swissbit®

Swissbit TSE SMAERS Firmware - Common Criteria Security Target

Version 1.9.7, 2020-11-26

Table of Contents

1. ST Introduction

The Fiscal Code of Germany [FCG] section146a requires that for an electronic record-keeping system, the accounts and the records must be protected by a certified technical security system. The Federal Office for Information Security defines requirements for the components of the certified technical security system, i. e. for the security module in form of Common Criteria Protection Profiles, and for the storage medium and the unified digital interface in form of Federal Office's technical guidelines (cf. [KSV] section 5). The security module consists of a controller, executing the security module application and the cryptographic service provider (CSP). This Security Target defines security requirements of the security module application. The security requirements for the CSP are defined in the Protection Profile Cryptographic Service Provider [PP-CSP].

1.1. ST Reference

- ST Reference: Swissbit TSE SMAERS Firmware - Common Criteria Security Target
- Sponsor: Swissbit
- ST Version: 1.9.7
- ST Date: 2020-11-26
- CC Version: 3.1 Revision 5
- Assurance Level: EAL 2
- Certification ID: BSI-DSZ-CC-1121-V2

1.2. TOE Reference

Table 1. TOE Reference

TOE Identifier	Hardware Version	Software Version
Swissbit TSE SMAERS Firmware USB	1.1.0	1.1.0
Swissbit TSE SMAERS Firmware SD	1.1.0	1.1.0
Swissbit TSE SMAERS Firmware microSD	1.1.0	1.1.0

It should be noted that the hardware version information that is listed in the table before, is relevant as the TOE comprises hardware aspects even though it is primarily a software TOE.

The TOE is delivered with the following additional documents:

Table 2. Delivery Items

Item	Version
Swissbit TSE - Guidance Manual [AGD]	1.4.1
Swissbit TSE - Integrator's Guidance Manual [AGD_integrator]	1.4.1
swissbit TSE - Functional Specification (ADV_FSP) [ADV_FSP]	1.3.0
swissbit TSE - Verpackungsprüfanweisung [Verpackungsprüfanweisung]	1.3.1
Swissbit TSE Data Sheets [DataSheet-SD], [DataSheet-MicroSD],and [DataSheet-USB]	1.1.0, 1.1.0, and 1.1.0
revocation passwords for certificate revocation	—
optional: host library for TSE communication	—
Fingerprint of the root CA	9604a459dd33b565c472cac07cdffe64734bea8c

Note that the Swissbit TSE Data Sheet exists for each configuration of the TOE. The documents Swissbit TSE SMAERS Firmware - Guidance Manual, swissbit TSE - Functional Specification (ADV_FSP), swissbit TSE - Verpackungsprüfanweisung and Swissbit TSE Data Sheets will be delivered electronically to customer, each time these get Swissbit TSEs delivered. This electronic delivery also contains the revocation passwords.

In addition, the host library is an optional component, which is not certified. It eases the communication with the TSE and therefore the TOE by wrapping the TOE's interface to an API. The library can be obtained from Swissbit.

1.3. TOE Overview

The TOE is named Swissbit TSE SMAERS Firmware and comprises the security relevant parts of the Swissbit TSE. As described in [PP-SMAERS] the TOE is a software TOE. However, due to technical constraints, it has been necessary to assign two aspects of the hardware of the Swissbit TSE to belong to the TOE as well. These are:

- The AES unit in the CPU of the controller of the TSE (which is needed for PACE) and
- the communication capabilities of the controller for USB or SD/microSD communication

The communication capabilities are needed for the primary functionality of the TOE. They form the primary interface to external entities. The primary functionality of the TOE (the provision of log messages) would not be possible without this part of the hardware.

All other hardware aspects (which specifically include the flash memory, the case of the Swissbit TSE and the rest of the controller) of the TSE, including the embedded CSP are not part of the TOE.

The TOE, together with a cryptographic service provider (CSP), is embedded into a token, which forms a certified technical security system (CTSS). The CTSS is also called Swissbit TSE and is the product that is actually sold to customers. The TOE creates cryptographically protected *Transaction Logs* of the financial transactions, which the Electronic Recordkeeping System (ERS) performs. To do so, the TOE uses the CSP, which is build into the CTSS, but is not part of the TOE.

The resulting *Transaction Logs* are stored in the flash memory of the Swissbit TSE, from which they can be exported. Note, that the flash memory is outside the TOE. The TOE manages the collection and processing of the ERS' data, the communication with the CSP and the storage and export of the *Transaction Logs*. To export the data, **[BSI-TR-03151]** specifies a data format and interface. The TOE implements this data format. Its components are shown in **Figure 1**. As visible in the picture, there is no component between the TOE and the ERS, brokering their communication. **[PP-SMAERS]** allows a *CTSS Interface Component* here, which is the external endpoint for the TOE's Data import and export functions. In our case, the endpoint of these operations is the ERS itself, so we consider the *CTSS Interface Component* to be part of the ERS.

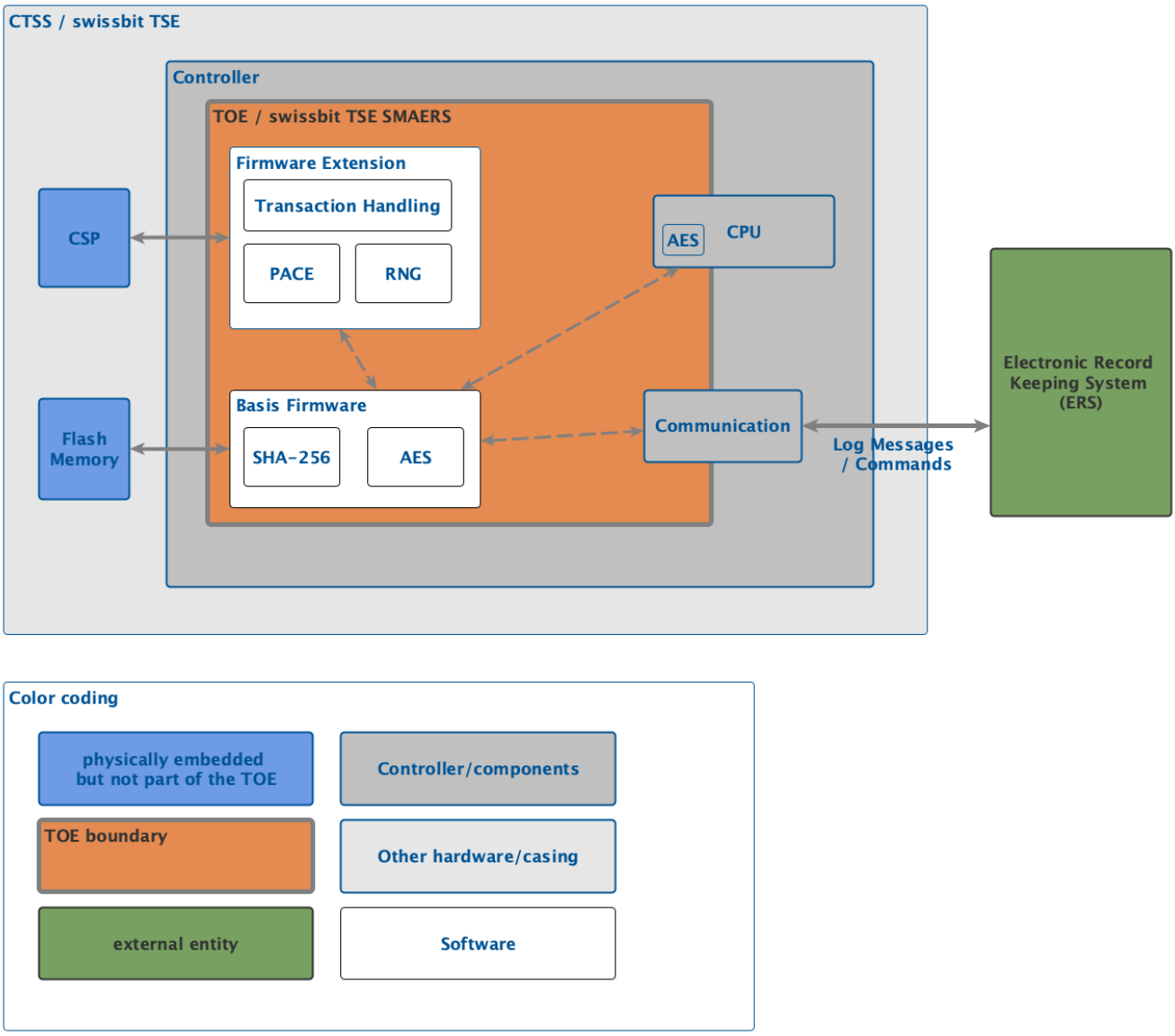

Figure 1. Overview of the components of the CTSS and the components within

The TOE, being executed on the controller of the Swissbit TSE and the CSP together are called the *security module* of the CTSS/token. The TOE implements 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

- tamper-proof determination of the point in time when a 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. [KSV] 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 accounts, records and security management activities in form of Log messages (cf. [BSI-TR-03153], chapter 3.1). The Log messages are created by the TOE using the services of the CSP.

Log messages comprise the certified data, the protocol data and the signature. There are three types of *Log messages*, i. e. *Transaction logs*, *System logs*, and *Audit logs*, cf. [BSI-TR-03153].

Transaction logs are created to protect the actual transaction data of the ERS as certified data. They will be created when a transaction is started, a transaction is finished (i. e. completed or terminated), and will be generated when transaction data is updated.

The protocol data of *Transaction logs* contains the transaction number of the actual transaction and time stamps. All *Transaction logs* with the same transaction number build together the transaction data defined in [KSV], section 2, sentence 2.

System logs are generated to document management or configuration operations of the security module. The certified data of the *Systems logs* provide information for interpretation of the transaction logs e. g. setting of the time source for the time stamps. The signature is generated for the certified data and the protocol data. It contains information about the signature algorithm and the signature value.

Audit logs are generated by the CSP, which the TOE converts to *System logs* and treats accordingly.

Overall, the TOE

- imports transaction data from the ERS as certified data of *Transaction logs*,
- generates part of the protocol data in the *Transaction log* including
    - the transaction number generated by the TSF,
    - the serial number as hash value of the public key included by the TSF for verification of the digital signature,
- includes to the *Transaction log* the digital signature created by the CSP over the certified data and the protocol data,
- imports audit records from the CSP (cf. [PPC-CSP-TS-Au], FAU_GEN.1) and exports them as system log,
- exports *Log messages* to the ERS,

- provides identification and authentication of users, access control and security management of the TSF for authorized users.

The signature counter enumerating the signatures created for *Log messages* and the time stamps when a signature was created are generated by the CSP and are part of the protocol data.

The TOE generates information about TSF security events as certified data of system logs exported to the CTSS interface component as specified in [BSI-TR-03151], Appendix A.

This TOE implements the Client-server architecture as described in [PP-SMAERS]. Therefore, this Security Target additionally uses the package Trusted Channel between the TOE and the CSP in chapter 7. The trusted channel is necessary because the TOE and the CSP are implemented as separated devices and shall interact through a trusted channel in order to protect the integrity of the communication data and to prevent misuse of the CSP signing and time stamping service provided for the TOE.

The TOE meets the BSI Technical Guidance [BSI-TR-03153] and uses cryptographic services of the CSP compliant with BSI TR-03116-5 [BSI-TR-03116].

1.3.1. Integration of the TOE in the Environment

The CTSS, which contains the TOE, CSP, flash memory, controller, and casing, bundled into one physical device, will be usually placed inside an Electronic Record Keeping System or attached to an external USB port of it. Figure 2 shows this setup.

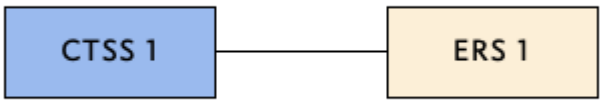

Figure 2. One CTSS directly connected to one ERS

Besides having the CTSS with the TOE placed next to or within the ERS, it is possible, to connect multiple CTSS to a *Hub*, which offers a REST-API. Now the ERS can connect via a local network to the *Hub* and use the TOE via the REST-API. To create a trusted channel, the *Hub* uses TLS 1.2 or 1.3, as specified by [RFC-5246] and [RFC-8446]. To provide a proper selection of TLS cipher suites, the choices which the *Hub* offers are restricted to the recommendations in [BSI-TR-02102-2].

In these scenarios there is no longer a one to one link between the CTSS and the ERS present, but one ERS can use multiple CTSS and multiple ERS can use one CTSS, depending on configuration. Still, each CTSS is associated with one Tax Identification Number / Tax Payer, which restricts the set of valid configurations (i.e. two ERS being associated with different Tax Payers can not share one CTSS).

In the first case, each ERS directly connects to the *Hub*. Then each ERS identifies itself with an API-Token inside the TLS channel for each call of a REST function. The *Hub* has a list of accepted API-Token and uses the API-Token to forward the ERS' calls to the correct CTSS.

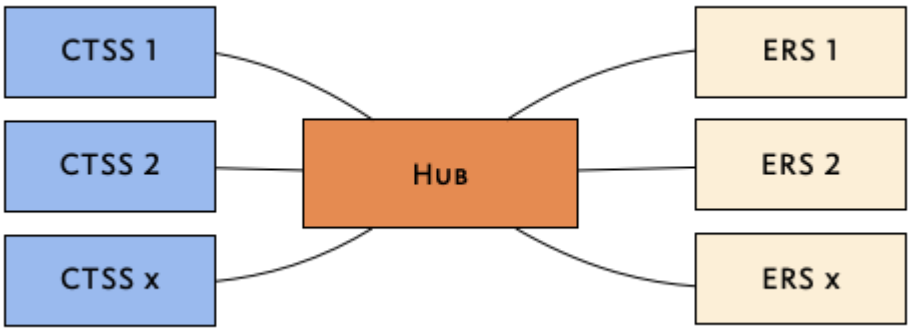

Figure 3. Multiple ERS connecting to multiple CTSS directly via a HUB

As a second case, it is also possible, that the ERS connect to a central ERS and communicate with the central ERS using a proprietary protocol. Here, the central ERS connects to the *Hub* using the REST-API, protected via TLS and provides the ERS' client ID via the REST-API. This is shown in **Figure 4**. Here the *Hub* decides which CTSS to forward the call to, based on the ClientID being provided with the REST function call.

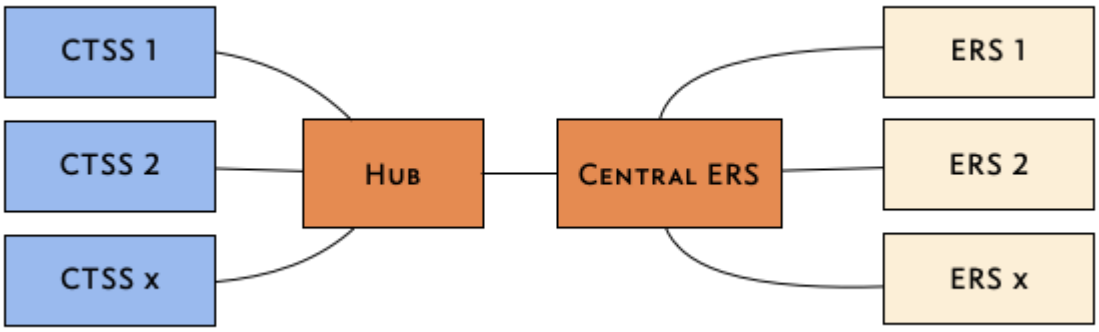

Figure 4. Multiple ERS connecting to multiple CTSS indirectly via a central ERS to the HUB

Note that this way of usage does not affect the Security Functions of the TOE itself, but affects the operational environment of the TOE. Swissbit, offering a *Hub* as sketched above, delivers the *Hub* with a manual, which contains clear instructions, how the used network connection has to be secured as well as clear and strict briefings about the operational environment of the *Hub* to maintain the physical security, which is required by **OE.SecOEnv**. Especially it is required to use an internal network and therefore the *Hub* and the TSEs have to be in the same store/location, which hosts also the ERS.

This different modes of integration of the CTSS with the TOE into the environment are not visible for

the CTSS (or TOE), though. They are used in the very same manner in each of the cases.

1.3.2. Usage and major security features

To use the CTSS with the TOE, the ERS communicates with it via a file interface. This means, the CTSS is visible as a file system to the ERS (in fact, it can also be inserted into a standard PC and will expose the file system) and status information can be read and commands sent by writing into special files. This file system is realized by the TOE in the CTSS.

This file system contains three (or more) special files:

- TSE_COMM.DAT
- TSE_INFO.DAT
- TSE_TAR.001
- optional: TSE_TAR.002, TSE_TAR.003, … if the amount of *Log messages* is too big to store in one file

The first one, TSE_COMM.DAT is used to execute API functions (authenticate, set time stamps, import *Transaction data*, or export filtered *Log messages*). The ERS writes the corresponding command into the file and the TOE executes it and writes the results into the file.

TSE_INFO.DAT contains status information of the TOE.

The last, TSE_TAR.00x contains a TAR-archive (compliant to **[BSI-TR-03151]**) of the saved *Log messages* (if the TOE's self test succeeded and the CTSS Interface Component was activated by the *Administrator*. Otherwise reading the file results in a string of 0x00).

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 inspection (cf. **[FCG]** section146b).

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* configures the communication channels between the TOE with the CTSS interface component and the CSP.

A second role *TimeAdmin* is used to update the CTSS' time stamp, which is stored in the CSP. Here the TOE is responsible to forward the corresponding information from the *TimeAdmin* to the CSP after authentication with a PIN.

The TOE supports receiving and integrity verification of *Update Code Packages (UCP)* for installation of a new TOE software version. To do so, the role *Administrator* can use the token's file interface to trigger an update command and provide the *Update Code Package* in small blocks, after which the CSP verifies and decrypts the package. If this is successful and the version number is higher than the current running one, the UCP gets installed.

1.3.3. TOE type

The Target of Evaluation (TOE) is a software type TOE, implementing the Client-server architecture from [PP-SMAERS]. While the TOE also comprises minor hardware parts, it can be categorized as a software type TOE as the prevailing parts of the TOE are build from software.

1.3.4. Required non TOE-Hardware/software/firmware

The TOE requires

- a CSP that is certified according to the Common Criteria Protection Profile Configuration Cryptographic Service Provider – Time Stamp Service and Audit [PPC-CSP-TS-Au] ,
- a flash memory for storage of information,
- a case, and
- the capabilities of the controller for execution of the TOE.

All these components are part of the CTSS, which Swissbit produces and sells as the product.

The CSP shall further meet [BSI-TR-03116]. The CSP shall export audit records in form of system logs meeting [BSI-TR-03151].

1.4. TOE Description

1.4.1. Introduction

The TOE is part of a CTSS to enable cash registers to process and store fiscal data following the requirements of the Fiscal Code of Germany [FCG] and [BSI-TR-03153]. The CTSS exists in different configurations, which differentiate by the tokens physical form-factor: micro SD-Card, SD-Card or USB-Token.

The three form-factors of the CTSS are shown exemplified in Figure 5. They form all configurations of the TOE. Their differences are solely based on the physical form factor and a different controller. This means the hardware, which the TOE is executed on differs, depending on the configuration. This is required to implement the different external interface (USB or SD). Accordingly, their Basis Firmware

differs, to realize the file system via the different interfaces. Still, the usage of all TOE configurations is the same, as well as the software, implementing the TOE's business logic. Also,the same certified CSP is used internally.

Table 2 lists all configurations of TOEs.

Table 3. TOE configuration overview

Configuration name	Form factor
Swissbit TSE SMAERS Firmware USB	USB
Swissbit TSE SMAERS Firmware SD	SD-Card
Swissbit TSE SMAERS Firmware microSD	micro SD-Card

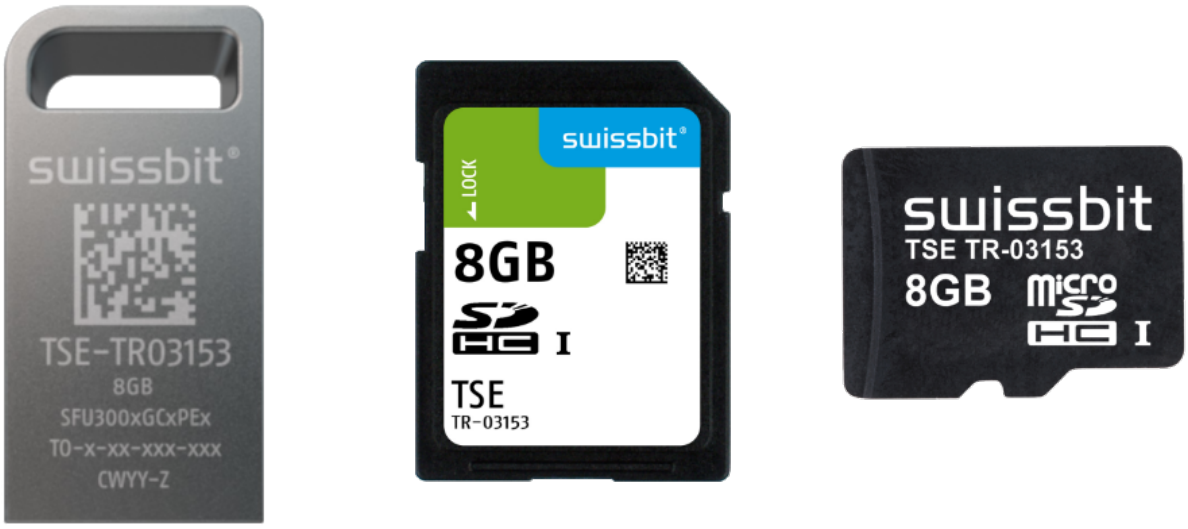

Figure 5. Form Factors of Swissbit TSEs, each containing one configuration of the TOE

The TOE is part of the CTSS Swissbit TSE, which is meant to be installed in a cash register (Electronic Record Keeping System, ERS) and receives the data via the file system, which the (micro) SD-Card or USB interface offer. The same interface is later used to export the corresponding processed fiscal data.

1.4.2. Architecture

The Swissbit TSE contains Flash Memory and the CSP beside the TOE. The token (i.e. the Swissbit TSE) also has a housing, which depends on the token's configuration and either forms a USB-token, a SD-Card or a micro SD-Card.

Note that the Flash Memory is used to store the TOE's data (configuration data, status data, authentication reference data and so on) as well as the *Log messages*.

The controller executes a basis-firmware and a firmware-extension, which implements the TOE's business logic. Both, basis-firmware and firmware-extension together form the TOE's software. In addition, the token contains a CSP, certified according to **[PPC-CSP-TS-Au]**. The controller is internally connected to the CSP via a Serial Interface. This CSP is not part of the TOE. This architecture of TOE is shown in **Figure 1**.

While the TOE is a software TOE, it has been necessary to assign two dedicated parts of the hardware of the Swissbit TSE to the TOE. Namely, the AES unit and the communication component of the controller. Both hardware parts are needed to realize security relevant functionality.

The basis firmware of the TOE offers the file system to the ERS and manages the persistently stored data on the flash memory. In addition, it contains a representation of the AES implementation and the implementation of the hash algorithm to hash data to be signed prior to signature creation by the CSP. The AES implementation is required to establish a trusted channel with the CSP and to additionally encrypt software updates.

The firmware extension contains the TOE's business logic, i.e. the roles and permissions, the communication with the CSP, including the PACE and random number generator implementation. Here, most of the SFRs are implemented. The business logic in the firmware extension includes the required functionality of the *CTSS interface component* from **[PP-SMAERS]**, which allows ERS to directly communicate and use the TOE. There is no need for an additional wrapper in form of an external *CTSS interface component*.

All components are wrapped in one of the three form-factors, listed above, which communicate with each other internally. The functionality, required by **[PP-SMAERS]** and **[BSI-TR-03153]** is implemented in the extension of the controller's firmware. Therefore the TOE uses the 'client-server architecture' of **[PP-SMAERS]**. Here the firmware extension manages (open) transactions, including the transaction counter.

It should be highlighted that the CSP which is packaged together with the TOE into one case and which is used by the TOE, is not part of the TOE. The CSP has been certified according to **[PP-CSP]**. The functionality of the CSP, its production, delivery, the required PKI and key handling procedures are not addressed during this certification.

For users (mainly cash registers, administrators and fiscal controllers), the TOE provides a file system that is formatted as FAT16/32 as specified in **[FAT32]**. Some of the space is pre-occupied to reserve space for the *Log messages*. In a special file, users place data, which get processed by the firmware extensions on the controller (which then communicates with the CSP if required). Afterwards, files with the resulting values can be found in the file system.

1.4.3. TOE boundaries

physical boundaries

As a software type TOE, the TOE has no physical boundary. The minor hardware parts of the TOE have a physical boundary in form of the interface of the controller that implements the AES and the communication capabilities that are part of the TOE.

logical boundaries

The logical boundaries of the TOE are formed by the interface, which can be accessed by the host / ERS via the file system, which the token presents to the host system. Using special files, the host is able to send commands and retrieve their results. The range of commands allows the host

- to authenticate,
- start, update and finish transaction,
- manage the internal settings of the TOE
- export the log messages and
- updating the TOE's and CSP's software.

The TOE has a second interface, which TOE and CSP use to communicate with each other. This interface does not implement TSFIs.

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 does claim strict conformance to [PP-SMAERS].

2.3. Package Claim

This ST claims to be compliant to the package *Trusted Channel between TOE and CSP* as defined in [PP-SMAERS].

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 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, modifications and gaps shall be detectable,
- the audit records imported from the CSP and exported to the CTSS interface component.
- the Update Code Package (UCP) imported and verified as user data.

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

Users and subjects

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

- *Electronic record-keeping system (ERS)*,
- *CTSS interface component*,
- *CSP* as sender of audit records,
- *Administrator*.

The ERS is tested by the TOE as external entity and communicating with the TOE through the file system interface. The TOE stores *Log messages* in its Flash Memory. The TOE uses the CSP also as external entity providing security services (i. e. the CSP is passive communicating with the TOE).

The subjects as active entities in the TOE perform operations on objects and obtaining their associated security attributes from the authenticated users on behalf they are acting, or by default.

Objects

The TSF operates 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 message* (LM) as *Transaction log* or *System log*,
- *Update Code Package* (UCP).

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* containing

- the clientID with the Identity of the CTSS interface device,
- the processData with
  - the *Transaction Type*,
  - the *Transaction Data*,
  - the *Monetary Type of Transaction*,

  - the *Serial number of ERS*

- the *Type of the Operation* as *StartTransaction*, *UpdateTransaction* or *FinishTransaction* provided by the command sent by the CTSS interface component to the TOE.

Audit records are data imported from CSP or may be generated by the TSF about TSF security events. 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 *Transaction log*: the *Transaction Data* with 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-03151]**, chapter 2.3.1)
  - in case of *System log*: the *Audit Record* with 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
- protocol data generated by the TSF
  - the *Transaction Number*,
  - the *Serial Number* as 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 *Time*, when the Log message is created,
- the *Signature counter* enumerating the signatures created with the signature-creation key.

The *Log message* consists of the

- the *Log message tag* and *Version of the Log message format*,
- the certified data,
- the protocol data,
- the Signature consisting of the identifier of the signature algorithm, parameters as defined by the signature algorithm and the signature value (cf. **[BSI-TR-03153]**).

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

The *UCP* are user data which are imported by the TOE for installation of a new TOE software version.

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

The user uses authentication verification data to prove its identity to the TOE. The TSF uses Authentication Reference Data to verify the claimed identity of a user. The TSF supports human user authentication by knowledge where the authentication verification data is a password and the authentication reference data is a password or an image of the password e. g. a salted hash value.

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

- *Unidentified User role*: 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 disabled *CTSS interface component*. The TOE allows user in this role to run self-test of the TOE. Note that the role *Administrator* is entitled to disable (and enable) the *CTSS interface component*.
- *Administrator role*: 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. The Administrator is allowed to activate and to deactivate the role CTSS interface.
- *CTSS interface role*: A subject in this role is allowed to import Transaction Data from CTSS interface component, to generate Transaction logs, and to export Transaction logs to the CTSS interface component. 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 device and the CSP are successfully tested according to **FPT_TEE.1**.
- *CSP role*: A subject in this role is allowed to import audit records from CSP and to export System logs to the CTSS interface component. 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**.

ST Application Note 1: The TOE has a dedicated administrator model, which consists of two roles: *Administrator* and *TimeAdmin*. The role *TimeAdmin* authenticates with a PIN and is entitled to import time information, which the TOE forwards to the CSP to adjust the CSP's internal clock.

The *Transaction Data* have the security attributes

- *Serial number of the ERS* to determine the signature-creation key to be used for signing the *Transaction log* and the *Serial number* 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.

The TOE accepts *Transaction Data* only if the serial number of the ERS is known, a signature key in the CSP and the *Serial number* is assigned to this ERS.

ST Application Note 2: The TOE / CSP supports the usage of use only one signature creation key, which simplifies the assignment of ERS to keys.

If the Type of the Operation is *StartTransaction* or *FinishTransaction* the TOE generates a *Transaction log* for the imported *Transaction Data*. If the *Type of the Operation* is *UpdateTransaction* the TOE may collect the imported *Transaction Data* and include them immediately or later on in one and only one *Transaction log* (cf. **[BSI-TR-03151]**).

ST Application Note 3: This TOE does not collect multiple updates to sign and store them together. Each *UpdateTransaction* always immediately results in a corresponding signed *Log message*.

The TOE manages for each known ERS a list of the last assigned transaction number and the transaction numbers of the ongoing transactions of this ERS. 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.

The *Log messages* have the security attributes in the protocol data and the signature used by the verifier

of the cash inspection

- *Transaction number* assigning the Log message to the transaction of the electronic record-keeping system.
- *Signature counter* enumerating the Log message continuously increasing without gaps,
- *Time stamp* as time when the Log message was created,
- *Type of the Operation* to determine whether the Log message was created for the start, update and finishing the transaction of the electronic record-keeping system,
- *Serial number* to determine the certificate to be used for verification of the digital signatures as check value of the transaction data.

The verifier of the cash inspection should interpret the Log message to determine a transaction **[KSV]** section 2 sentence 2 as follows:

- number 1: the point in time when the transaction starts is the *Time stamp* of the *Log message* with the *Type of the Operation* equal to *StartTransaction* and the transaction number identified as number 2.
- number 2: the transaction number is the *Transaction number* in the protocol data of the *Log message*.
- number 3 the transaction type, number 4 the transaction data and number 5 the monetary type of transaction are contained in the certified data of all *Log messages* with the transaction number identified as number 2.
- number 6: the point in time when the transaction is completed or terminated is the *Time stamp* of the *Log message* with *Type of the Operation* equal to *FinishTransaction* and the *Transaction number* identified as number 2.
- number 7: the check value is a set of signatures in the protocol data of all *Log messages* with the same Transaction number identified as number 2.
- number 8: the serial number of the security module generated for the transaction is contained in the protocol data of the *Log messages*.

The 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.

The UCP may have a version number.

ST Application Note 4: UCPs always have a version number.

3.2. Threats

T.EvadTD: Evading *Transaction Data*

The attacker evades 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 CSP. A forged *Transaction log* may result in forged transaction data provided for cash inspection. A forged *system log* may result in faulty interpretation of the transaction data.

T.ManipLM: Manipulation of a *Log message*

The attacker manipulates undetected a *Log message* exported to the CTSS interface component and used for cash inspection.

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

The attacker manipulates undetected the *Log message sequence* exported to the CTSS interface component and used for cash inspection.

T.ManipTN: Manipulation of *Transaction Number*

The attacker manipulates the TOE internal *Transaction Number* used in *Log messages*.

T.FaUpD: Faulty *Update Code Package*

An unauthorized entity provides an unauthorized faulty *Update Code Package* enabling attacks against integrity of TSF implementation, confidentiality and integrity of user data or TSF data after installation of the faulty *Update Code Package*.

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 on 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 the time stamps, when the transaction starts and when the transaction is completed or terminated, and the transaction number (cf. [KSV] section 2 sentence 3).The security module of the certified security device shall be certified according to Federal Office's Common Criteria Protection Profiles.

OSP.ProtDev: Protection of electronic record-keeping system and certified security device

The taxpayer shall use correctly the electronic record-keeping system (cf. [FCG] section 379 (1) sentence 1 number 4), and protect correctly 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 in encrypted form and signed by the authorized issuer. The TOE verifies the authenticity of the received *Update Code Package* using the CSP before storing in the TOE.

3.4. Assumptions

A.CSP: Cryptographic service provider

The operational environment provides a cryptographic service provider certified according to a Security Target compliant the Common Criteria Protection Profile Configuration Cryptographic Service Provider – Time Stamp Service and Audit [PPC-CSP-TS-Au]. The CSP exports audit records in form of system logs meeting [BSI-TR-03151].

A.ProtComCSP: Protection of communication between TOE and CSP

The operational environment protects the integrity of communication data between the TOE and the CSP. In case of 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.

Application Note 1: The main part of the Protection Profile in hand assumes the TOE being implemented as software running on the CSP as secure execution platform (cf. Platform architecture [PP-CSP]). In case of the Client-server architecture (cf. [PP-CSP]) the Security Target shall claim additionally the package Trusted Channel between the TOE and the CSP in chapter 7. If the security module follows the client-server architecture, the CSP is assumed to use the trusted channel provided by the TOE.

Consideration of Application Note 1: This TOE is implemented in Client-server architecture, so chapter 7 is implemented by the TOE and part of this Security Target.

A.ProtComERS: Protection of communication between TOE and electronic record-keeping system

The electronic record-keeping system provides transaction data when the transaction starts, transaction data are updated, and the transaction is completed or terminated. The operational environment protects the integrity of communication data between the TOE and the electronic record-keeping system.

A.VerifLMS: Verification of Log message Sequences

The operational environment verifies the digital signatures, the transaction numbers and the time stamps of the Log messages in the sequence in order to detect forged or missing Log messages. The certificate of the signature-verification data is securely distributed to the verifier.

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.

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.

O.ImpExp: Import of *Transaction Data* from and Export of *Log message* 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 CSP and export *Log messages* to the CTSS interface component.

O.IAA: Identification of external entities and authentication of Administrators

The TOE shall identify and test the external entities electronic record-keeping system and cryptographic service provider, and verify the claimed identity of the Administrators by means of password.

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 on electronic record-keeping system and cryptographic service provider connected to the TOE, allow generation of *Log messages* only if both pass the tests, and 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, the test of electronic record-keeping system fails, or the test of cryptographic service provider fails.

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

The TSF shall verify the authenticity of received encrypted *Update Code Package* and decipher authentic *Update Code Package* by means of the cryptographic service provider before it stores the *Update Code Package*. The TOE shall allow only authenticated Administrators to install *Update Code Package* for creation of a new security module application.

O.SecCommCSP Trusted 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 5: 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 use correctly an electronic record-keeping system that provides separately, correctly, completely and in real time all *Transaction Data* that are legally required for generation of *Log messages* to the TOE. The electronic record-keeping system shall support its testing as external entity by the TOE. The electronic record-keeping system shall produce receipt including 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 (i.e. the CSP).

OE.CSP: Cryptographic service provider component

The operational environment shall provide a cryptographic service provider for the TOE that is certified as compliant with Common Criteria Protection Profile Configuration Cryptographic Service Provider – Time Stamp Service and Audit **[PPC-CSP-TS-Au]**. The CSP shall export audit records in form of system logs meeting **[BSI-TR-03151]**.

Application Note 2: The Common Criteria Protection Profile Configuration Cryptographic Service Provider – Time Stamp Service and Audit **[PPC-CSP-TS-Au]** requires the cryptographic service provider to provide security services for digital signing of *Transaction Data*, verification of signature of *Update Code Packages*, decryption of *Update Code Packages*, and time service. The CSP audit records shall be exported meeting **[BSI-TR-03151]** in order to avoid transformation of the 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 2: The TOE of this Security Target is physically bundled with a cryptographic service provider. In addition, the TOE exports the CSP's audit records and stores them accordingly.

OE.CSPPlatform: CSP as 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 3: In case of client-server architecture the TOE and the CSP are physically separated components and the TOE does not need the CSP as secure execution platform.

Consideration of Application Note 3: The TOE of this Security Target uses the client-server architecture and does not use the CSP as secure execution platform.

OE.Transaction: Verification of Transaction

The operational environment shall verify the validity of *Log message Sequences* by verification of

the digital signatures, the *Transaction Numbers* as being consecutive without gaps, the points in time when the transaction starts as being consecutive 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 that is linked to the taxpayer. The certificate shall be securely distributed to the verifier.

OE.SecOEnv: Secure operational environment

The operational environment shall protect the electronic record-keeping system and the certified technical security system including the TOE against manipulation, perturbation and misuse. It protects the integrity of the communication between the electronic record-keeping system and the TOE.

Application Note 4: The main part of the Protection Profile in hand assumes the TOE being implemented as software running on the CSP as secure execution platform (cf. Platform architecture [PP-CSP]). In case of the Client-server architecture (cf. [PP-CSP]) the Security Target shall claim additionally the package Trusted Channel between the TOE and the CSP in chapter 7. If the security module follows the client-server architecture, i. e. the TOE and the CSP are physically separated components and the operational environment cannot ensure the integrity of the communication between the TOE and the CSP, the TOE shall support trusted channel functionality between the TOE and the CSP. The usage of the trusted channel is a specific form how the operational environment meets OE.SecCommCSP.

Consideration of Application Note 4: This Security Target follows the client-server architecture and takes chapter 7 into account accordingly. For this reason, OE.SecCommCSP has been removed from the Security Target and replaced by an objective for the TOE (O.SecCommCSP). It should be noted that the corresponding assumption A.ProtComCSP is not longer required but has not been removed by the ST author as the functional package from the [PP-SMAERS] did not contain any instructions to do so.

OE.SUCP: Signed Update Code Packages

The issuer shall issue encrypted and digital signed secure *Update Code Packages* together with its security attributes.

4.3. Security objectives rationale

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

Table 4. Security objective rationale

	T.Evad TD	T.Manip TD	T.Manip DTBS	T.Manip LM	T.Manip LMS	T.Manip TN	T.FaUpD	OSP.SecERS	OSP.CertSecDev	OSP.ProtDev	OSP.ValidTrans	OSP.Update	A.CSP	A.ProtComCSP	A.ProtComERS	A.VerifLMS
O.GenLM	x			x	x						x					
O.IAA											x					
O.ImpExp					x						x					
O.SecMan						x					x					
O.SecUCP							x					x				
O.TEE	x	x	x	x	x			x								
O.TST				x												
OE.CSP				x					x				x			
OE.ERS	x	x						x								
O.SecCommCSP			x													
OE.SecOEnv	x	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 Log Messages" is mitigated by:

- The security objective for the TOE **O.GenLM** requiring the TSF to *Transaction logs* containing *Transaction Data, Transaction Number* generated by the TSF, and time stamps and digital signatures, therefore allowing to decide whether presented TD have corresponding TDS in the TDSS
- The security objective for the TOE **O.TEE** requiring the TSF to test on 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 electronic record-keeping system, the TOE and the communication between them 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 on 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.SecOEnv** requiring the operational environment to protect the electronic record-keeping system and 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 on CSP connected to the TOE.
- The security objective for the operational environment **OE.SecOEnv** "Secure operational environment" protecting the CSP and the certified technical security system including the TOE against manipulation, perturbation and misuse.
- The security objective **O.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 TOE shall protect the integrity of the communication

between the TOE and the cryptographic service provider. 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]** 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 signature generated by CSP, which allows to detect manipulation of TDS according to **OE.Transaction**.
- The security objective for the TOE **O.TEE** "Test of external entities" requiring the TSF to test on CSP connected to the TOE.
- The security objective for the TOE **O.TST** "Self-test and secure state" detects failure and prevents generation of TDS if time source is not available or the test of CSP fails.
- The security objectives for the operational environment **OE.CSP** "Cryptographic service provider component" ensures the availability of certified CSP for generation of time stamps and digital signatures, and 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 CSP and the TOE against manipulation, perturbation and misuse of signature-creation service.

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, TSF time stamps when the transaction starts, is completed or aborted, TSF *Transaction Number* and a digital signature of the *Transaction Data* created using the digital signature-creation service of 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 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 on 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 CSP and the TOE against manipulation, perturbation and misuse of

signature-creation service.

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

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

- The security objectives for the TOE **O.SecUCP** "Secure download and authorized use of Update Code Package" ensuring that only authentic *Update Code Packages* are stored and installed by authorized Administrators only.
- The security objective for the operational environment **OE.SUCP** ensures that the authentic *Update Code Packages* are signed and distributed with security attributes.

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 ERS as 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 CSP and the TOE against manipulation, perturbation and misuse of signature-creation service

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 conform to the Protection Profile in hand.

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, TSF time stamps when the transaction starts, is completed or aborted, TSF Transaction Number and a digital signature of the *Transaction Data* created using the digital signature-creation service of cryptographic service provider,
- the security objectives for the TOE **O.IAA** "Identification of external entities and authentication of Administrators" requiring the TSF to authenticate the Administrators by means of 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 interface component and 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 TDS.

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 digital signature of secure Update Code Packages together with its security attributes and the security objectives for the TOE **O.SecUCP** "Secure download and authorized use of Update Code Package" ensuring verification of digital signature.

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.ProtComCSP** "Protection of communication between TOE and CSP" mainly refers to TOEs implementing the platform architecture, as stated in Application Note 1.

This TOE implements client-server architecture, so 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]** and by the TOE claiming the package Trusted Channel between the TOE and the CSP, cf. chapter 7. This way, the TOE protects the integrity of the communication.

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.

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".

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 denoted by the word "refinement" in **bold** text and the added/changed words are in **bold** text, or 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].

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*,
- [*and TimeAdmin*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

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 Serial Numbers]*

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 functions 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,*
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.*

Application Note 5: The refinements of **FMT_MOF.1**, bullet (2) to (5) 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 *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 "Serial number of ERS" to Administrator,*
2. ***define depending on the Serial number of ERS the identity of the signature-creation key to be used for the Transaction log to Administrator,***
3. ***define depending on the Serial number of ERS the Serial number in the protocol data of Transaction log to Administrator,***

4. ***define the identity of the signature-creation key to be used for the System logs and the Serial number in the protocol data of System logs to Administrator,***
5. ***increase by 1 the internally stored security attribute "Transaction Number" when transaction is started to subjects in CTSS interface role,***
6. ***modify the TD security attribute "Transaction Number" imported from the TD to none,***
7. ***modify the security attributes of UCP to none.***

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.

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~~ [**refinement: Administrator and TimeAdmin**]:

1. *Identity,*
2. *Authentication Reference Data,*
3. *Role*

and

a. **security attribute *Identity [and SerialNumber]* belonging to the ERS**
b. **security attribute *Identity [and PACE-PIN]* 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 by user by Administrator 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 6: 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, TimeAdmin 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 *[block the corresponding reference data for further use. In case of Administrator PIN or TimeAdmin PIN, the reference data can be reset using the PUK, which resets the number of failed authentication tries for this credential to 0. A blocked PUK can not be reset].*

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****.*
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*~~ *[****refinement: Administrator role or TimeAdmin role****] after successful authentication.*
4. *The Administrator is allowed to activate and deactivate the CTSS interface role.*

ST Application Note 7: **FIA_USB.1.3** (3) associates the subject with role *Administrator* after successful authentication with the Administrator Pin and with role *TimeAdmin* after successful authentication with the TimeAdmin Pin.

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 if testing was successful and the role CTSS interface 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 to reset the Administrator PIN, TimeAdmin PIN, or PUK, if correct PUK value is provided*
5. *Allow unidentified User to read TOE Serial Number, Software Version, some status information of the TOE and extended Card Life Time Information.*]

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

ST Application Note 8: The Card Life Time Information mentioned in **FIA_UAU.1.1** consists of Read disturb management enable status; Global wear level status; Global remap status; Host transfer CRC errors;Total LBAs read; Total LBAs written; ECC correction capability; Card Life Time Information as in SD Status register; Total number of sectors read from flash; Number of uncorrectable ECC errors during startup; Number of correctable ECC errors during startup; Minimum block erase count;Maximum block erase count;Anchor block write count;Initial read disturb threshold;Current read disturb threshold; RDM Block refresh count; Extended number of correctable ECC errors; Warm reboot count; Commit count; Flush count; Firmware update count; Total number of read retries; Total number of read retries during startup; Number of kept uncorrectable ECC read errors; Protect status; Total number of sectors written to flash;

These data are used to analyze defect TOEs and help the ERS Systems to foresee upcoming problems of TOEs, such that these can suggest to replace the TOE in time. The read information are low level information giving insight in the usage and possible problems of the flash memory of the TOE.

The additional Status Information are data, which can be read in the file TSE_Info.dat. These contain the capacity of the TOE, the TOE's initialization Status, if the initial PINs were already changed and if the TOE passed the last self test.

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 Administrator* [
2. *password authentication shall be used for TimeAdmin*
3. *successful PACE-Channel establishment shall be used for CSP role*
4. *provision of valid ERS Serial Number shall be used for CTSS Interface role*]

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*

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;*
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.*

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**,*
    b. *export the DTBS of Transaction log to the CSP according to **FDP_ETC.2/DTBS**,*
    c. *import the protocolData with signature from the CSP according to **FDP_ITC.2/TSS**,*
    d. *export the Transaction log to the CTSS interface component according to **FDP_ETC.2/LM**.*
2. *A subject in activated CTSS interface role 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 System logs to the CTSS interface component according to **FDP_ETC.2/LM**.*

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*
2. *a subject in activated CTSS interface role and CSP role and (Administrator role or TimeAdmin role) is allowed to import time Stamps from ERS and export them to the CSP.*
3. *a subject in Administrator role is allowed to clear the list of transactions, if it was exported successfully*

]

FDP_ACF.1.4/LM

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

1. *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. *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 9: 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 10: To set up the TOE, the TOE requires the *Administrator* to provide an ERS Serial Number 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 Serial Number of the ERS if the Serial Number of the ERS is in the set of accepted values according to **FMT_MSA.1**. If the Serial Number of the ERS 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 CSP.*

Application Note 8: If the TOE is used by more than one taxpayer than each taxpayer shall use its own signature key identified by the serial numbers of ERS.

Consideration of Application Note 8: This TOE only supports one taxpayer and one signature key in the CSP. Therefore matching of taxpayer and signature key is trivially given.

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 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]***.

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~~ **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~~ **CSP** :

1. [*none*]

Application Note 9: 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]**). Note, the TOE of the protection profile in hand may use CSP providing time stamps by administrator settable internal clock (sf. 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 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 9: 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 are used to test the CSP, according to **FPT_TEE.1**. The roles *Administrator* and *TimeAdmin* are able to update the CSP's internal clock.

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 ERS 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]** enumerating all Log messages,*
    c. *Type of the Operation,*
    d. *Time stamp when the Log message was signed,*
    e. *Serial Number 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. *Audit records of the CSP shall be exported unchanged as system logs to the CTSS interface component.*

Application Note 10: 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. The ERS uses the TDS fields 1,2, 6 and 8 for creation of receipts only. The TDS data fields number 1, 2, 6, 7 and 8 are used as security attributes of Log messages by the verifier of transactions for cash inspection.

Consideration of Application Note 10: The CTSS interface component is part of the TOE. So no component outside of the TOE implements SFRs from **[PP-SMAERS]**.

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. *Serial Number of the ERS,*
2. *Type of the Operation,*
3. *Transaction Number,*
4. *Signature Counter,*
5. *Time stamp,*
6. *Serial Number 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 security attributes

1. *Transaction Numbers building a strong increasing sequence without gaps,*

2. *Time stamps of the Log messages building a not decreasing sequence with consideration of adjustments of the CSP time source.*

Application Note 11: 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 11: 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 Serial Number of the ERS imported with Transaction Data to determine the signature-creation key be used by FDP_DAU.2/TS with ECDSA in **[PPC-CSP-TS-Au]** 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 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 12: 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 fail. The exit of the secure state requires all conditions listed in the refinement being fulfilled.

Consideration of Application Note 12: The TOE only exists 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 [ERS Serial Number] and*
2. *CSP Identity [PACE PIN, signature counter, and time stamp].*

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** [no additional action]*.

Application Note 13: The Administrator may by able to define the actions in **FPT_TEE.1** according to **FMT_MOF.1.1** (5). E. g. the test of the ERS may include the interface used by the ERS for

communication with the CTSS as reported by the CTSS interface component. The suite of tests determine whether the configured CSP is available for the TOE and Log messages can be signed. The TOE may use signature counter and time stamps received from CSP to test the CSP. 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]. Please refer for further explanations to the user notes and evaluator notes in CC part 2 [CC2], chapter J.12.

Consideration of Application Note 13: To test the ERS, the ERS has to provide its serial number. This means, the interface between ERS and TOE is used to test the ERS. The test of the CSP 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***.

6.1.5. Code Update Package import

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: *Administrator*;
2. objects: *Update Code Package*;
3. operations: *import, decrypt*

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: Administrator*;
2. *objects: Update Code Package with security attributes Issuer and Signature.*

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. *Administrator is allowed to import and store received Update Code Package if*
    a. *the digital signature of the UCP generated by the Issuer is successful verified by the CSP and*
    b. *the verified UCP is deciphered by means of CSP.*

FDP_ACF.1.3/UCP

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

1. [*none*]

FDP_ACF.1.4/UCP

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

1. *Administrator is not allowed to import received Update Code Package if verification of digital signature by means of CSP fails;*
2. [*Administrator is not allowed to install received Update Code Package if decipherment of UCP by CSP fails;*]

Application Note 14: The Administrator should be allowed to execute the stored Update Code Package if the version number of the Update Code Package is equal or 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 14: If the version of the UCP is equal or higher then the version of the corresponding installed version, the UCP gets installed and executed after a reboot of the TOE. Otherwise, the UCP will not be installed or made use of.

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

Hierarchical to

No other components.

Dependencies

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

FDP_ITC.2.1/UCP

The TSF shall enforce the *Update SFP* when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/UCP

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

FDP_ITC.2.3/UCP

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

FDP_ITC.2.4/UCP

The TSF shall ensure that interpretation of the security attributes of the imported user data is as

intended by the source of the user data.

FDP_ITC.2.5/UCP

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

1. *storing of encrypted Update Code Package only after successful verification by means of CSP*,
2. *decrypts authentic Update Code Package by means of CSP*.

FDP_RIP.1/UCP Subset residual information protection:

Hierarchical to

No other components

Dependencies

No dependencies.

FDP_RIP.1.1/UCP

The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource* ***after unsuccessful verification of the digital signature of the issuer by means of CSP*** the following objects: *received Update Code Package*.

6.2. 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.

6.2.1. Dependency rationale

This chapter demonstrates that each dependency of the security requirements is either satisfied, or justifies the dependency not being satisfied.

Table 5. Dependency rationale

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

6.2.2. Security functional requirements rationale

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

Table 6. Security functional requirements rationale

	O.GenLM	O.ImpExp	O.IAA	O.SecMan	O.TEE	O.TST	O.SecUCP
FDP_ACC.1/LM	x	x					
FDP_ACC.1/UCP							x
FDP_ACF.1/LM	x	x					
FDP_ACF.1/UCP							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							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				
FIA_UID.1			x				
FIA_USB.1			x				
FMT_MOF.1	x		x	x	x		
FMT_MSA.1	x			x			x

	O.GenLM	O.ImpExp	O.IAA	O.SecMan	O.TEE	O.TST	O.SecUCP
FMT_MSA.2	x			x			
FMT_MSA.3	x			x			x
FMT_MSA.4	x	x		x			
FMT_MTD.1/AD			x	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 FMT_MSA.1 clauses (4) prevents the manipulation of the *Transaction Number*.
- The SFR FMT_MSA.2 ensures that the security attributes of the *Log message* are generated in a

way that the Log message build valid transaction.

- The SFR **FMT_MSA.3** ensures restrictive security attributes of *Log message* as defined and prevent alternative initial values of the security attributes of Log message.
- The SFR **FMT_MSA.4** describes the generation of security attributes which are included in the *Log message*.
- The SFR **FMT_MOF.1** clauses (2), describes the behavior of FMT_MSA.4 for *Serial Number* in the 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 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 import of Transaction Data; and export of Log messages to CTSS interface component for roles defined by **FMT_SMR.1**.
- The SFR **FDP_ITC.2/TD** requires the TSF to import the 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 export of Log messages with security attributes defined by **FMT_MSA.4** to CTSS interface component for generation of receipts and verification of Log messages.
- The SFR **FPT_TDC.1** ensures that the security attributes of the imported *Transaction Data* and of the exported *Log messages* are correctly interpreted.

The security objective for the TOE **O.IAA** "Identification of external entities and authentication of Administrators" is met by the following SFR:

- The SFR **FMT_SMR.1** lists the roles known to the TSF, where subject CTSS interface component is automatically started and identified only, and Administrator and CSP are requested to authenticated themselves according to FIA_UAU.5.
- The SFR **FIA_UID.1** defines self-test as the only TSF mediated action allowed before user and subjects are identified.
- The SFR **FIA_UAU.1** defines the TSF mediated action allowed before user 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_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 the Administrator.
- The SFR **FIA_UAU.6** defines the condition for re-authentication.
- The SFR **FIA_AFL.1** defines action if password authentication fails.
- The SFR **FIA_ATD.1** defines the security attributes of users known to TSF and the SFR **FIA_USB.1** require binding of these security attributes to successful authenticated users.
- The SFR **FMT_MTD.1/AD** and FMT_MTD.2/PW require the TSF to manage authentication data of users.

ST Application Note 11: Here the Protection Profile refers to FMT_MTD.2/PW, which the ST authors kept. The BSI informed the ST authors so far, that the SFR that should be referenced is **FMT_MTD.3/PW** instead.

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

- 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 behavior of and modify the behavior of security functions to Administrator.
- The SFR **FMT_MTD.1/AD** and FMT_MTD.2/PW require 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 generation security attributes of TDS and TDSS including the security attributes time stamps.
- The SFR **FMT_MSA.4** prevents management of the *Transaction Numbers*.

ST Application Note 12: Here the Protection Profile refers to FMT_MTD.2/PW, which the ST authors kept. The BSI informed the ST authors so far, that the SFR that should be referenced is **FMT_MTD.3/PW** instead.

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 SFR **FIA_ATD.1** defines the security attribute *Identity* for ESR and 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 SFR:

- 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 security objective for the TOE **O.SecUCP** "Secure download and authorized use of *Update Code Package*" is met by the following SFR:

- The SFR **FDP_ACC.1/UCP** and **FDP_ACF.1/UCP** requires the TSF to provide access control to enforce SFP *Update*. Note the verification of the authenticity of UCP and decryption of authentic UCP are performed by CSP under control of the TSF. The SFR **FMT_MSA.1** prevents the modification of security attributes of UCP.
- The SFR **FDP_ITC.2/UCP** requires the TSF to import UCP as user data with security attributes if the authenticity of UCP is successful verified.
- The SFR **FMT_MSA.3** requires to provide restrictive initial security attributes to enforce the SFP *Update*.
- The SFR **FDP_RIP.1/UCP** requires the TSF to remove the received UCP after unsuccessful verification of its authenticity by means of CSP.

6.2.3. 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

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 7. 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

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~~ **[using physical separated ports]** 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*.

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.

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 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 remote entity according to clause (1) for trusted channel according to* ***FTP_ITC.1/TC***.

Application Note 15: 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]**.

Consideration of Application Note 15: This ST does not contain another method of mutual authentication. The channel between TOE and CSP 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~~ **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 [brainpoolP256r1] and Generic Mapping in PCD role* and specified cryptographic key sizes *256 bits* that meet the following: ***[ICAO-Doc9303]***, *section 4.4*

Application Note 16: 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 16: 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.*

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 - $2^{(-8)}$].*]

2. [*(DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A [and an online test according to section 5.5 of [AIS-31]]*]

Application Note 17: The TOE is defined as as software running on the CSP platform (referred as Platform architecture in [PP-CSP]) or as device (referred as Client-server architecture in [PP-CSP]). The TOE may use internal source or 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 TR3116-5 [BSI-TR-03116] and therefore of class DRG.3 or higher according to [AIS-20].

Consideration of Application Note 17: The TOE uses client-server architecture. It uses the CSP for seeding.

ST Application Note 13: 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 8. 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

8.1. SF.Log

After successful boot and self test, the TOE allows the host / ERS to provide transaction data via commands, which the TOE uses to create Transaction logs, being signed by the CSP. The corresponding signed logs are returned to the host and stored within the TOE. To do so, the TOE manages a 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**, and **FPT_TDC.1** are implemented.

In addition, the TOE uses only one key for signature creation in the CSP, which makes the association of client-id / ERS serial number to the key to use easier and implements **FMT_MSA.4.1** (1) in the trivial way.

Imported data are checked by the TOE, if ERS serial numbers are configured accordingly and if invoked operations match the internal state of the TOE. This implements **FMT_MSA.4**. Data being imported from the CSP 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**.

Note that *Administrator* is also allowed to clear the sequence of *Log messages*, if the ERS confirmed to

have received them.

8.1.1. Transaction Counter

The transaction counter is managed by the TOE. It is stored at two positions in memory to harden manipulation of it. It is only changed, when a new transaction is opened and the TOE ensures that it will always be incremented by one when a new transaction is started.

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 and a random number generator, being required for PACE. In addition, the TOE encrypts incoming *Update Code Packages* to make them unavailable, if they fail to get verified by the CSP.

8.2.1. Random Number generation

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 by entropy input acquired from the CSP with a resulting seed of 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 a secure channel with the CSP. The channel is initiated from the TOE to the CSP during the boot/ (self) test phase and successful channel creation is the first part of the CSP test. This way, all communication with the CSP 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 PIN gets stored within the TOE and CSP at production time. This device dependent PIN cannot be changed in the TOE's life cycle. The PACE uses the elliptic curve brainpoolP256r1 and the resulting secure messaging channel AES-CMAC. The derived PACE keys are not stored persistently and kept in the TOE's RAM 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**.

8.2.3. Encryption of incoming Update Code Packages

Update Code Packages get an additional encryption layer by the TOE to implement **FDP_RIP.1/UCP**, i.e. to make them unavailable, if the CSP fails to verify them. To do so, the incoming chunks of the package are additionally encrypted using AES-256. The incoming package was already encrypted and signed by the issuer This means, here is an additional, outer layer of encryption added. The corresponding key for this layer is generated using the random number generator from **FCS_RNG.1** and kept in RAM only. The encrypted *UCP* is stored in a non-host-readable area of the flash memory. When the complete package is present, it gets read from the flash, decrypted and sent to the CSP piece by piece. This way it can be ensured, that the *UCP* is not stored in a decrypted way on the flash memory before it gets verified by the CSP. If the verification via the CSP fails, the temporary AES-key gets deleted (overwritten with zeros) and the handler to the encrypted *UCP* package gets freed. Otherwise, the decrypted parts of the *UCP* get sent again to the CSP to decipher the inner encryption layer, which was added by the UCP's issuer. Afterwards, the verified and deciphered *UCP* gets stored on the flash memory and can get installed, if the version number is higher than the one of the current installed version.

Since the temporary AES-key is in RAM only, it is easier and cheaper to delete than to overwrite the stored *UCP*, which is unavailable after the AES-key got deleted.

This implements **FDP_RIP.1/UCP**

8.3. SF.Management

8.3.1. Updating CSP Time Stamp

Based on the (physical) architecture, only the TOE is able to directly communicate with the CSP. Therefore, the TOE implements a method to receive external time stamps and forwards them to the CSP, as described in **[BSI-TR-03151]**. To do so, the host system has to authenticate with the role TimeAdmin and is then able to set an updated time, which the TOE forwards to the CSP. This is implemented by the SFR **FDP_ACF.1.3/LM**.

8.3.2. 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 host system, the TOE checks, what roles the host

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 current host system has and has authenticated as.

Since the TOE only offers one interface and can not distinguish between different entities using this interface, it has always exactly one user. Note, that roles are implemented additively, i.e. the host system can be authenticated as more than one role at a time and has correspondingly the union of the permissions of all of its roles.

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, *Admin* and *TimeAdmin* authenticate 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. If *TimeAdmin* looses the PIN, *Administrator* is able to reset it. Both PINs and the the PUK have a a retry counter with an initial value of 3. The PINs have to be of length 5, while the PUK has to have a length of 6, following the recommendation of **[BSI-TR-03147_Anforderungskatalog]** with level "Substantiell" for the PINs and level "Hoch" for the PUK.

The role *Administrator* can change the Administrator PIN, the role *TimeAdmin* can change the TimeAdmin PIN. In addition, *Administrator* can change the Administrator PIN, TimeAdmin PIN and PUK using the PUK as credential.

The initial PINs are derived from the TOEs serial number and stored at production time. They have to be changed after the first login. The file TSE_INFO.DAT in the file system indicates, if the PINs were already changed.

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

8.3.3. Startup Process and self test

On Power, the TOE boots and performs a set of tests. Prior to and while the tests are running, the host / user has the role *unidentified user*. Depending on the test results and if the *CTSS role* was (de-) activated by *Administrator*, the user has afterwards the roles *CSP* and/or *CTSS* and can then additionally authenticate as *Admin* and/or *TimeAdmin*. If the test fails, the TOE enters a *secure state*.

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

If the self test, test of ERS or CSP fails, 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 TOE through the role *Administrator* (i.e. configure the ERS Serial Numbers, 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 *Administrator* and is periodically executed 25 hours after the last invocation of the test suite.

Initial Startup

When the TOE is first started, it requires the user to change PUK, TimeAdmin PIN and Administrator PIN. In addition, the Client ID(s) of the ERS(s) has/have to be configured which gets stored as a system log.

8.3.4. Management of ERS Serial Numbers

To manage, which ERS are accepted at startup and which client ids can be used to start (update and finish) transactions, the TOE maintains a list of registered ERS Serial Numbers. The *Administrator* is allowed to manage this list.

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

8.3.5. Terminating open transactions

The TOE does not terminate open transactions. It requires the ERS to do so. In case the ERS is not aware, which transactions are still open, the TOE offers a function to retrieve a list of open transactions. 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).

8.3.6. Other Management functions

The TOE and the CSP are physically coupled together at production time. Then, the PACE-PIN is set for both components of the security module. This implements **FMT_MOF.1**(4).

8.3.7. Updating firmware and firmware extension

The TOE receives updates of its firmware and firmware extensions via the file interface. Here the host, if authenticated as role Admin, is able to use a command, which allows to import the firmware updates into the firmware extension. Then the TOE lets the CSP first verify the update packages and -if successful- lets the CSP decrypt them. Afterwards the processed update packages are stored in a special non-host-accessible memory, from which the firmware takes and installs them. If the verification or

decryption of the update packages fails, they will be made unavailable by additional temporary encryption and deletion of the encryption key from the memory. Details of this can be found in the SF.Crypto section of this chapter.

This way, the SFRs **FDP_ACC.1/UCP**, **FDP_ACF.1/UCP**, **FDP_ITC.2/UCP**, and **FDP_RIP.1/UCP** are implemented.

8.4. SF.Audit

The TOE fetches audit records from the CSP and stores them. In addition, it creates *System log messages* and also stores them in the flash memory as required. These logs can be exported in the same way as the *Transaction logs* or the filtered export can be used to export non-*Transaction logs* only.

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**.

9. Related 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-02102-1] Technische Richtlinie BSI TR-02102-1 Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Version 2019-1
- [BSI-TR-02102-2] Technische Richtlinie TR-02102-2 Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 2 – Verwendung von Transport Layer Security (TLS), TR-02102-2, Version 2019-01
- [BSI-TR-03111] Technische Richtlinie BSI TR-03111 Elliptische-Kurven-Kryptographie (ECC), TR-03111, Version 2.10
- [BSI-TR-03145] Technische Richtlinie BSI TR-03145 Secure Certification Authority operation, TR-03145, Version 1.1
- [BSI-TR-03153] Technische Richtlinie BSI TR-03153 Technische Sicherheitseinrichtung für elektronische Aufzeichnungssysteme, TR-03153, Version 1.0.1
- [BSI-TR-03151] Technical Guideline BSI TR-03151 Secure Element API (SE API), TR-03151, Version 1.0.1
- [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
- [CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.
- [CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.
- [CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
- [CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2017-04-004, Version 3.1 Revision 5, April 2017.
- DEBASAFE Sicherheitstaschen - **https://www.debatin.de/wp-content/uploads/sites/4/2020/06/DEBASAFE-Sicherheitstaschen_dt.pdf**
- [DSFinV-K] Digitale Schnittstelle der Finanzverwaltung für Kassensysteme (DSFinV-K 2.2), Version 2.2, **https://www.bzst.de/DE/Unternehmen/Aussenpruefungen/DigitaleSchnittstelleFinV/digitaleschnittstellefinv_node.html**
- [FAT32] Microsoft Extensible Firmware Initiative FAT32 File System Specification, Version 1.03, December 2000
- [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
- [ISO-18033-3] ISO/IEC 18033-3 Information technology - Security techniques, Encryption algorithms - Part 3: Block ciphers, 2010
- [ISO-IEC-7816-3] ISO/IEC 7816-3 Identification cards - Integrated circuit cards - Part 3: Cards with contacts — Electrical interface and transmission protocols, 2006
- [NIST2005] NIST Special Publication 800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication May 2005
- [NIST2007] NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November, 2007
- [NIST2008] FIPS PUB 198-1 The Keyed-Hash Message Authentication Code (HMAC), July 2008

- [NIST2010] NIST Special Publication 800-38A Recommendation for Block Cipher Modes of Operation: Methods and Techniques, October 2010
- [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-SMAERS] Common Criteria Protection, Profile Security Module Application for Electronic Record-keeping Systems (SMAERS), Aktuell in Version 0.7.5
- [PP-CSP] Common Criteria Protection Profile, Cryptographic Service Provider, Version 0.9.8
- [PPC-CSP-TS-Au] Common Criteria Protection Profile Configuration, Cryptographic Service Provider - Time Stamp Service and Audit, Version 0.9.5
- [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-2986] PKCS #10: Certification Request Syntax Specification, Version 1.7, November 2000
- [RFC-5246] The Transport Layer Security (TLS) Protocol Version 1.2, August 2008
- [RFC-5280] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008
- [RFC-5639] Elliptic Curve Cryptography (ECC) Brainpool Standard, Curves and Curve Generation, March 2010
- [RFC-6816] Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, January 2013
- [RFC-8446] The Transport Layer Security (TLS) Protocol Version 1.3, August 2018
- [SD-Spec] SD Specifications, Part 1, Physical Layer - simplified Specification, Version 5.00, August 2010
- [SP-800-38E] Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices, SP 800-38E, January 2010
- [TAR-FORMAT] The Open Group: POSIX.1-1988 -Portable Operating System Interface, 1988
- [USB-Spec] Universal Serial Bus Specification , Revision 2.0, April 2000
- [CSP-User-Guide] TCOS CSP Security Module, Version: 1.0.1d7, Date: 28.06.2019

- [RNG-testvectors] https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Standards-and-Guidelines/documents/examples/Hash_DRBG.pdf
- [TCOS-Guidance] TCOS CSP Security Module - User's Guidance Manual, Version 1.0.4
- [ISO7816-3] ISO/IEC 7816-3:2006 IDENTIFICATION CARDS — INTEGRATED CIRCUIT CARDS — PART 3: CARDS WITH CONTACTS — ELECTRICAL INTERFACE AND TRANSMISSION PROTOCOLS
- [ISO2187] ISO/IEC 21827:2008 Information technology — Security techniques — Systems Security Engineering — Capability Maturity Model(SSE-CMM)
- [ST-TSE] Swissbit TSE SMAERS Firmware - Common Criteria Security Target, Version 1.9.7
- [ADV_FSP] Swissbit TSE - Functional Specification (ADV_FSP), Version 1.3.0
- [AGD] Swissbit TSE - Guidance Manual, Version 1.4.1
- [AGD_integrator] Swissbit TSE - Integrator's Guidance Manual, Version 1.4.1
- [Verpackungsprüfanweisung] Swissbit TSE - Verpackungsprüfanweisung, Version 1.3.1
- [DataSheet-SD] Product Data Sheet Swissbit SD TSE, Version 1.1.0
- [DataSheet-MicroSD] Product Data Sheet Swissbit microSD TSE, Version 1.1.0
- [DataSheet-USB] Product Data Sheet Swissbit USB TSE, Version 1.1.0