CSP: Cryptographic service provider

A CSP is *either* remotely accessible via trusted channel to the TOE (client-server architecture) and certified as compliant to [PPC-CSP-TS-Au], [PPC-CSP-TS-Au-Cl], or [PPC-CSPLight-TS-Au-Cl] running on hardware that meets Appendix: Operational Requirements for CSPlight (in [PP-

SMAERS]) as well as the requirements in chapter 1.2 section "TOE Life Cycle" (of **[PP-SMAERS]**) *Or*, the operational environment provides a cryptographic service provider for the TOE that is certified as compliant to **[PPC-CSP-TS-Au]** or **[PPC-CSP-TS-Au-Cl]** (platform architecture). The CSP exports audit records in form of audit logs meeting **[BSI-TR-03151]**.Also, the CSP must provide a fully defined API description.

A.ProtComCSP: Protection of communication between TOE and CSP

The integrity of the communication data between TOE and CSP in the client-server architecture is protected via a trusted channel, and the security target must claim the package *Trusted Channel*, defined in Chapter 7. In case of the platform architecture of the CSP,the CSP provides a secure execution environment for the TOE and protects the integrity of communication data with the TOE directly using the security services of the CSP.

A.ProtComERS: Protection of communication between TOE and Electronic Record-Keeping System

The electronic record-keeping system provides transaction data whenever a transaction starts, transaction data are updated, or when the transaction is completed or terminated. The ERS and the TOE must be contained in the same physical operational environment that must protect the integrity of communication data between the TOE and the electronic record-keeping system see **Figure 4**.

A.VerifLMS: Verification of Log message Sequences

The operational environment verifies the digital signatures, the transaction numbers and the time stamps of *log messages* in sequence in order to detect forged or missing *log messages*. The certificate of the signature-verification data is securely distributed to the tax inspector. The tax inspector ensures that the transactions are created by a certified security module, e.g. in form of test transactions.

A.Admin: Trustworthy Administrator

The administrator acts in a trustworthy way and must be independent of the taxpayer (cf. Application note 1).

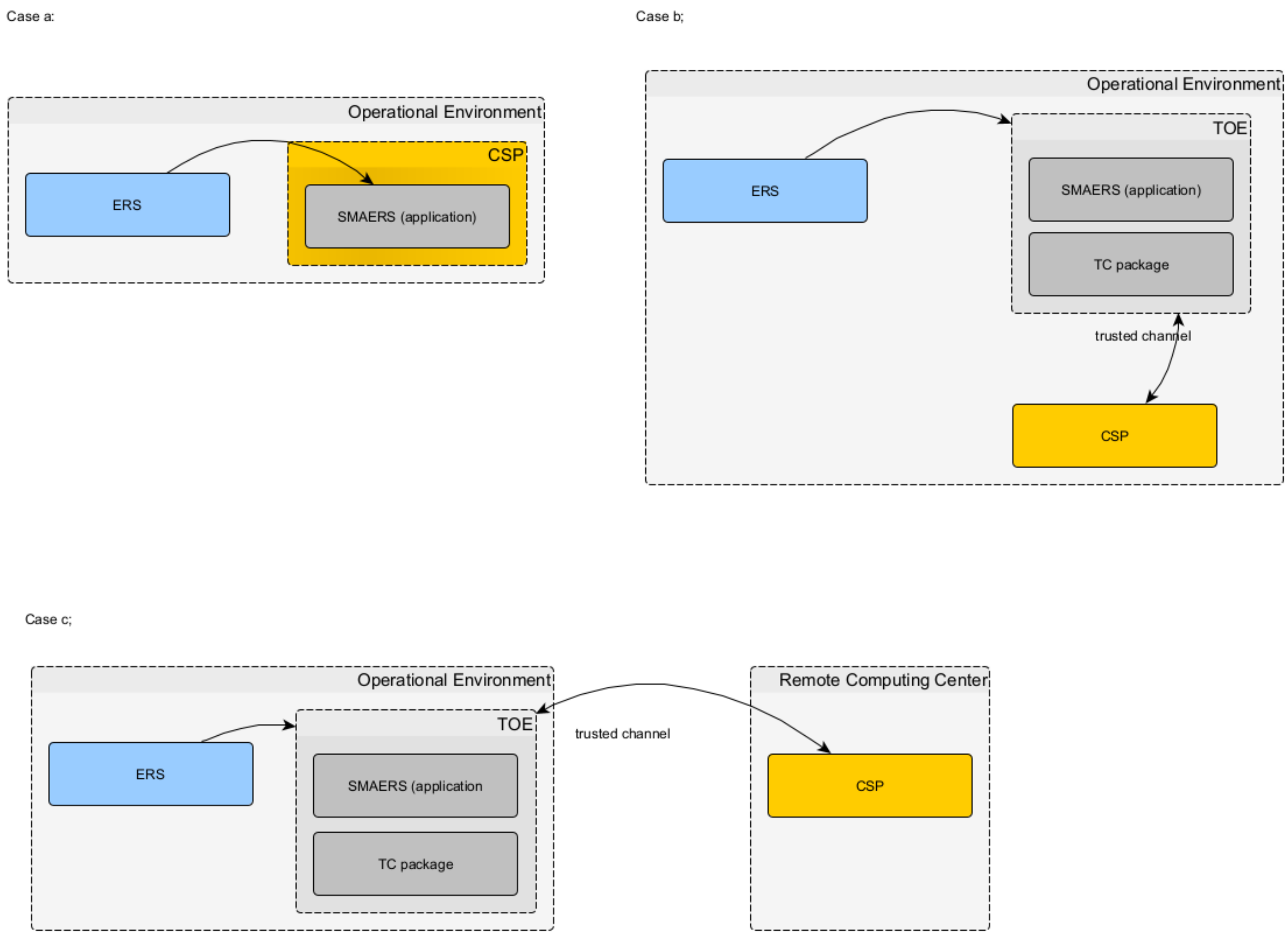

Figure 4. The TOE is always operated as a local component. a) platform architecture b) client-server architecture with local computing center c) client-server architecture with remote computing center

ST Application note 7: Figure 4 shows the operation modes of the TOE as specified by [PP-SMAERS]. It is a remake of an image of [PP-SMAERS], which shows the same as in PP but in the style of the images of this ST. Swissbit Cloud SMAERS is always operated in in mode c), i.e. the TOE is operated in the neighborhood of the TSE and connected with a CSPl in a remote computing center. Note that also the operation of the TOE itself is not under control of the tax payer, but managed by a trustworthy TSE operator.

Besides, this TOE implements the client-server architecture. So this ST uses the functional package for the PACE channel in chapter 7 of [PP-SMAERS], which adds O.SecCommCSP to the list of Security Objectives of [PP-SMAERS], as required by chapter 7.

4. Security objectives

The Security Objectives chapter is slightly changed to the one of [PP-SMAERS]. This Security Target uses the client-server architecture, so it uses the optional functional package being defined in chapter 7. Correspondingly, slight changes had to be made to adopt the Security Objectives. They are marked with ST Application notes. In addition, one security objective to model the access of the platform to the TOE's AES-hardware was added and also marked with an ST Application Note.

4.1. Security Objectives for the TOE

O.GenLM: Generation of *Log Messages*

The TSF shall generate *transaction logs* containing

- *transaction data*, *transaction number* created by the TSF, and
- time stamps and digital signatures created by the cryptographic service provider.

The TSF shall generate *system logs*. The TSF shall compute a partial hash and provide the partial hash and internal state of the hash function to the CSP-L for signature computation.

ST Application note 8: The last sentence was added by the ST authors to properly model the hash last round on card approach between CSP-L and TOE for signature computation.

O.ImpExp: Import of *Transaction Data* from and Export of *Log Messages* to CTSS Interface Component

The TSF shall import *transaction data* from the electronic record-keeping system through the CTSS interface component, import *audit records* from the CSP and export *log messages* to the CTSS interface component.

O.IAA: Authentication of Administrators

The TOE shall verify the claimed identity of the administrators by means of password. The TOE shall store authentication reference data of administrators and users in hashed form.

ST Application note 9: The last sentence was added by the ST authors to properly model that authentication reference data are stored as salted hash.

O.SecMan: Security Management

The TOE shall restrict the security management of TSF and TSF data to authenticated *administrators* The TSF prevents management of the *transaction number* generation.

O.TEE: Test of External Entities

The TSF shall test the presence and identity of the electronic record-keeping system and cryptographic service provider connected to the TOE, and allow generation of *transaction logs* only if both pass the tests, and must enter a secure state if any test fails.

O.TST: Self-test and Secure State

The TSF shall perform self-tests. The TSF enters a secure state if the self-test fails, or the test of the presence and identity of the electronic record-keeping system fails, or the test of the presence and identity of cryptographic service provider fails. It shall also test for new successfully installed update code packages and the correctness of the increased version number.

O.ImpExpUCP: Import and Export of User Data

The TSF shall securely export the user data and TSF data to the secure storage of the platform and import the user data and TSF data after the successful update process.

O.SecCommCSPTrusted channel between TOE and CSP

The TOE shall protect the integrity of the communication between the TOE and the cryptographic service provider by means of a trusted channel.

ST Application note 10: This TOE implements the client-server architecture. So this ST uses the functional package for the PACE channel in chapter 7 of **[PP-SMAERS]**, which adds **O.SecCommCSP** to the list of Security Objectives of **[PP-SMAERS]**, as required by chapter 7.

4.2. Security objectives for the operational environment

OE.ERS: Trustworthy Electronic Record-Keeping System

The taxpayer shall correctly use an electronic record-keeping system that provides separately, correctly, completely and in real time all *transaction data* that are legally required for the generation of *log messages* to the TOE (cf. Application note 1). The electronic record-keeping system shall support testing its presence and identity as an external entity by the TOE. The electronic record-keeping system shall produce receipts including not only the transaction data, but also the points in time whenever a transaction is started, completed or terminated, as well as the transaction number provided by the certified security device.

OE.SMARSPlatform: Secure platform storage

The platform that executes the TOE has to ensure the integrity of the TOE iTSElf and to provide secure storage which protects the integrity and confidentiality of stored security relevant objects as required (cf. Chapter 1.2 “TOE Type”). The platform verifies and installs the UCP.

OE.CSP: Cryptographic Service Provider Component

A CSP must be either remotely accessible via a trusted channel to the TOE (client-server architecture) and certified as compliant to **[PPC-CSP-TS-Au]**, **[PPC-CSP-TS-Au-Cl]**, or **[PPC-CSPLight-TS-Au-Cl]** running on hardware that meets Appendix: Operational Requirements for CSPlight (of **[PP-SMAERS]**).

Or, the operational environment shall provide a cryptographic service provider for the TOE that is certified as compliant to **[PPC-CSP-TS-Au]** or **[PPC-CSP-TS-Au-Cl]**, i.e. using the platform architecture.

The CSP shall export audit records in form of audit logs meeting **[BSI-TR-03151]**.

Application note 3: The Common Criteria Protection Profile Configurations **[PPC-CSP-TS-Au]**, **[PPC-CSP-TS-Au-Cl]**, and **[PPC-CSPLight-TS-Au-Cl]** require the cryptographic service provider to provide security services to digitally sign *transaction data*, to verify a signature of an *update code package*, and for time services. The CSP audit records shall be exported meeting **[BSI-TR-03151]** in order to avoid a transformation of an audit record into a log message. The vendor of the TOE may provide the TOE together with a certified cryptographic service provider.

Consideration of Application note 3: The TOE of this Security Target is connected to one remote CSP-L. The TOE exports the CSP's audit records and exports / stores them accordingly.

OE.CSPPlatform: CSP as a Secure Platform of the TOE

In case of the platform architecture, the CSP provides a secure execution environment and security services for the TOE running on top.

Application note 4: In the typical case of a client-server architecture, the TOE and the CSP are physically separated components and the TOE cannot rely on the CSP as a secure execution platform. Instead, the security target shall claim the package trusted channel (Chapter 7) to protect the integrity of the communication between the TOE and the CSP.

Consideration of Application note 4: The TOE of this Security Target uses the client-server architecture and does not use a CSP as secure execution platform. For this reason it claims the package trusted channel (Chapter 7).

OE.Transaction: Verification of Transaction

The operational environment shall verify the validity of *log message sequences* by verification of the corresponding digital signatures, shall verify the *transaction numbers* as being consecutive without gaps, and shall verify the points in time when the transaction starts as being consecutively increasing with increasing *transaction numbers*, and consider the *log messages*. The taxpayer shall ensure that the cryptographic service provider holds digital signature creation data and a corresponding valid certificate. The certificate shall be securely distributed to the tax inspector.

OE.SecOEnv: Secure Operational Environment

The operational environment shall protect the integrity of the communication between the electronic record-keeping system and the TOE. The administrator shall act in a trustworthy way and is assumed to be the manufacturer or integrator. The administrator must be independent of the taxpayer.

OE.SecCommCSP: Secure communication between TOE and CSP

The security target shall claim the package trusted channel (Chapter 7) to protect the integrity of the communication between the TOE and the CSP in the client-server architecture. In case of the platform architecture, the operational environment shall protect the integrity of the communication between the TOE and the cryptographic service provider.

ST Application note 11: This security target claims Chapter 7 accordingly.

OE.SUCP: Signed Update Code Packages

The manufacturer shall issue digitally signed *update code packages* together with its security attributes.

OE.SecUCP: Secure download and authorized use of *Update Code Package*

The platform shall verify the authenticity of received *update code packages* and install only authentic *update code packages*.

4.3. Security objectives rationale

The following table traces a security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective, and a 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.

Table 5. Security objective rationale

	T.EvadTD	T.ManipTD	T.ManipDTBS	T.ManipLM	T.ManipLMS	T.ManipTN	T.FaUpD	OSP.SecERS	OSP.CertSEcDev	OSP.ProtDev	OSP.ValidTrans	OSP.Update	OSP.AdditionalCrypto	A.CSP	A.SMAERSPlatform	A.ProtComCSP	A.ProtComERS	A.VerifLMS	A.Admin
O.GenLM	x			x	x						x		x						

O.IAA				x							x		x						
O.ImpExp					x						x								
O.SecMan						x					x								
O.TEE	x	x	x	x	x			x											
O.TST				x			x												
O.ImpExpUCP							x					x							
O.SecCommCSP																x			
OE.CSP				x					x					x					
OE.SMAERSPlatform		x	x				x								x				
OE.CSPPlatform			x													x			
OE.ERS	x	x						x											
OE.SecUCP							x					x							
OE.SecCommCSP			x													x			
OE.SecOEnv	x			x	x			x		x							x		x
OE.SUCP							x					x							
OE.Transaction											x							x	

The following part of the chapter demonstrates 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.EvadTD *Evading Transaction Data* is mitigated by:

- The security objective for the TOE O.GenLM requiring the TSF to create *transaction logs* containing *transaction data* and a *transaction number* generated by the TSF, and time stamps and digital signatures, therefore allowing to decide whether presented transaction data have a corresponding transaction data set in the transaction data set sequence.
- The security objective for the TOE O.TEE requiring the TSF to test the presence and identity of the electronic record-keeping system connected to the TOE.
- The security objective for the operational environment OE.ERS requiring the taxpayer to use an electronic record-keeping system that provides completely and in real time all *transaction data* that are legally required for generation of *log messages* to the TOE.
- The security objective for the operational environment OE.SecOEnv requiring the operational environment to protect the communication between ERS and TOE against manipulation and perturbation.

The threat **T.ManipTD** *Manipulation of Transaction Data* is mitigated by:

- The security objective for the TOE **O.TEE** requiring the TSF to test the presence and identity of the CTSS interface component connected to the TOE,
- The security objective for the operational environment **OE.ERS** requiring the taxpayer to use an electronic record-keeping system that provides correctly, completely and in real time all transaction data that are legally required for generation of *log messages* to the TOE,
- The security objective for the operational environment **OE.SMAERSPlatform** requiring the operational environment to protect the TOE against manipulation and misuse.

The threat **T.ManipDTBS** *Manipulation of Data-To-Be-Signed-And-Time-Stamped* is mitigated by:

- The security objective for the TOE **O.TEE** requiring the TSF to test the presence and identity of the CSP connected to the TOE.
- In case of the platform architecture, the **OE.CSPPlatform** "CSP as Secure Platform of the TOE" requires the CSP to provide a secure execution environment. In case of the client-server architecture, the **OE.SMAERSPlatform**.
- The security objective for the operational environment **OE.SecCommCSP** "Secure communication between TOE and CSP" ensures the protection of the integrity of the communication between the TOE and the cryptographic service provider. The operational environment shall protect the integrity of the communication between the TOE and the cryptographic service provider. In case of the client-server architecture, the TOE and the CSP component are physically separated components. The integrity of the communication between the TOE and the CSP shall be protected by means of a trusted channel as provided by the CSP according to **[PPC-CSP-TS-Au][PPC-CSP-TS-Au-Cl][PPC-CSPLight-TS-Au-Cl]** and by the TOE claiming the package trusted channel between the TOE and the CSP, cf. Chapter 7.

The threat **T.ManipLM** *Manipulation of Log messages* is countered by:

- The security objective for the TOE **O.GenLM** "Generation of Log messages" by means of digital signatures generated by the CSP, which allows to detect manipulation of transaction data sets according to **OE.Transaction**.
- The security objective for the TOE **O.IAA** requiring the TSF to authenticate administrators by means of a password.
- The security objective for the TOE **O.TEE** "Test of External Entities" requiring the TSF to test the presence and identity of the CSP connected to the TOE.
- The security objective for the TOE **O.TST** "Self-Test and Secure State" detects failure and prevents generation of transaction data sets if time source is not available or the test of the CSP

fails.

- The security objectives for the operational environment **OE.CSP** “Cryptographic Service Provider Component” ensures the availability of a certified CSP for generation of time stamps and digital signatures, and the distribution of the certificate linked to the taxpayer for signature verification.
- The security objective for the operational environment **OE.SecOEnv** “Secure Operational Environment” protecting the communication between ERS and TOE.

The threat **T.ManipLMS** *Manipulation of a Log Message Sequence* is countered by:

- The security objective for the TOE **O.GenLM** “Generation of Log Messages” requiring the TSF to generate *log messages* containing *transaction data* imported from the electronic record-keeping system, requiring the TSF to generate time stamps whenever a transaction starts, is completed or aborted, and requiring the TSF to create a *transaction number* and a digital signature of the *transaction data* using the digital signature-creation service of the cryptographic service provider.
- The security objective for the TOE **O.ImpExp** “Import of Transaction Data from and Export of Log Message to CTSS Interface Component“ requiring the TSF to import *transaction data* from the electronic record-keeping system through the CTSS interface component and to export *log messages* to the CTSS interface component.
- The security objective for the TOE **O.TEE** “Test of External Entities” requiring the TSF to test the availability of the CTSS interface component and CSP connected to the TOE.
- The security objective for the operational environment **OE.SecOEnv** “Secure Operational Environment” protecting the communication between ERS and TOE.

The threat **T.ManipTN** *Manipulation of Transaction Number* is countered by the security objectives for the TOE **O.SecMan** TSF preventing management of transaction number generation.

The threat **T.FaUpD** *Faulty Update Code Package* is countered by:

- The security objectives for the TOE **O.ImpExpUCP** “Secure Import and Export of User Data” ensuring that user data are exported and imported after successful update process.
- The security objective for the TOE **O.TST** “Self-Test and Secure State” ensuring a correctly increased version number after installation of an update code package..
- The security objective for the operational environment **OE.SUCP** ensures that the authentic *update code packages* are signed and distributed with security attributes.
- The **OE.SecUCP** “Secure download and authorized use of *Update Code Package*” ensures that only authentic UCPs are installed.

- The **OE.SMAERSPlatform** ensures verifying the UCP.

The organizational security policy **OSP.SecERS** *Secure use of the electronic record-keeping system* is directly enforced by:

- The security objective for the TOE **O.TEE** requiring the TSF to test the presence and identity of the ERS as an external entity.
- The security objective for the operational environment **OE.ERS** "Trustworthy Electronic Record-Keeping System".
- The security objective for the operational environment **OE.SecOEnv** "Secure Operational Environment" protecting the communication of ERS and TOE.

The organizational security policy **OSP.CerTSEcDev** *Certified Security Device* is directly enforced by the security objectives for the operational environment **OE.CSP** "Cryptographic Service Provider Component" and the certification conformant to this Protection Profile.

The organizational security policy **OSP.ProtDev** *Protection of ERS and Security Module* is directly ensured by the security objective for the operational environment **OE.SecOEnv** "Secure Operational Environment".

The organizational security policy **OSP.ValidTrans** *Validation of transactions* is enforced by the security objectives for the TOE

- the security objective for the TOE **O.GenLM** "Generation of Log messages" requiring the TSF to generate *log messages* containing *transaction data* imported from the electronic record-keeping system, to generate time stamps whenever a transaction starts, is completed or aborted, and to generate a *transaction number* and a digital signature of the *transaction data* created using the digital signature-creation service of the cryptographic service provider,
- the security objectives for the TOE **O.IAA** "Authentication of Administrators" requiring the TSF to authenticate administrators by means of a password,
- the security objective for the TOE **O.ImpExp** "Import of Transaction Data from and Export of Log Message to CTSS Interface Component" requiring the TSF to import *transaction data* from the electronic record-keeping system through the CTSS interfaAEce component and to export *log messages* to the CTSS interface component.
- the security objective for the TOE **O.SecMan** "Security Management" preventing manipulation of the *transaction numbers* and limiting the authorized manipulation of the time source to administrators.
- The security objective for the operational environment **OE.Transaction** "Verification of Transaction" ensures the condition for verification of the digital signature of the transaction data

set.

The organizational security policy **OSP.Update** *Authorized Update Code Packages* is implemented by the security objective for the operational environment **OE.SUCP** "Signed Update Code Packages" ensuring a digital signature of a *secure update code package* together with its security attributes and the security objectives for the operational environment **OE.SecUCP** "Secure Download and Authorized Use of Update Code Package" ensuring the verification of the digital signature.

The organizational security policy **OSP.AdditionalCrypto** *Additional Cryptographic Functionality of the TOE* is directly enforced by:

- the security objective for the TOE **O.GenLM** "Generation of Log messages" requiring the TSF to compute a partial hash over the Data to be signed.
- the security objectives for the TOE **O.IAA** "Authentication of Administrators" requiring the TSF to store authentication reference data for administrators and users in hashed form.

The assumption **A.CSP** *Cryptographic service provider* is directly implemented by the security objective for the operational environment **OE.CSP** "Cryptographic service provider component".

The assumption **A.SMAERSPlatform** *Secure platform storage* is directly implemented by the security objective for the operational environment **OE.SMAERSPlatform** that requires secure storage of sensitive objects.

The assumption **A.ProtComCSP** *Protection of Communication* between TOE and CSP is directly implemented by the security objectives for the operational environment **OE.SecCommCSP** which requires the protection of the communication between the TOE and the CSP. In case of the platform architecture, the **OE.CSPPlatform** requires the CSP to provide a secure execution environment. In case of the client-server architecture, the TOE and the CSP component are physically separated components. The integrity of the communication between the TOE and the CSP shall then be protected by means of a trusted channel as provided by the CSP according to **[PPC-CSP-TS-Au]**, **[PPC-CSP-TS-Au-Cl]**, or **[PPC-CSPLight-TS-Au-Cl]** and by the TOE claiming the package trusted channel, cf. Chapter 7.

ST Application note 12: This security target claims Chapter 7 accordingly. Therefore **A.ProtComCSP** is directly implemented by **O.SecCommCSP**.

The assumption **A.ProtComERS** *Protection of Communication between TOE and Electronic Record-Keeping System* is directly implemented by the security objective for the operational environment **OE.SecOEnv** "Secure Operational Environment" protecting the integrity of the communication between the electronic record-keeping system and the TOE.

The assumption **A.VerifLMS** *Verification of Log Message Sequences* is directly implemented by the security objective for the operational environment **OE.Transaction** “Verification of Log message Sequences”.

The assumption **A.Admin** *Trustworthy Administrator* is directly implemented by the security objective for the operational environment **OE.SecOEnv** “Secure Operational Environment”.

5. Extended component definition

The extended components FIA_API.1 and FCS_RNG.1 are used only in the package Package Trusted Channel between TOE and CSP, cf. chapter 7. They are defined in **[PP-SMAERS]**.

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 ST.

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~~.

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. Selections to be filled in by the ST author appear in square brackets and are underlined.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the PP authors are denoted by showing as *italic* text. Assignments to be filled in by the ST author appear in square brackets and are *italicized*.

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

This chapter consists exclusively of the SFRs from **[PP-SMAERS]** and closes open operations in them. It does not contain new SFRs, which are not present in **[PP-SMAERS]**. Be aware, that one SFR (**FCS_COP.1**) from the chapter Trusted Channel was iterated though.

ST Application note 13: All SFRs apply independently to each SMAERS unit. I.e. an authenticated role applies only to one SMAERS unit, not to all. The same holds for the unit's state with respect to a secure state and so on.

6.1.1. Security Management

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*, *administrator*, *CTSS interface role*, and *CSP role*, [*TR administrator role and Logger role*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

ST Application note 14: [BSI-TR-03151] section 3.2.2.1 requires the CC-TOE to manage the roles logger and Tr administrator. For this reason they had to be included them here. Note that *Tr administrator* and *Logger* require role *CTSS interface role* to be able to login. For this reason, they both can be seen as a refinement of *CTSS interface role*, because an authenticate *Logger* or *Tr administrator* always implies the TOE to have *CTSS interface role*.

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 behavior (cf. FMT_MOF.1),*
2. *management of authentication reference data (cf. FMT_MTD.1/AD, FMT_MTD.3/PW),*
3. *management of security attributes (cf. FMT_MTD.3/PW, FMT_MSA.3, FMT_MSA.4),*
4. *[management of acceptable ERS clientIDs]*

FMT_MOF.1: Management of security functions behavior

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 and disable the function ~~s~~ password authentication according to* ***FIA_UAU.5.2****, clause (2) if defined* to *administrator,*
2. ***determine the behavior of and modify the behavior of*** **the function** ***FDP_ACF.1/LM by definition of a life time limit of ongoing transactions after which the transaction is terminated by the TSF*** **to ~~administrator~~ [refinement: none],_**
3. ***determine the behavior of*** **the function** ***FPT_TEE.1 by definition of the identity and features to be tested of ERS*** **to** ***administrator,***
4. ***determine the behavior of*** **the function** ***FPT_TEE.1 by definition of the identity and features to be tested of CSP*** **to** ***administrator,***
5. ***determine the behavior of and modify the behavior of*** **the function** ***FPT_TEE.1 in case the test of CTSS interface component or CSP fails*** **to** ***administrator.***
6. ***determine the behaviour of and modify the behaviour of*** **the functions** *select the auditable events* ***according to FAU_GEN.1/SYS*** **to** ***administrator,***
7. ***determine the behaviour of and modify the behaviour of*** **the functions** *automatic export of audit* ***trails according to FAU_STG.3.1/SYS clause (1)*** **to** ***administrator***

Application note 5: The refinements of **FMT_MOF.1**, bullet (2) to (7) are made in order to avoid iterations of the component. The life time of a transaction starts with receiving the *transaction data* with *type of operation* being *StartTransaction*.

Consideration of Application note 5: The application note has no implications to this Security Target.

FMT_MSA.1: Management of security attributes

Hierarchical to

No other components.

Dependencies

- [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
- FMT_SMR.1 Security roles
- FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1

The TSF shall enforce the *log message SFP* and *update SFP* to restrict the ability to

1. *define the set of accepted values of the security attributes "clientID" to ~~CTSS interface role~~* ***[refinement: Tr administrator role or Logger role],***
2. ***define depending on the clientID the identity of the signature-creation key (keyID) to be used for the transaction log*** **to** ***~~CTSS interface role~~*** *[refinement: None]**,***
3. ***define the identity of the signature-creation key (keyID) to be used for the system log and audit logs*** **to** ***~~CTSS interface role~~*** *[refinement: None]**,***
4. ***increase by 1 the internally stored security attribute "transaction number" when transaction is started*** **to** ***subjects in CTSS interface role,***
5. ***modify the TD security attribute "transaction number" imported from the TD*** **to** ***none,***
6. ***increase the security attribute "version number" of UCP after successful installation*** **to** ***CSP role.***

Application note 6: The refinements of **FMT_MSA.1** are made in order to avoid iteration of the component.

Consideration of Application note 6: The application note has no implications to this Security Target.

ST Application note 15: The definite of the mapping from clientID to keyID is here trivial. Remember each SFR applies independently to each SMAERS unit, so here we have only one keyID (the one of the SMAERS unit the SFR applies to). Therefor all configured clientIDs (in this unit) use the one keyID.

FMT_MSA.3: Static attribute initialization

Hierarchical to

No other components.

Dependencies

- FMT_MSA.1 Management of security attributes
- FMT_SMR.1 Security roles

FMT_MSA.3.1

The TSF shall enforce the *log message SFP* and *update SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the *none* to specify alternative initial values to override the default values when an object or information is created.

6.1.2. User identification and authentication

FIA_ATD.1 User attribute definition

Hierarchical to

No other components.

Dependencies

No dependencies.

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to ~~individual users~~ **administrator**:

1. *identity,*
2. *authentication reference data,*
3. *role*

and

a. **security attribute *identity [and clientID]* belonging to the ERS**
b. **security attribute *identity [and PACE Password]* belonging to the CSP.**

Application note 7: The refinements distinguish between the sets of security attributes maintained for authenticated user administrator, and the tested user ERS and CSP according to **FPT_TEE.1**. The security attributes are defined for users by the dministrator according to **FMT_MSA.1**.

Consideration of Application note 7: This Security Target separates the security attributes accordingly.

FMT_MTD.1/AD Management of TSF data - Authentication data

Hierarchical to

No other components.

Dependencies

- FMT_SMR.1 Security roles
- FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/AD

The TSF shall restrict the ability to

1. *delete and create* the *authentication data record of all authorized users* to *administrator*.
2. ***modify* the *authentication reference data* to *the corresponding authorized user*.**

ST Application note 16: The Protection Profile contained a footnote at the **create** of **FMT_MTD.1.1/AD** which contained the following text: "create" denotes initial creation and setting a new value in case a user forgot/lost their authentication data

FMT_MTD.3/PW Secure TSF data - Password

Hierarchical to

No other components.

Dependencies

FMT_MTD.1/AD Management of TSF data

FMT_MTD.3.1/PW

The TSF shall ensure that only secure values are accepted for *passwords* **and enforce changing initial passwords after first successful authentication of the user to a different secure operational password.**

FIA_AFL.1 Authentication failure handling

Hierarchical to

No other components.

Dependencies

FIA_UAU.1 Timing of authentication

FIA_AFL.1.1

The TSF shall detect when [*3*] unsuccessful authentication attempts occur related to [*authentication with either administrator PIN, or PUK (the attempts are counted per credential, not in total)*].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been [met], the TSF shall *[in case of a PIN, block the corresponding reference data for further use. For each PIN, the reference data can be reset using the PUK of the same role. This resets the number of failed authentication tries for this credential to 0. In case of PUK the PUK is blocked for (2 to the power of (failed retries -3)) seconds].*

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.*

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.*

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. *A subject is associated with attribute ´identity´ and ´CTSS interface role´ after the ERS is successfully tested according to **FPT_TEE.1** **[refinement: and is associated afterwards with attribute ´Logger role` or ´Tr administrator role´ after authenticating with the corresponding PIN]**.*
2. *A subject is associated with attribute ´identity´ and ´CSP role´ after the CSP is successfully tested according to **FPT_TEE.1**.*
3. *A subject is associated with attribute ´identity´ and ´administrator´ role after successful authentication.*

FIA_UID.1 Timing of identification

Hierarchical to

No other components.

Dependencies

No dependencies.

FIA_UID.1.1

The TSF shall allow *self test according to* ***FPT_TST.1*** 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.

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. *testing of external entity ERS according to* ***FPT_TEE.1*** *and start the subject CTSS interface component if testing was successful and the role CTSS interface component is activated,*
3. *testing of external entity CSP according to* ***FPT_TEE.1*** *and start the subject CSP if testing was successful,*
4. [*allow the administrator, TR administrator or Logger to reset the role's PIN if correct PUK value for this role is provided allow unidentified user to retrieve the TSE description and TSS initialization state allow unidentified user to fetch the SMAERS-Unit's time and time until next self test is required*]

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

FIA_UAU.1.2

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

FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to

No other components.

Dependencies

No dependencies.

FIA_UAU.5.1

The TSF shall provide *password authentication* to support user authentication.

FIA_UAU.5.2

The TSF shall authenticate any user's claimed identity according to the *rule that*

1. *password authentication shall be used for an administrator*
2. [
    a. *successful PACE-Channel establishment shall be used for CSP role*
    b. *provision of valid ERS clientIDs shall be used for CTSS Interface role*
    c. *PIN authentication shall be used for Logger and Tr administrator*
    d. *administrator and Tr administrator are automatically de-authenticated after five minutes of inactivity*
    e. *Logger is automatically de-authenticated after sixty minutes of inactivity*]

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 *power on or reset*

FCS_COP.1/HashPasswords Cryptographic Operation

Hierarchical to: No other components.

Dependencies

- [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
- FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HashPassword

The TSF shall perform *[hashing of authentication reference data]* in accordance with a specified cryptographic algorithm *[SHA256]* and cryptographic key sizes *[256 bits]* that meet the following: *[[FIPS_180-4]]*.

ST Application note 17: This SFR was introduced to allow the TOE to store the authentication reference data for **FIA_UAU.5.2** (1) and **FIA_UAU.5.2** (2) (c), i.e. the PINs and PUKs of *Administrator*, *TR Admin*, and *Logger* not as clear text, but as a salted hash.

6.1.3. User data protection

FDP_ACC.1/LM Subset access control – Access to Logging

Hierarchical to

No other components

Dependencies

FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/LM

The TSF shall enforce the *log Message SFP* on

1. *subjects:*
    a. *subject acting for CTSS interface component,*
    b. *subject acting for CSP;*
2. *objects:*
    a. *transaction data,*
    b. *audit record,*
    c. *data-to-be-signed,*
    d. *protocolData with signature,*
    e. *log message,*

   f. *commands;*

3. *operations:*
   a. *import,*
   b. *export.*

FDP_ACF.1/LM Security attribute based access control – Access to TDS

Hierarchical to

No other components.

Dependencies

- FDP_ACC.1 Subset access control
- FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/LM

The TSF shall enforce the *log Message SFP* to objects based on the following:

1. *subjects:*
   a. *subject in CTSS interface role with security attribute activated or deactivated.*
   b. *subject in CSP role;*
2. *objects:*
   a. *transaction data,*
   b. *audit record,*
   c. *data-to-be-signed,*
   d. *protocol data with signature,*
   e. *log message*
   f. *commands.*

FDP_ACF.1.2/LM

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

1. *A subject in activated CTSS interface role is allowed to*
   a. *import the transaction data from the CTSS interface component according to* ***FDP_ITC.2/TD*** ***[refinement: if role Logger is authenticated],***

   b. *import commands from activated CTSS interface component*
   c. *export the DTBS of transaction log to the CSP according to* ***FDP_ETC.2/DTBS****,*
   d. *import the protocolData with signature from the CSP according to* ***FDP_ITC.2/TSS****,*
   e. *export the transaction log to the CTSS interface component according to* ***FDP_ETC.2/LM****.*

2. *~~A subject in activated CTSS interface role~~ [**refinement:** none] is allowed to terminate the transaction after time limit defined according to* ***FMT_MOF.1.1*** *clause (2) is reached.*
3. *A subject in CSP role is allowed to import audit records from the CSP according to* ***FDP_ITC.2/TSS*** *and to export audit logs to the CTSS interface component according to* ***FDP_ETC.2/LM****.*

ST Application note 18: The refinement of <<**FDP_ACF.1.2/LM** (1) is done to be compliant to **[BSI-TR-03153]**, here only the role Logger can start, update and finish transactions. Note that this intentionally not done to the other list entries, as once imported transaction data might have to be signed later, after leaving a secure state before role Logger authenticates again. The refinement in of **FDP_ACF.1.2/LM** (3) is done to be more compliant to **[BSI-TR-03153]**. Here each transaction has to be terminated explicitly by the ERS.

FDP_ACF.1.3/LM

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules [

1. *a subject in activated CTSS interface role is allowed to export the list of open transactions*]

FDP_ACF.1.4/LM

The TSF shall explicitly deny access of subjects to objects based on the rules

1. *a user in other role than CTSS interface role is not allowed to perform actions listed in* ***FDP_ACF.1.2/LM*** *clause (1) and (2).*
2. *a user in other role than CSP role is not allowed to perform actions listed in* ***FDP_ACF.1.2/LM*** *clause (3).*

ST Application note 19: Some of the additional rules of **FDP_ACF.1.3/LM** require the user to have multiple roles at the same time, which is possible (roles and permissions are additive). The term "and" above indicates, that the user has to be authenticated as both roles.

ST Application note 20: To set up the TOE, the TOE requires the *CTSS Interface role* to provide a clientId of an ERS that has been registered before and which gets stored in a system log. It is impossible to open transactions before this registration is done, because the TOE's self test according to **FPT_TST.1** verifies, that this step has been completed.

FDP_ITC.2/TD Import of user data with security attributes – Transaction Data

Hierarchical to

No other components.

Dependencies

- [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
- [FDP_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/TD

The TSF shall enforce the *log message SFP* when importing ~~user data~~ **transaction data** controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/TD

The TSF shall use the security attributes associated with the imported ~~user data~~ **transaction data**.

FDP_ITC.2.3/TD

The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the ~~user data~~ **transaction data** received.

FDP_ITC.2.4/TD

The TSF shall ensure that interpretation of the security attributes of the imported ~~user data~~ **transaction data** is as intended by the source of the user data.

FDP_ITC.2.5/TD

The TSF shall enforce the following rules when importing ~~user data~~ **transaction data** controlled under the SFP from outside of the TOE:

1. *The TSF shall import the transaction data with the security attribute clientID if the clientID is in the set of accepted values according to **FMT_MSA.1**. If the clientID is not in the set of accepted values the TSF must not import the transaction data.*
2. *The TSF shall import the transaction data with the security attribute ´type of the operation`.*
3. *The transaction data shall be imported with the security attribute ´transaction number` if the ´type of the operation` is UpdateTransaction or FinishTransaction, and the transaction number meets a transaction number of an ongoing transaction.*
4. *The TSF shall import audit records from the CSP.*

FCS_COP.1/HashLastRoundOnCard Cryptographic Operation

Hierarchical to: No other components.

Dependencies

- [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
- FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HashLastRoundOnCard

The TSF shall perform *[Hash last round on card of Data to be signed]* in accordance with a specified cryptographic algorithm *[SHA256 last round on card]* and cryptographic key sizes *[256 bits]* that meet the following: *[partial hash computation except last round from* ***[FIPS_180-4]****]*.

ST Application note 21: Hash last round on card is a hashing scheme, were all data, except the last block are hashed. Then the internal state of the hash function and the last block of data are sent to a second entity, which completes the hash calculation (and usually signs the hash). Here the TOE performs all but the last round of the hash calculation and sends all required information to finalize the hash to the CSP-L. This way the amount of data being sent to the CSP-L is always constant, speeding up the signature creation process. Also the CSP-L still ensures that it signs a hash (by completing the hash calculation). This way, the data imported via **FDP_ITC.2/TD** are prepared for signature creation by **FCS_COP.1/HashLastRoundOnCard** and then exported to the CSP-L via **FDP_ETC.2/DTBS**.

FDP_ETC.2/DTBS 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/DTBS

The TSF shall enforce the *log message SFP* when exporting ~~user data~~ **data-to-be-signed**, controlled under the SFP(s), ~~outside of the TOE~~ **to the CSP**.

FDP_ETC.2.2/DTBS

The TSF shall export the user data with the ~~user data's associated~~ security attributes **associated with data-to-be-signed**.

FDP_ETC.2.3/DTBS

The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported ~~user data~~ **data-to-be-signed**.

FDP_ETC.2.4/DTBS

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

1. *Data-to-be-signed shall be exported for generation of a log message with security attribute identifying the private signature key to be used by FDP_DAU.2/TS according to [PPC-CSP-TS-Au][PPC-CSP-TS-Au-Cl][PPC-CSPLight-TS-Au-Cl].*

FDP_ITC.2/TSS Import of user data with security attributes – Time stamp and signature

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/TSS

The TSF shall enforce the *log message SFP* when importing ~~user data~~ **protocolData with signature and audit records**, controlled under the SFP, from ~~outside of the TOE~~ **the CSP**.

FDP_ITC.2.2/TSS

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

FDP_ITC.2.3/TSS

The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the ~~user data~~ **protocolData with signature and audit records** received.

FDP_ITC.2.4/TSS

The TSF shall ensure that interpretation of the security attributes of the imported ~~user data~~ **protocolData with signature and audit records** is as intended by the source of the user data.

FDP_ITC.2.5/TSS

The TSF shall enforce the following rules when importing ~~user data~~ **protocolData with signature and audit records** controlled under the SFP from ~~outside of the TOE~~ **the CSP** :

1. [*none*]

Application note 8: The CSP shall generate and return to the TOE at least the signature counter of the used signature-creation key, the time stamp and the signatures for the *data-to-be-signed* exported by the TOE according to **FDP_ETC.2/DTBS**. The CSP shall generate time stamps according to FDP_DAU.2/TS using time source according to FPT_STM.1 (cf. **[PPC-CSP-TS-Au][PPC-CSP-TS-Au-Cl][PPC-CSPLight-TS-Au-Cl]**). Note, the TOE of the Protection Profile in hand may use the CSP to provide time stamps by an administrator settable internal clock; cf. selection clause (4) in FPT_STM.1.1. If the CSP meets **[BSI-TR-03151]** for the *transaction logs* then the CSP returns a *log message* to the TOE. If the CSP generates the time stamp and signatures with a signature counter, then the TOE shall compile the *log message* according to **[BSI-TR-03153]**. The signature counter and the time stamp of *transaction logs* and of audit data received as system logs may be used to test the CSP according to **FPT_TEE.1**.

Consideration of Application note 8: This TOE compiles *log messages* as required by **[BSI-TR-03153]** as required. In addition, during runs of the test suite, signature counter and time stamp data returned by the CSP-L are used to test the CSP-L, according to **FPT_TEE.1**.

FDP_ETC.2/LM Export of user data with security attributes – Log messages

Hierarchical to

No other components.

Dependencies

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

FDP_ETC.2.1/LM

The TSF shall enforce the *log message SFP* when exporting user data **log message**, controlled under the SFP(s), ~~outside of the TOE~~ **to CTSS interface component**.

FDP_ETC.2.2/LM

The TSF shall export the user data with the user data’s associated security attributes.

FDP_ETC.2.3/LM

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/LM

The TSF shall enforce the following rules when user data is exported from the TOE: *Log messages shall be exported with security attribute*

1. *transaction logs:*
    a. *transaction number of the transaction and identifying the log messages which belongs to the transaction,*
    b. *signature counter of the private signature key used by FDP_DAU.2/TS according to **[PPC-CSP-TS-Au][PPC-CSP-TS-Au-Cl][PPC-CSPLight-TS-Au-Cl]** enumerating all log messages,*
    c. *type of the operation,*
    d. *time stamp when the log message was signed,*
    e. *keyID as hash value of the public key for verification of the signature,*
    f. *signature for verification of the authenticity of the certified data and protocol data.*
2. *system logs:*
    a. *type of the operation or TSF security event*
    b. *signature counter of the private signature key used by FDP_DAU.2/TS according to **[PPC-CSP-TS-Au][PPC-CSP-TS-Au-Cl][PPC-CSPLight-TS-Au-Cl]** enumerating all log messages,*
    c. *time stamp when the log message was signed,*
    d. *keyID as hash value of the public key for verification of the signature,*
    e. *signature for verification of the authenticity of the certified data and protocol data.*
3. *audit records of the CSP shall be exported unchanged as audit logs to the CTSS interface component.*

Application note 9: The CTSS interface component does not implement any security functionality addressed in this PP and imports and stores log message received from the TOE as user data.

Consideration of Application note 9: The architecture of the TSE, to which this TOE belongs implements this requirement. The CTSS interface component of the TSE does not implement any security functionality addressed in this ST.

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to

No other components.

Dependencies

No dependencies.

FPT_TDC.1.1

The TSF shall provide the capability to consistently interpret

1. *clientID,*
2. *type of the operation,*
3. *transaction number,*
4. *signature counter,*
5. *time stamp,*
6. *keyID as hash value of the public key,*
7. *signature*

when shared between the TSF and another trusted IT product.

FPT_TDC.1.2

The TSF shall use *[BSI-TR-03151] and [BSI-TR-03153]* when interpreting the TSF data from another trusted IT product.

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

1. *transaction numbers building a strong increasing sequence without gaps,*
2. *Time stamps of the log messages building a non-decreasing sequence with consideration of adjustments of the CSP's time source.*

Application note 10: The rules may be enforced by internal storing of the *transaction Number* and last time stamp provided by the CSP in the log messages.

Consideration of Application note 10: The TOE stores the last signature counter and last time stamp provided by the CSP. The transaction counter is managed by the TOE iTSElf, so it is also stored.

FMT_MSA.4 Security attribute value inheritance

Hierarchical to

No other components.

Dependencies

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

FMT_MSA.4.1

The TSF shall use the following rules to set the value of security attributes:

1. *The TSF uses the security attribute clientID imported with transaction data to determine the signature-creation key be used by FDP_DAU.2/TS with ECDSA in **[PPC-CSP-TS-Au][PPC-CSP-TS-Au-Cl][PPC-CSPLight-TS-Au-Cl]** to sign the corresponding log message as defined according to **FMT_MSA.1**.*
2. *If the type of the operation of imported transaction data is StartTransaction then the last internally generated transaction number of the respective keyID shall be increased by 1, and this value shall be assigned to the ongoing transaction and the transaction log of imported transaction data.*
3. *If the type of the operation of imported transaction data is UpdateTransaction or FinishTransaction and meets the transaction number of an ongoing transaction then the transaction number of the imported transaction data shall be assigned to the protocol data of the transaction log.*

6.1.4. 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 according to **FPT_TST.1** fails,*
2. *test of ERS according to **FPT_TEE.1** fails,*
3. *test of CSP according to **FPT_TEE.1** fails.*

The TSF shall exit the secure state only if the self-test, the test of the ERS and the test of the CSP are passed.

Application note 11: The self-test according to **FPT_TST.1** and test of external entities according to **FPT_TEE.1** cause the secure state if the self-test or the tests of the ERS or CSP fail. The exit of the secure state requires all conditions listed in the refinement being fulfilled.

Consideration of Application note 11: The TOE only exits the secure state, if the test suite of **FPT_TST.1** and **FPT_TEE.1** was executed successfully.

FPT_TEE.1 Testing of external entities

Hierarchical to

No other components.

Dependencies

No dependencies.

FPT_TEE.1.1

The TSF shall run a suite of tests *during start-up, periodically during normal operation, user initiated shutdown and before exiting the secure state according to* ***FPT_FLS.1*** to check the fulfillment of

1. *ERS identity [clientID] and*
2. *CSP identity [PACE Password].*

The tests include the identification of the TOE to the tested device.

FPT_TEE.1.2

If the test fails, the TSF shall *enter the secure state according to* ***FPT_FLS.1*** *[none additional action]*.

Application note 12: The administrator may by able to define the actions in **FPT_TEE.1** according to **FMT_MOF.1.1** (5). In case of a failure, additional actions may e.g. include reading the stored audit logs. The suite of tests determine whether the configured CSP is available for the TOE and log messages can be signed. The TOE may use the signature counter and time stamps received from the CSP to test it. The signature counter shall increase strong monotonically without gaps because any gap may indicate unauthorized signature-creation. The tests of the CSP should allow the CSP to identify the TOE as user of the CSP, cf. FIA_UID.1.1 clause (2) in **[PP-CSP][PP-CSPLight]**. Please refer for further explanations to the user notes and evaluator notes in CC part 2 **[CC2]**, Chapter J.12.

Consideration of Application note 12: To test the ERS, the ERS has to provide its clientID. This means, the interface between ERS and TOE is used to test the ERS. The test of the CSP-L allows both sides to identify each other.

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, at the request of the authorised user, periodically during normal operation and before exiting the secure state according to* ***FPT_FLS.1*** to demonstrate the correct operation of *parts of TSF*.

FPT_TST.1.2

The TSF shall provide authorised users with the capability to verify the integrity of *TSF data*.

FPT_TST.1.3

The TSF shall provide authorised users with the capability to verify the integrity of *TSF* ***implementation***.

Application note 13: The security attribute “version number” of the UCP is part of the TSF data. During TSF testing, the consistency of the version number has to be checked to detect upgrades or attempted downgrades of the installed code of the TOE. In case of a detected change of the version number, the TOE must follow the UCP SFP and log the events according to **FAU_GEN.1/SYS** .

Consideration of Application note 13: As part of the TSF testing the TOE detects updates and logs them accordingly. In addition, downgrades are also detected and handled accordingly.

6.1.5. Security Audit

FAU_GEN.1/SYS Audit data generation – System Log

Hierarchical to

No other components.

Dependencies

FPT_STM.1 Reliable time stamps

FAU_GEN.1.1/SYS

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

1. start-up and shutdown of the audit functions;
2. all auditable events for the *not specified* level of audit; and
3. *other auditable events*
   a. *system operation commands as specified in [BSI-TR-03151], Appendix A,*
   b. *authentication failure handling (FIA_AFL.1): the reaching of the threshold for the unsuccessful authentication attempts with claimed Identity of the user,*
   c. *failure with preservation of secure state (FPT_FLS.1): entering and exiting secure state,*
   d. *setting of the version number of the UCP and upgrade of stored data,*
   e. *[all audit records as required by [BSI-TR-03151],[BSI-TR-03153] if not covered by the conditions above]*

FAU_GEN.1.2/SYS

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

1. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
2. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*the specified audit event information*].

Application note 14: The security relevant events that have to be logged according to FAU_GEN.1/SYS are part of the system log.

Consideration of Application note 14: The security relevant events that have to be logged according to FAU_GEN.1/SYS are logged as of the system logs.

FMT_MTD.1/SYSCTSS Management of TSF data – System log – CTSS Interface Component

Hierarchical to

No other components.

Dependencies

- FMT_SMR.1 Security roles
- FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/SYSCTSS

The TSF shall restrict the ability to

1. *manual export,*
2. *clear after manual export,*

the *system logs* to *CTSS Interface Component*

FMT_MTD.1/SYSAdmin Management of TSF data – System log -Administrator

Hierarchical to

No other components.

Dependencies

- FMT_SMR.1 Security roles
- FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/SYSAdmin

The TSF shall restrict the ability to

1. *select audited events in* ***FAU_GEN.1/SYS*** *,*
2. *define the number of audit records causing automatic export and clearing of exported audit records according to* ***FAU_STG.3.1/SYS*** *clause (1),*
3. *define the percentage of storage capacity of audit records if actions are assigned in* ***FAU_STG.3.1/SYS*** *clause (2)*

the *system logs* to ~~*Administrator*~~ *[**refinement:** none]*.

ST Application note 22: The TOE always creates all audit events, being listed in this document. Therefore nobody is able to deselect any of them. This justifies the refinement for the first bullet point.

Besides, this TOE always exports audit records immediately as detailed in the consideration of the application note of **FAU_STG.3.1/SYS**. For this reason, there is no threshold and storage to be define by Administrator in bullet point two and three. This justifies the refinement for these remaining points.

FAU_STG.1/SYS Protected audit trail storage – System log

Hierarchical to

No other components.

Dependencies

FAU_GEN.1 Audit data generation

FAU_STG.1.1/SYS

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

FAU_STG.1.2/SYS

The TSF shall be able to *prevent* unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.3/SYS Action in Case of Possible Audit Data Loss – System log

Hierarchical to

No other components.

Dependencies: :FAU_STG.1 Protected audit trail storage

FAU_STG.3.1/SYS

The TSF shall

1. *automatically export audit trails and clear automatically exported audit records if the audit trail exceeds an Administrator defined number of audit records within [1 record]*
2. ***[no actions] if the audit trail exceeds an Administrator settable percentage of storage capacity.***

Application note 15: The ST writer shall perform the open operations in **FAU_STG.3.1/SYS** 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 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).

Consideration of Application note 15: Each audit record is automatically exported, so no additional action is required.

Application note 16: The automatic export shall prevent loss of internal audit data due to storage constraints, by protecting the audit data and storing the signed and timestamped data in the CTSS interface component, i.e. outside the TOE.

Consideration of Application note 16: The TOE automatically exports the data to the secure storage of the TSE.

6.1.6. Update Code Package

FDP_ACC.1/UCP Subset access control – Use of *Update Code Package*

Hierarchical to

FDP_ACC.1 Subset access control

Dependencies

FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/UCP

The TSF shall enforce the *update SFP* on

1. subjects: *CSP role*;
2. objects: *stored data*;
3. operations: *upgrade*

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 initialization

FDP_ACF.1.1/UCP

The TSF shall enforce the *Update SFP* to objects based on the following:

1. *subjects: CSP role*;
2. *objects: update code package with security attributes version number.*

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. *CSP role is allowed to upgrade the stored data if*
    a. *the digital signature of the UCP generated by the Issuer is successful verified by the SMAERS*

platform

FDP_ACF.1.3/UCP

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

FDP_ACF.1.4/UCP

The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

1. *a CSP role is not allowed to upgrade the stored data if the verification of digital signature of UCP by means of SMAERS Platform fails;*
2. [*none*]

Application note 17: The CSP role should be allowed to apply the stored *update code package* if the version number of the *update code package* is higher than the version number of the TSF. The execution of UCP is outside the TSF-mediated functionality of the PP on hand.

Consideration of Application note 17: The TOE only upgrades the version number if it is higher. Otherwise, the version number is not altered and the self test of the TOE fails. So if a platform would downgrade the TOE, the TOE enters its secure state and will not accept *transaction data* until the platform fixes the downgrade.

FDP_ETC.2/UCP_UD Export 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]

FDP_ETC.2.1/UCP_UD

The TSF shall enforce the *log message SFP* when exporting user data, controlled under the SFP(s), ~~outside of the TOE~~ **to the storage of the platform**.

FDP_ETC.2.2/UCP_UD

The TSF shall export the user data with the user data’s associated security attributes.

FDP_ETC.2.3/UCP_UD

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/UCP_UD

The TSF shall enforce the following rules when user data is exported from the TOE: [*none*]

FDP_ITC.2/UCP_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/UCP_UD

The TSF shall enforce the *update SFP* when importing user data, controlled under the SFP, from ~~outside of the TOE~~ **the storage of the platform**.

FDP_ITC.2.2/UCP_UD

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

FDP_ITC.2.3/UCP_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/UCP_UD

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_UD

The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [*none*]

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 successful upgrade of the stored data*** the following objects: *previous code and data*.

6.2. Security Assurance Requirements

The PP **[PP-SMAERS]** and this ST require the TOE to be evaluated according to EAL2 augmented with ALC_CMS.3 (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.

This chapter is equivalent to the corresponding chapter in **[PP-SMAERS]**, because no additional augmentation was added in this Security Target, which was not present in the Protection Profile.

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 and software platform and operating system) on which the TOE is designed to run shall be provided in addition.

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

This chapter is equivalent to the corresponding chapter in **[PP-SMAERS]**, because no additional SFRs were introduced in this Security Target, which were not already present in the Protection Profile.

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.

Table 6. Dependency rationale

SFR	Dependencies of the SFR	SFR components
FAU_GEN.1/SYS	FPT_STM.1 Reliable time stamps	FPT_STM.1 provided by the CSP PP Module Time Stamp Service and Audit
FAU_STG.1/SYS	FAU_GEN.1 Audit data generation	**FAU_GEN.1/SYS**
FAU_STG.3/SYS	FAU_STG.1 Protected audit trail storage	**FAU_STG.1/SYS**
FDP_ACC.1/LM	FDP_ACF.1 Security attribute based access control	**FDP_ACF.1/LM**
FDP_ACC.1/UCP	FDP_ACF.1 Security attribute based access control	**FDP_ACF.1/UCP**
FDP_ACF.1/LM	FDP_ACC.1 Subset access control	**FDP_ACC.1/LM**
	FMT_MSA.3 Static attribute initialization	**FMT_MSA.3**
FDP_ACF.1/UCP	FDP_ACC.1 Subset access control	**FDP_ACC.1/UCP**
	FMT_MSA.3 Static attribute initialization	**FMT_MSA.3**
FCS_COP.1/HashLastRoundOnCard	[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]	The SFR solely hashes data, so no key material is required.
	FCS_CKM.4 Cryptographic key destruction	The SFR solely hashes data, so no key material is required and deleted.
FDP_ETC.2/DTBS	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	**FDP_ACC.1/LM**
FDP_ETC.2/LM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	**FDP_ACC.1/LM**
FDP_ETC.2/UCP_UD	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	**FDP_ACC.1/UCP**

SFR	Dependencies of the SFR	SFR components
FDP_ITC.2/TD	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.1/LM
	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	Dependency on FTP_ITC.1 or FPT_TRP.1 is not fulfilled because secure import is ensured by OE.SecOEnv.
	FPT_TDC.1 Inter-TSF basic TSF data consistency	FPT_TDC.1
FDP_ITC.2/TSS	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.1/LM
	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	FTP_ITC.1/TC
	FPT_TDC.1 Inter-TSF basic TSF data consistency	FPT_TDC.1
FDP_ITC.2/UCP_UD	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.1/UCP
	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	FTP_ITC.1 is not included for UCP transfer but FDP_ACC.1/UCP ensure integrity and confidentiality of UCP
	FPT_TDC.1 Inter-TSF basic TSF data consistency	FPT_TDC.1 is not included because CSP uses the security attributes of UCP
FDP_RIP.1/UCP	No dependencies	
FIA_AFL.1	FIA_UAU.1 Timing of authentication	FIA_UAU.1
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

SFR	Dependencies of the SFR	SFR components
FCS_COP.1/HashPasswords	[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]	The SFR solely hashes data, so no key material is required.
	FCS_CKM.4 Cryptographic key destruction	The SFR solely hashes data, so no key material is required and deleted.
FMT_MOF.1	FMT_SMR.1 Security roles	**FMT_SMR.1**
	FMT_SMF.1 Specification of Management Functions	**FMT_SMF.1**
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	**FDP_ACC.1/LM** and **FDP_ACC.1/UCP**
	FMT_SMR.1 Security roles	**FMT_SMR.1**
	FMT_SMF.1 Specification of Management Functions	**FMT_SMF.1**
FMT_MSA.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	**FDP_ACC.1/LM**, **FDP_ACC.1/UCP**
	FMT_MSA.1 Management of security attributes	**FMT_MSA.1**
	FMT_SMR.1 Security roles	**FMT_SMR.1**
FMT_MSA.3	FMT_MSA.1 Management of security attributes	**FMT_MSA.1**
	FMT_SMR.1 Security roles	**FMT_SMR.1**
FMT_MSA.4	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	**FDP_ACC.1/LM**
FMT_MTD.1/AD	FMT_SMR.1 Security roles	**FMT_SMR.1**
	FMT_SMF.1 Specification of Management Functions	**FMT_SMF.1**
FMT_MTD.1/SYSCTSS	FMT_SMR.1 Security roles	**FMT_SMF.1**
	FMT_SMF.1 Specification of Management Functions	**FMT_SMR.1**
FMT_MTD.1/SYSAdmin	FMT_SMR.1 Security roles	**FMT_SMF.1**
	FMT_SMF.1 Specification of Management Functions	**FMT_SMR.1**
FMT_MTD.3/PW	FMT_MTD.1 Management of TSF data	**FMT_MTD.1/AD**
FMT_SMF.1	No dependencies	
FMT_SMR.1	FIA_UID.1 Timing of identification	**FIA_UID.1**
FPT_TDC.1	No dependencies	
FPT_FLS.1	No dependencies	

SFR	Dependencies of the SFR	SFR components
FPT_TEE.1	No dependencies	
FPT_TST.1	No dependencies	

6.3.2. Security functional requirements rationale

The tables trace each SFR in chapter 6.1 back to the security objectives for the TOE.

Table 7. Security functional requirements rationale

	O.GenLM	O.ImpExp	O.IAA	O.SecMan	O.TEE	O.TST	O.ImpExpUCP
FAU_GEN.1/SYS	x						
FAU_STG.1/SYS	x						
FAU_STG.3/SYS	x						
FDP_ACC.1/LM	x	x					
FDP_ACC.1/UCP						x	
FDP_ACF.1/LM	x	x					
FDP_ACF.1/UCP						x	
FCS_COP.1/HashLastRoundOnCard	x						
FDP_ETC.2/DTBS	x						
FDP_ETC.2/LM		x					
FDP_ITC.2/TSS	x						
FDP_ITC.2/TD	x	x					
FDP_ITC.2/UCP_UD						x	x
FDP_ETC.2/UCP_UD						x	x
FDP_RIP.1/UCP						x	
FIA_AFL.1			x				
FIA_ATD.1			x		x		
FIA_UAU.1					x		
FIA_UAU.5			x				
FIA_UAU.6			x				

	O.GenLM	O.ImpExp	O.IAA	O.SecMan	O.TEE	O.TST	O.ImpExpUCP
FIA_UID.1					x		
FIA_USB.1			x				
FCS_COP.1/HashPasswords			x				
FMT_MOF.1	x		x	x	x		
FMT_MSA.1	x			x	x		
FMT_MSA.2	x			x			
FMT_MSA.3	x			x			
FMT_MSA.4	x	x		x			
FMT_MTD.1/AD			x	x			
FMT_MTD.1/SYSCTSS	x						
FMT_MTD.1/SYSAdmin	x						
FMT_MTD.3/PW			x	x			
FMT_SMF.1	x	x		x			
FMT_SMR.1	x	x		x	x		
FPT_TDC.1	x	x					
FPT_FLS.1					x	x	
FPT_TEE.1					x	x	
FPT_TST.1						x	

The following part of the chapter demonstrate that the SFRs meet all security objectives for the TOE.

The security objective for the TOE **O.GenLM** *Generation of Log messages* is met by the following SFR:

- The SFR **FDP_ACC.1/LM** and **FDP_ACF.1/LM** require access control of import of TD and signatures, export of DTBS and log messages for roles defined by **FMT_SMR.1**.
- The SFR **FDP_ITC.2/TD** and **FDP_ITC.2/TSS** requires the TSF to import transaction data from CTSS interface component, audit records, time stamps, signature counter and signatures from CSP to generate log messages.
- The SFR **FDP_ETC.2/DTBS** requires the TSF to export data-to-be-signed to CSP for time stamping and signature generation.
- The SFR **FCS_COP.1/HashLastRoundOnCard** enables the TOE to compute a partial hash of the Data to be signed before they get exported to the CSP-L. Then the CSP-L is able to complete

the hash computation.

- The SFR **FMT_MSA.1** clause (3) prevents the manipulation of the *transaction number*.
- The SFR **FMT_MSA.2** ensures that the security attributes of a *log message* are generated in a way that the log message build valid transaction.
- The SFR **FMT_MSA.3** ensures restrictive security attributes of a *log message* as defined, and prevents alternative initial values of the security attributes of a log message.
- The SFR **FMT_MSA.4** describes the generation of security attributes which are included in a *log message*.
- The SFR **FMT_MOF.1** clauses (2), describes the behavior of FMT_MSA.4 for *keyId* in a log message.
- The SFR **FMT_MOF.1**, **FMT_MTD.3/PW**, **FMT_MSA.3**, **FMT_MSA.4** defined for SFR **FDP_ACC.1/LM** and **FDP_ACF.1/LM** are listed in SFR **FMT_SMF.1**.
- The SFR **FPT_TDC.1** ensures that the security attributes of imported *transaction data* and of the exported *log messages* are correctly interpreted.
- The SFR **FAU_GEN.1/SYS** , **FMT_MTD.1/SYSCTSS**, **FMT_MTD.1/SYSAdmin**, **FAU_STG.1/SYS**, **FAU_STG.3/SYS** decribes the generation and management of system logs.
- The security objective for the TOE **O.ImpExp** *Import of Transaction Data from and Export of Log message to CTSS Interface Component* is met by the following SFR:
- The SFR **FDP_ACC.1/LM** and **FDP_ACF.1/LM** require access control on the import of *transaction data*; and export of *log messages* to the CTSS interface component for roles defined by **FMT_SMR.1**.
- The SFR **FDP_ITC.2/TD** requires the TSF to import *transaction data* with security attributes in order to determine the security attributes of *log messages* according to **FMT_MSA.4**.
- The SFR **FDP_ETC.2/LM** requires the export of *log messages* with security attributes defined by **FMT_MSA.4** to the CTSS interface component for generation of receipts and verification of *log messages*.
- The SFR **FPT_TDC.1** ensures that the security attributes imported with transaction data and exported with *log messages* are correctly interpreted.

The security objective for the TOE **O.IAA** *Authentication of Administrators* is met by the following SFR:

- Administrator and CSP are requested to authenticate themselves according to **FIA_UAU.5**.
- The SFR **FIA_UAU.5** defines the authentication mechanisms supported by the TSF.
- The SFR **FMT_MOF.1.1**, clause (1) defines the rule that additional authentication (except for the administrator iTSElf) may be enabled and disabled by an administrator.

- The SFR **FIA_UAU.6** defines the condition for re-authentication.
- The SFR **FIA_AFL.1** defines required actions if password authentication fails.
- The SFR **FIA_ATD.1** defines the security attributes of users known to the TSF and the SFR FIA_USB.1 requires binding these security attributes to successfully authenticated users.
- The SFR **FMT_MTD.1/AD** and **FMT_MTD.3/PW** require the TSF to manage authentication data of users.
- The SFR **FCS_COP.1/HashPasswords** enables the TOE to store and compare authentication reference data without the need to store them in a recoverable form.

The security objective for the TOE **O.SecMan** *Security management* is met by the following SFRs:

- The SFR **FMT_SMR.1** defines the roles known to TSF and requires the TSF to associate users with these roles.
- The SFR **FMT_SMF.1** lists the management functions as management of functions **FMT_MOF.1**, management of TSF data **FMT_MTD.1/AD** and **FMT_MTD.3/PW**, and management of security attributes **FMT_MSA.1**, **FMT_MSA.2**, **FMT_MSA.3** and **FMT_MSA.4**.
- The SFR **FMT_MOF.1** restricts the ability to modify, enable, disable, determine the behaviour of and modify the behaviour of security functions to an administrator.
- The SFR **FMT_MTD.1/AD** and **FMT_MTD.3/PW** requires the TSF to manage authentication data of users.
- The SFR **FMT_MSA.1** and **FMT_MSA.3** describes the requirements for restrictive security attributes and limits the management of security attributes for the SFP *Log Message and Update*.
- The SFR **FMT_MSA.2** and **FMT_MSA.4** define requirements for the generation of security attributes of TDSs and TDSSs including the security attribute *time stamp*.
- The SFR **FMT_MSA.4** prevents management of the transaction numbers.

The security objective for the TOE **O.TEE** *Test of External Entities* is met directly by the SFR **FPT_TEE.1**.

The SFR **FMT_MOF.1**, clause (5), restricts the definition and modification of the **FPT_TEE.1** behaviour to the administrator. The **O.TEE** *Test of External Entities* is furthermore met by the following SFRs:

- The SFR **FMT_SMR.1** lists the roles known to the TSF, where subject CTSS interface component is automatically started and identified only.
- The SFR **FIA_UID.1** defines the self-test as the only TSF mediated action allowed before users

and subjects are identified.

- The SFR **FIA_UAU.1** defines the TSF mediated action allowed before users and subjects are authenticated. The subject CTSS interface component is allowed to perform automatically TSF mediated actions according to **FPT_TST.1** and **FPT_TEE.1** before users are authenticated.
- The SFR **FIA_ATD.1** defines the security attribute identity for the ERS and the CSP tested by **FPT_TEE.1**. If any test fails, the TSF enters a secure state according to **FPT_FLS.1**.

The security objective for the TOE **O.TST** *Self-test* is met by the following SFRs:

- The SFR **FPT_TST.1** requires the TSF to perform self-tests and **FPT_FLS.1** requires the TSF to enter a secure state if self-tests fails.
- The SFR **FPT_FLS.1** requires the TSF to enter a secure state if the self-test fails, the test of electronic record-keeping system fails, or the test of cryptographic service provider fails.
- The SFR **FPT_TEE.1** requires the TSF to enter the secure state according to **FPT_FLS.1** if testing of CTSS interface component or CSP fails.
- The SFR **FDP_ACC.1/UCP** and **FDP_ACF.1/UCP** requires the TSF to provide access control to enforce the update SFP. The SFR **FMT_MSA.1** prevents the modification of security attributes "version number" of the UCP.
- The SFR **FDP_RIP.1/UCP** requires the TSF to remove the received UCP after unsuccessful verification of its authenticity. The verification must be done by means of the platform

The security objective for the TOE **O.ImpExpUCP** *Secure Import and Export of User Data* is directly met by the SFR **FDP_ITC.2/UCP_UD** and **FDP_ETC.2/UCP_UD** that requires the TSF to export and import user data during an update process.

6.3.3. Security assurance requirements rationale

The EAL2 was chosen by **[PP-SMAERS]**, to which this Security Target conforms.

7. Package Trusted Channel between TOE and CSP

ST Application note 23: All SFRs apply independently to each SMAERS unit. I.e. an authenticated role applies only to one SMAERS unit, not to all. The same holds for the unit's state with respect to a secure state and so on.

The functional package for a trusted channel support between the TOE and the CSP is used by this Security Target as mandated by **[PP-SMAERS]**. The Security Objective **OE.SecCommCSP** has been replaced by the Security Objective **O.SecCommCSP** as mandated by the functional package.

This chapter contains the Security Functional Requirements that belong to this functional package.

The SFRs for cryptographic mechanisms based on elliptic curves refer to the following table for selection of curves, key sizes and standards.

Table 8. Elliptic curves, key sizes and standards

elliptic curve	key size	standard
brainpoolP256r1	256 bits	[RFC-5639], [BSI-TR-03111], section 4.1.3
brainpoolP384r1	384 bits	[RFC-5639], [BSI-TR-03111], section 4.1.3
brainpoolP512r1	512 bits	[RFC-5639], [BSI-TR-03111], section 4.1.3
Curve P-256	256 bits	[FIPS_186-4] B.4 and D.1.2.3
Curve P-384	384 bits	[FIPS_186-4] B.4 and D.1.2.4
Curve P-521	521 bits	[FIPS_186-4] B.4 and D.1.2.5

To perform mutual authentication using the PACE protocol, both endpoints need to share a static secret (PACE Password). The integrity and confidentiality of the shared secret have to be preserved by the TOE, using the secure storage of its platform.

Table 9. Additional assets in package Trusted Channel to be protected by the TOE

Asset	Protection
PACE password	integrity, confidentiality

7.1. Security Functional Requirements

7.1.1. Trusted Channel between TOE and CSP

FTP_ITC.1/TC Inter-TSF trusted channel

Hierarchical to

No other components.

Dependencies

No dependencies.

FTP_ITC.1.1/TC

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

FTP_ITC.1.2/TC

The TSF shall permit *the TSF* to initiate communication via the trusted channel.

FTP_ITC.1.3/TC

The TSF shall initiate communication via the trusted channel *for communication with the CSP*.

Application note 18: Protection against modification is required for the trusted channel. If sensitive data is transferred over the trusted channel, the ST writer shall provide additional cryptographic operations to protect the exchanged data against disclosure.

Consideration of Application note 18: The TOE implements a trusted channel with protection against modification.

FIA_UAU.5/TC Multiple authentication mechanisms

Hierarchical to

No other components.

Dependencies

No dependencies.

FIA_UAU.5.1/TC

The TSF shall provide

1. *PACE with Generic Mapping with user in ICC role with establishment of trusted channel according to FTP_ITC.1/TC*,

2. *[none]*
3. *message authentication by MAC verification of received messages* to support user authentication.

to support user authentication.

FIA_UAU.5.2/TC

The TSF shall authenticate any user's claimed identity according to ~~the~~

1. *PACE may be used for authentication of a CSP with establishment of trusted channel according to FTP_ITC.1/TC*,
2. *message authentication by MAC verification of received messages shall be used after initial authentication of a remote entity according to clause (1) for a trusted channel according to FTP_ITC.1/TC*.

Application note 19: The ST writer may assign another method of mutual authentication with key establishment in **FIA_UAU.5.1/TC** clause (2) if this method is supported by the certified CSP and therefore meets the OSP.SecCryM *Secure cryptographic mechanisms* in **[PP-CSP][PP-CSPLight]**.

Consideration of Application note 19: This ST does not contain another method of mutual authentication. The channel between TOE and CSP-L is secured using PACE as specified in clause (1). For this reason, the author assigned "none" to the open assignment in **FIA_UAU.5.1/TC** (2).

FIA_API.1 Authentication Proof of Identity – PACE authentication to Application Component

Hierarchical to

No other components.

Dependencies

No dependencies.

FIA_API.1.1

The TSF shall provide a *PACE in PCD role* to prove the identity of the *TOE* to ~~an external entity~~ **a CSP and establishing a trusted channel according to FTP_ITC.1/TC**.

FCS_CKM.1 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

The TSF shall generate cryptographic keys ***for FCS_COP.1*** in accordance with a specified cryptographic generation algorithm *PACE with [Curve P-256] and Generic Mapping in PCD role* and specified cryptographic key sizes *256 bits* that meet the following: ***[ICAO-Doc9303]****, section 4.4*

Application note 20: PACE is used to authenticate the TOE and the CSP. It establishes a trusted channel with MAC integrity protection of the following communication trough the trusted channel.

Consideration of Application note 20: The application note does not require any action in this ST, but is meant for clarification only.

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]
- FMT_MSA.2 Secure security attributes

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *[zeroization]* that meets the following: *[[FIPS_140-2] zeroization standards, chapter 4.7.6]*.

FCS_COP.1 Cryptographic Operation

Hierarchical to: No other components.

Dependencies

- [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
- FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1

The TSF shall perform *MAC calculation and MAC verification* in accordance with a specified cryptographic algorithm *according to AES-256 [FIPS_197] in [CMAC NIST SP 800-38B [NIST2005]]* and cryptographic key sizes *256 bits* that meet the following: *the referenced standards above according to the chosen selection.*

The following extended components are defined in **[PP-CSP][PP-CSPLight]** and are used here for the generation of ephemeral keys during the execution of PACE according to **FCS_CKM.1**.

FCS_RNG.1 Random Number Generation

Hierarchical to

No other components.

Dependencies

No dependencies.

FCS_RNG.1.1

The TSF shall provide a [*deterministic*] random number generator that implements:

1. [*(DRG.3.1) If initialized with a random seed [using a PTRNG of class PTG.2 as random source], the internal state of the RNG shall have [125 bit of entropy]*]
2. [*(DRG.3.2) The RNG provides forward secrecy*].
3. [*(DRG3.3) The RNG provides backward secrecy even if the current internal state is known*].

FCS_RNG.1.2

The TSF shall provide random numbers that meet

1. [*(DRG.3.4) The RNG, initialized with a random seed [of at least 125 bit], generates output for which [> 2^{14}] strings of bit length 128 are mutually different with probability [>1 - $2^{(-8)}$].*]

Application note 21: The TOE may use an internal source or an external source or more than one source of randomness providing seeds of at least 125 bits entropy. The deterministic part of the RNG shall meet **[BSI-TR-03116]** and must therefore be of class DRG.3 or higher according to **[AIS-20]**.

Consideration of Application note 21: The TOE implements a deterministic random number generator of class DRG.3, which gets seeded in the set up process of the TOE accordingly.

ST Application note 24: The choices of parameters in **FCS_RNG.1.1** were made in accordance with **[AIS-31]**. The random number generator is implemented according to **[NIST-800-90A]**, Chapter

10.1.1.

The dependencies are fulfilled:

Table 10. Dependency rationale for the functional package

SFR	Dependencies of the SFR	SFR components
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]	**FCS_COP.1**
	FCS_CKM.4 Cryptographic key destruction	**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**
FCS_COP.1	[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**
	FCS_CKM.4 Cryptographic key destruction	**FCS_CKM.4**
FCS_RNG.1	No dependencies	
FIA_API.1	No dependencies	
FIA_UAU.5/TC	No dependencies	
FTP_ITC.1/TC	No dependencies	

The security objective for the TOE **O.SecCommCSP** *Trusted channel between TOE and CSP* is implemented by the SFR:

- **FTP_ITC.1/TC** Inter-TSF trusted channel directly requiring the trusted channel between the TOE and the CSP protecting the integrity for their communication.
- **FIA_UAU.5/TC** requires the TSF to authentication the CSP as communication end point of the trusted channel.
- **FIA_API.1** requires the TSF to authentication themselves as communication end point of the trusted channel to the CSP.
- **FCS_CKM.1** requires the TSF to generate MAC keys for **FCS_COP.1**.
- **FCS_CKM.4** requires secure key destruction in order to fulfill the dependency of **FCS_CKM.1**.
- **FCS_COP.1** requires the TSF to calculate MAC for the own messages and to verify MAC for the CSP messages.

- **FCS_RNG.1** requires the TSF to implement a random number generator used for key generation according to **FCS_CKM.1**.

8. TOE Summary Specification

As stated above, all SFRs apply to each SMAERS unit independently. Therefor, this also applies to the functionality of this chapter.

8.1. SF.Log

After successful startup and self test, the SMAERS unit of the TOE allows the host / ERS to provide transaction data via commands, which the CTSS interface component forwards to the SMAERS unit of the TOE. Then the SMAERS unit of the TOE creates *transaction logs*. The *transaction logs* are partially signed using Hash Last Round on Card and the partial hash (i.e. the internal state of the hash function) and the last block of the hash are sent to the CSP-L, where the hash computation is finalized and the hash being signed by the CSP-L. The corresponding signed logs are exported to the CTSS interface component and stored on the secure storage of the TSE. To do so, each SMAERS unit of the TOE manages one *transaction counter* and keeps track, which transactions are open.

This way, the SFRs **FDP_ACC.1/LM**, **FDP_ACF.1/LM**, **FDP_ITC.2/TD**, **FDP_ETC.2/DTBS**, **FDP_ITC.2/TSS**,**FDP_ETC.2/LM**, **FPT_TDC.1**, and **FCS_COP.1/HashLastRoundOnCard** are implemented.

The TOE uses multiple keys for signature creation, each being associated with exactly one SMAERS unit. The dispatcher of the TOE ensures, that each incoming request is associated with the correct SMAERS unit and instantiates the unit in a worker thread. This makes sure, that the correct key is used. This implements **FMT_MSA.4.1** (1). Note that as stated in the beginning of this document, the *clientID* identifies SMAERS unit as well as ERS, so the dispatcher can identify which SMAERS unit to instantiate as well as the identify the ERS by considering the provided *clientID*.

Imported data are checked by the SMAERS unit of the TOE. Here the unit assures that ERS clientIDs are configured accordingly and if invoked operations match the internal state of the SMAERS unit of the TOE. This implements **FMT_MSA.4**. Data being imported from the CSP-L are also checked to implement **FMT_MSA.2.1**(2).

With respect to the formats of im- and exported data, the TOE is conformant to the specification in **[BSI-TR-03151]**, which implements **FPT_TDC.1.2**.

8.1.1. Transaction Counters

The transaction counters are managed by the SMAERS unit of the TOE. They are only changed, when a new transaction is opened and immediatly persisted in the corresponding SMAERS unit and the SMAERS unit of the TOE ensures that it will always be incremented by one when a new transaction is started. The worker thread concept ensures, that each time the correct transaction counter is taken into account, when needed.

This implements the SFRs **FMT_MSA.2.1**(1). To detect manipulations, the transaction counter is validated in the self test of **FPT_TST.1** against stored transaction logs, which were signed by the CSP.

8.2. SF.Crypto

The TOE implements cryptographic operations to establish a PACE channel with the CSP-L and a random number generator, being required for PACE. This implementation is used by each SMAERS unit on its own independently.

8.2.1. Random Number generation

Each SMAERS unit of the TOE implements a DRG.3 random number generator following the iterated hash example of **[AIS-31]** in Example 39 or **[NIST-800-90A]** accordingly. The random number generator is seeded during the setup of the TOE by entropy input acquired from a physical RNG which has the required properties. The seed has at least 125 bit entropy. This implements the SFR **FCS_RNG.1**.

8.2.2. PACE for secure channel with CSP

The TOE implements PACE to establish onee secure channel with one CSP-L. The channel is initiated from the SMAERS unit of the TOE to the CSP-L during the (self) test phase and successful channel creation is the first part of the CSP-L test. This way, all communication with the CSP-L is transported through a secure messaging channel, which was established using PACE. To execute PACE, a shared PACE-PIN of 256 bit is made use of. The the PIN gets stored within the SMAERS unit of the TOE at instantiation time of the unit. This unit dependent PIN cannot be changed in the TOE's life cycle. The PACE uses the elliptic curve *Curve P-256* and the resulting secure messaging channel AES-CMAC. The derived PACE keys are not stored persistently and kept in the RAM of the TOE hardware platform exclusively. They are overwritten with zeros, if possible, as soon as they are no longer needed to communicate with the CSP. In case of an unexpected power down (or comparable event) the key can not be overwritten with zeros.

This implements the SFRs **FTP_ITC.1/TC**, **FIA_UAU.5/TC**, **FIA_API.1**, **FCS_CKM.1**, **FCS_COP.1**, and **FCS_CKM.4**.

8.3. SF.Management

8.3.1. Role Management

The set of roles is fixed for the TOE and cannot be updated during the operation. Also all access rights, i.e. which role is able to execute which function are fixed, so there is no need for a flexible implementation of roles and their rights.

Instead the roles and their permissions are hardcoded in the TOE.

At execution of a command, triggered by the CTSS interface component, the TOE checks what roles the CTSS interface component currently has and whether the roles suffice to execute the command in question. To be able to do so, the TOE tracks which role the CTSS interface component currently has and has authenticated as. Note, that these checks happen within the SMAERS unit, i.e. a (SMAERS) administrator being logged in into SMASERS unit a can not execute adminstrative operations in SMAERS unit b. Even more the roles and credentials are per SMEARS unit, i.e. the (SMEARS) administrator of SMAERS unit b is a different one, than the logged in one (although both might be impersonated by the same human being).

Due to the TOE having multiple SMAERS units and using multiple keys, the TOE identifies the ERS, which have triggered the commands and instantiates a worker thread with the corresponding SMAERS unit to handle the ERS' request.

In addition, there is no interface to configure default values for security attributes, which implements **FMT_MSA.3.2**. By sticking to the provided default values from **[PP-SMAERS]**, restrictive choices were made to implement **FMT_MSA.3.1**

This way, the SFRs **FMT_SMR.1**, **FMT_SMF.1**, **FMT_MSA.1**, **FMT_MTD.1/AD**, and **FMT_MSA.3** are implemented.

To authenticate, *administrator* authenticates using a PIN. In addition, *administrator* has a PUK in case the PIN gets lost. The TOE offers a function to reset the PIN by the use of the PUK. The PIN and the the PUK have a a retry counter with an initial value of 3. PINs are permanently blocked after 3 failed retries. PUKs are blocked for (2 to the power of (failed retries -3)) seconds, i.e. 1 second before the 4th retry can be made, 2 secondes before the 5th, 4 seconds before the 6th, and so on.

The PINs and PUKs have to be of length at least 10 characters (each character is one byte, all possible values can be used). This is substantially stronger than the recommendation of **[BSI-TR-03147_Anforderungskatalog]** with level "HOCH".

The role *administrator* can change the Administrator PIN.

To store the authentication reference data, the TOE concats a salt of 128 bit length, taken from the TOE"s random number generator and hashes the concatenation with SHA256. This way the cleartext of

PINs and PUKs is not stored but can be compared efficiently.

Last but not least, *administrator* and *TR administrator* are automatically logged out after five minutes of inactivity and *Logger* is logged out after sixty minutes of inactivity.

This implements the SFRs **FIA_ATD.1**, **FMT_MTD.1/AD**, **FMT_MTD.3/PW**, **FIA_AFL.1**, **FIA_UAU.1**, **FIA_UAU.5**, **FIA_UAU.6**, and **FCS_COP.1/HashPasswords**.

8.3.2. Startup Process and self test

On startup the SMAERS unit of the TOE performs a set of tests. Prior to and while the tests are running, the CTSS Interface Component has the role *unidentified user*. Depending on the test results, the user has afterwards the roles *CSP* and/or *CTSS* and can then additionally authenticate as *administrator*. If the test fails, the SMAERS unit of the TOE enters a *secure state*.

This implements the SFRs **FIA_USB.1**, **FIA_UAU.6**, **FPT_TST.1**, and **FPT_TEE.1**.

As another part of this selftest, the SMAERS unit of the TOE compares the version of the code being executed against a stored reference version number. If the reference number is greater to the version of the code, the test fails and the SMAERS unit of the TOE enters a secure state. If the reference number is smaller, the TOE was updated. In that case it creates the corresponding *log messages* and updates the reference version number by overwriting it. This implements **FDP_RIP.1/UCP**, **FDP_ACC.1/UCP** and **FDP_ACF.1/UCP**. Also the required system logs are created, implementing **FAU_GEN.1/SYS** .

ST Application note 25: Note that the Update process itself is not handled within the TOE. Instead, the TSE operator has to stop the execution of the TOE, verifies and installs the new TOE version (i.e. replaces the TOE's jar file) and restarts the new binary. After restart, each SMAERS-Unit requires the execution of a selftest. In the course of this selftest each SMAERS-Unit on its own detects the Update of the TOE binary by comparing the version being hard coded in the binary with an expected version in the SMAERS-Units configuration data and creates the required log messages.

If the self test, test of ERS or CSP-L fails, the SMAERS unit of the TOE enters a secure state, which only allows to re-run the self test. No transaction data can be processed and the only operations being performable are a rerun of the test suite and configuration of the SMAERS unit (i.e. configure the ERS clientIDs, which is required to be done before the self test can succeed). This implements the SFRs **FPT_FLS.1**, **FIA_UID.1**, and **FMT_MOF.1** (5). Note that the self test can be initiated by the any user and has to be periodically executed 25 hours after the last invocation of the test suite, to use the SMAERS unit.

In addition entering and leaving the secure state creates the required system log, which implements **FAU_GEN.1/SYS** .

By only updating (or reading) user data on the storage of the platform, if the test was executed successfully, **FDP_ETC.2/UCP_UD** and **FDP_ITC.2/UCP_UD** are implemented.

By entering a secure state, which does not allow to manually export or clear logs, the TOE implements **FMT_MTD.1.1/SYSCTSS**

Initial Startup

At the initial startup, the TSE operator has set up SMAERS units. He also has to bring in the seed for the deterministic random number generator of the SMAERS unit of the TOE and PACE passwords to authenticate to the CSP-L. Besides, the the *clientIds* of the ERS(s) has/have to be configured which gets stored as a system log. Also the required system logs are created, implementing **FAU_GEN.1/SYS** .

8.3.3. Management of ERS clientIDs

To manage which ERS are accepted at startup and which *clientIds* can be used to start (update and finish) transactions, the SMAERS units of the TOE maintain lists of registered ERS *clientIDs* and associated SMAERS units.

This implements **FMT_MOF.1** (3) and **FMT_MSA.1**.

8.3.4. Terminating open transactions

The TOE does not terminate open transactions on its own. It requires the ERS to do so. In case the ERS is not aware which transactions are still open, the SMAERS unit of the TOE offers a function to retrieve a list of open transactions in the corresponding SMAERS unit. This way, the TOE does not have to make assumptions about the transactions or perform business decisions for the ERS.

Due to this behavior, there is no method to determine the life time limit of open transactions. This implements **FMT_MOF.1**, (2).

8.3.5. Other Management functions

The PACE-PIN is set in the instantiation of a SMEARS unit. This implements **FMT_MOF.1**(4).

By hardcoding the auditable events and automatically export each audit record, the TOE implements **FMT_MTD.1/SYSAdmin**

8.4. SF.Audit

The TOE fetches audit records from the CSP-L and stores them. In addition, it creates *system log* *messages* and also exports them to the CTSS Interface Compoent, which stores them in the secure storage of Swissbit Cloud-TSE 2 as required.

This is implemented according to **FDP_ITC.2/TSS** and **FDP_ETC.2/LM** to implement **FDP_ACF.1.2/LM** and **FDP_ITC.2.5/TD**.

The TOE has no interface to delete audit trails or log messages, which are not exported. This implements **FAU_STG.1/SYS**. In addition, all *system logs* are directly exported to the CTSS Interface component, implementing **FAU_STG.3/SYS**.

Related Documents

Note that version numbers of the following documents are partially missing. A separate, central list of references including document versions is created and provided to prevent version mismatches in different documents.

- [AIS-20] Evaluation of random Number Generators, BSI AIS 20, Version 3
- [AIS-31] A proposal for: Functionality classes for random number generators, BSI AIS 31, Version 2.0, September 2011
- [BSI-TR-03111] Technische Richtlinie BSI TR-03111 Elliptische-Kurven-Kryptographie (ECC), TR-03111, Version 2.10
- [BSI-TR-03116] Technische Richtlinie BSI TR-03116 Kryptographische Vorgaben für Projekte der Bundesregierung Teil 5: Anwendungen der Secure Element API, Datum: 1. Februar 2019
- [BSI-TR-03147_Anforderungskatalog] Anforderungskatalog zur Prüfung von Identifikationsverfahren gemäß TR-03147 in Version 1.0, Version 0.9, Dezember 2018
- [BSI-TR-03151] Technical Guideline BSI TR-03151 Secure Element API (SE API), TR-03151, Version 1.1.1
- [BSI-TR-03153] Technische Richtlinie BSI TR-03153 Technische Sicherheitseinrichtung für elektronische Aufzeichnungssysteme, TR-03153, Version 1.1.1
- [CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.
- [CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.
- [CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
- [FIPS_140-2] Security Requirements for Cryptographic Modules, FIPS 140-2, May 2001
- [FIPS_180-4] Secure Hash Standard (SHS), FIPS 180-4, October 2015
- [FIPS_186-4] Digital Signature Standard (DSS), FIPS 186-4, July 2013
- [FIPS_197] ADVANCED ENCRYPTION STANDARD (AES), FIPS 197, November 2001
- [ICAO-Doc9303] Machine Readable Travel Documents , ICAO, Doc 9303,Part 11: Security Mechanisms for MRTDSs, Seventh Edition, 2015
- [NIST2005] NIST Special Publication 800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication May 2005

- [NIST-800-90A] 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
- [PP-CSP] Common Criteria Protection Profile, Cryptographic Service Provider, Version 0.9.8
- [PP-CSPLight] Common Criteria Protection Profile Cryptographic Service Provider Light, BSI-CC-PP-0111-2019
- [PP-SMAERS] Common Criteria Protection, Profile Security Module Application for Electronic Record-keeping Systems (SMAERS), BSI-CC-PP-0105-V2-2020, version 1.0
- [PPC-CSP-TS-Au] Common Criteria Protection Profile Configuration, Cryptographic Service Provider - Time Stamp Service and Audit, Version 0.9.5
- [PPC-CSP-TS-Au-Cl] Common Criteria Protection Profile Configuration Cryptographic Service Provider – Time Stamp Service and Audit - Clustering, BSI-CC-PP-0108-2019
- [PPC-CSPLight-TS-Au-Cl] Common Criteria Protection Profile Configuration Cryptographic Service Provider Light – Time Stamp Service and Audit - Clustering, BSI-CC-PP-0113-2019
- [KSV] Verordnung zur Bestimmung der technischen Anforderungen an elektronische Aufzeichnungs- und Sicherungssysteme im Geschäftsverkehr,(Kassensicherungsverordnung – KassenSichV), Bundesgesetzblatt Jahrgang 2017 Teil I Nr. 66, ausgegeben zu Bonn am 6. Oktober 2017
- [FCG] Fiscal Code of Germany in the version promulgated on 1 October 2002 (Federal Law Gazette [Bundesgesetzblatt] I p. 3866; 2003 I p. 61), last amended by Article 6 of the Law of 18. July 2017 (Federal Law Gazette I p. 2745)
- [RFC-5639] Elliptic Curve Cryptography (ECC) Brainpool Standard, Curves and Curve Generation, March 2010
- [Umgebungsschutz] Swissbit Cloud CSP - Umgebungsschutzkonzept, Version TBD
- [SMAERS-AGD] Swissbit Cloud SMAERS - Guidance Manual, Version 1.0.5