Bundesamt
für Sicherheit in der
Informationstechnik

Common Criteria Protection Profile

Cryptographic Modules, Security Level “Low”

Common Criteria

BSI-CC-PP-0044

Endorsed by the
Bundesamt für Sicherheit in der Informationstechnik

BSI

Bundesamt für Sicherheit in der Informationstechnik

Godesberger Allee 185-189, 53175 Bonn ▪ Postfach 200363, 53133 Bonn
Tel.: +49 (0) 1888 9582-0 ▪ Fax: +49 (0) 1888 9582-400 ▪ Internet: www.bsi.bund.de

Foreword

This ‘Protection Profile - Cryptographic Modules, Security Level „Low”- is issued by Bundesamt für Sicherheit in der Informationstechnik, Germany.

The document has been prepared as a Protection Profile (PP) following the rules and formats of Common Criteria version 2.3.

Correspondence and comments to this Protection Profile should be referred to:

CONTACT ADDRESS

Bundesamt für Sicherheit in der Informationstechnik
Godesberger Allee 185-189
D-53175 Bonn, Germany

Tel	+49 228 9582-0
Fax	+49 228 9582-400
Email	bsi@bsi.bund.de

Change history

Version	Date	Reason	Remarks
1.0	28th October 2008	Final version	

Last Version: 1.0 (28th October 2008)

Table of Content

1 PP Introduction

1.1 PP reference

Title: Cryptographic Module, Security Level "Low"
Sponsor: BSI
Editors: T-Systems GEI GmbH, Prüfstelle
CC Version: 2.3
Assurance Level: EAL 4
General Status: final version
Version Number: 1.0
Registration: BSI-CC-PP-0044
Keywords: Cryptography

1.2 PP Overview

This protection profile describes the security requirements for cryptographic modules which provide Endorsed cryptographic security functions with secret or private cryptographic keys and is resistant against low attack potential.

The cryptographic module must not provide non-Endorsed cryptographic security functions. If a cryptographic module uses only asymmetric cryptographic algorithms with public keys (e.g. like a signature-verification application) some of the security requirements required by this protection profile may be not necessary relevant (e.g. side channel resistance).

1.3 Conformance Claim

This protection profile claims conformance to

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 2.3, August 2005, CCMB-2005-08-001

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 2.3, August 2005, CCMB-2005-08-002

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 2.3, August 2005, CCMB-2005-08-003

as follows

- Part 2 extended,
- Part 3,
- Package conformant to EAL4

2 TOE Description

TOE Definition

The Target of Evaluation (TOE) is a cryptographic module that implements Endorsed cryptographic security functions. These Endorsed cryptographic security functions protect the confidentiality or the integrity or both of user data and provide security services according to a security policy of an IT system. The TOE uses, manages and protects the cryptographic keys for these Endorsed cryptographic security functions.

This PP is indented for cryptographic modules, which implement secret or private keys. The cryptographic modules must not provide non-Endorsed cryptographic security functions.

The TOE is physically defined as a set of hardware and software and/or firmware, which is contained within the cryptographic boundary.

The TOE is logically defined by the provided security functions depending on the implemented cryptographic algorithms and protocols. The cryptographic algorithms and protocols provide at least one of the following security functions based on cryptographic key management.

1. Encryption to protect the confidentiality of information represented in ciphertext data, which are known to an attacker if only the decryption key for these data is kept confidentially[1]. The encryption key shall be assigned to the authorized receiver of the information and in case of asymmetric cryptographic algorithm may be public.
2. Decryption to support the protection in confidentiality of information represented in ciphertext data. The decryption key for these data shall be kept confidentially.
3. Digital signature creation to support the services origin authentication, data integrity, and non-repudiation for the signed data to the signer. The signature-creation key shall be kept private.
4. Digital signature verification, which allow to detect any modification of the signed data and to proof the origin and the integrity of unmodified signed data. The signature-verification key shall be authentically assigned to the holder of the signature-creation key and may be public available to the verifier.
5. Generation and the verification of Message Authentication Codes to detect modification of the related data by anybody not knowing the message authentication key used for the Message Authentication Code of these data.
6. Prove of its own identity to an external entity based on the knowledge of a private key without revealing this secret to the verifier.
7. Verification of the identity of an external entity based on a public key assigned to this entity.

The TOE manages the cryptographic keys necessary for its implemented cryptographic algorithms and protocols. The cryptographic key management controls the access and the use

[1] In case of a symmetric encryption algorithm the confidentiality of the decryption key implies the confidentiality of the encryption key because they are identical or the decryption key can be easily derived from the encryption key.

of the cryptographic keys by the Endorsed cryptographic functions. The cryptographic key management includes at least one of the following techniques:

1. Generation of cryptographic keys using a random number generator and implementing the key generation algorithms depending on the intended use of the keys.
2. Import of cryptographic keys in encrypted form or cryptographic key components using split-knowledge procedures.
3. Key agreement protocols establishing common secrets with external entities.

The TOE may export cryptographic keys to authorized external entities while protecting the confidentiality and the integrity as required for the intended use of the cryptographic key.

In many cases the mutual authentication of communicating entities and the key agreement are combined to initiate secure communication between trusted parties protecting the confidentiality and integrity of the transmitted data.

Method of use

The IT system is assumed to protect the confidentiality, the integrity and the availability of the information processed, stored and transmitted according to the IT system security policy. The IT system will use the TOE to protect user data during transmission over channels or storage on media to which unauthorised user have access to. The IT systems security policy defines the protection of the confidentiality or the integrity or both of the user information. It is expressed by a security attribute with values “confidential”, “integrity sensitive” and “confidential and integrity sensitive” assigned to the user information and their user data. The need of protection for the user information defines the need for cryptographic protection of their user data provided by the TOE.

In case of encryption and message authentication with message recovery the information contained in cryptographically protected data cannot be processed until the cryptographic protection is removed. In case of message authentication with appendix the information contained in the cryptographically protected data may be directly processed but the cryptographic integrity protection should be created for the newly generated data. The TOE verifies the data integrity or origin of data received before output them to further processing by the IT system. The protection of the user data passes over to the protection of the cryptographic keys. The TOE IT environment ensures the availability of the user data and the cryptographic keys.

The TOE provides the following types of interfaces/ports[2]:

- Data input interface/port: All data (except control data entered via the control input interface) that is input to and processed by the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and status information from another entities).
- Data output interface/port: All data (except status data output via the status output interface) that is output from the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and control information for another entity).

[2] A port is a physical implementation of an logical interface that provides access to the module for physical signals, represented by logical information flows.

- Control input interface/port: All input commands, signals, and control data (including function calls and manual controls such as switches, buttons, and keyboards) used to control the operation of a cryptographic module shall enter via the "control input" interface.
- Status output interface/port: All input commands, signals, and control data (including calls and manual controls such as switches, buttons, and keyboards) used to control the operation of the cryptographic module).
- Power interface/port: all external electrical power supply.

The key interfaces used for the input and output of plaintext cryptographic key components, CSPs and the authentication interface used for input of confidential authentication data, are logically separated from all other interfaces. All data output via the data output interface is inhibited when the TOE is in an error mode or in power-up self-test mode.

3 Security Problem Definition

3.1 Introduction

This protection profile describes the security problem for cryptographic modules, which may provide a wide range of cryptographic security functions depending on the indented protection of the user data. This indented protection of primary assets is addressed by organisational security policies. The TOE protects the user data in confidentiality and integrity. The use of cryptographic methods implies specific threats, which are common for all TOEs as cryptographic modules.

Assets

The cryptographic module is intended to protect primary assets:

1. **Plaintext data** containing information, which need protection in confidentiality.
2. **Original data** containing information, which need protection in integrity or a proof of origin and authenticity to third parties.

User data requires protection in confidentiality and integrity i.e. they may be original plaintext data.

The use of cryptographic algorithms and protocols requires the protection of the cryptographic keys as secondary assets. The cryptographic keys need protection as the primary assets they protect and depending on the cryptographic technique they are used for:

3. **Secret keys** of symmetric cryptographic algorithms and protocols need protection in confidentiality and integrity.
4. **Private keys** of asymmetric cryptographic algorithms and protocols need protection in confidentiality and integrity.
5. **Public keys** of asymmetric cryptographic algorithms and protocols need protection in integrity and authenticity.

Where the need of confidentiality of secret and private keys follows directly from the cryptographic technique the integrity protection for these keys prevents indirect attacks (e.g. substitution of an unknown secret key by a known key compromise the subsequent encryption of plaintext data, an undetected modification of a private key may enable attacks against this key).

The CC deals with cryptographic keys as user data and as TSF data depending on their specific use by the TSF. Cryptographic keys are user data in the terminology of CC if they are used to protect cryptographically the confidentiality or integrity of data provided by the IT system “cryptographically unprotected data” or to transform “black data” into “cryptographically unprotected data” by cryptographic functions. Encryption and decryption keys are examples of such keys. Cryptographic keys are TSF data in the terminology of CC if their information is used by the TSF in making TSP decisions. Root public keys are examples of cryptographic keys as TSF because they are used to verify the authenticity of all other public keys of the public key infrastructure, which may be provided by any user. Public keys may be used as authentication reference data for external entities as user of the TOE.

Subjects

The following roles are defined in the context of this protection profile. A security target conform to this protection profile shall use all except the Maintenance personal if no maintenance functionality is provided by the TOE of the ST.

Roles	**Description**
Administrator	An authorized user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given to them.
Crypto officer[3]	An authorized user who has been granted the authority to perform cryptographic initialization and management functions. These users are expected to use this authority only in the manner prescribed by the guidance given to them.
End User	An authorized user assumed to perform general security services, including cryptographic operations and other Endorsed security functions.
Maintenance Personal	An authorized user assumed to perform physical maintenance and/or logical maintenance services (e.g., hardware/software diagnostics).
Unidentified User	A user not being identified.
Unauthenticated User	An identified user not being authenticated.

The term “user” is used to include both authorized and unauthorized users. Authorized users are known to the TOE and their security attributes are maintained by the TOE as prerequisite for their identification and authentication. Unauthorized users are unknown to the TOE. An authenticated authorized user is an authorized user that has been successfully authenticated for one or more of the following roles: Administrator role, Crypto Officer role, End User role or Maintenance Personal role. The roles may be refined e.g. the administrator role may be split into a User administrator for user management (i.e. creation and deletion of user accounts) and IT administrator (i.e. management of non-cryptographic functions of the TOE except user).

A user in the End-user role may be a human user or an IT system communicating with the TOE.

The TOE maintains at least the following security attributes of authorized users:

(1) **Identity** that identify uniquely the user,

(2) **Role** for which the user is authorized,

And TSF data

(3) **Reference Authentication Data** for users.

The TOE maintains at least the following security attributes of subjects:

(1) **Identity** of the user bind to this subject,

(2) **Role** for which this user is currently authenticated,

Objects

The following objects are defined in the context of this protection profile. The security target

[3] The “cryptographic administrator” is some times called “crypto officer” in the guidance documentation.

conform to this protection profile may use all or a subset of them depending on the implemented Endorsed cryptographic security functions.

Object	**Description**
Plaintext data	User data encoded in an public known way which will be transformed by an encryption algorithm into ciphertext data (i.e. plaintext input data) or which is the result of decryption of the corresponding ciphertext data (i.e. plaintext output data). Plaintext data contain confidential information.
Ciphertext data	User data as result of the application of an encryption algorithm to plaintext data and an encryption key. The knowledge of ciphertext data by an attacker does not compromise the confidential information represented by the corresponding plaintext.
Original data	User data for which a digital signature or a message authentication code is calculated or verified. Original data contain integrity sensitive information.
Cryptographic keys	Parameters used in conjunction with a cryptographic algorithm that determines the transformation of plaintext data into ciphertext data, the transformation of ciphertext data into plaintext data, a digital signature computed from data, the verification of a digital signature computed from data, a message authentication code computed from data, a proof of the knowledge of a secret, a verification of the knowledge of a secret or an exchange agreement of a shared secret.
Cryptographic key component	Parameters used in split knowledge procedures for manual key export methods and manual key import methods.
Critical security parameters	Security-related information (e.g., secret and private cryptographic keys, and TSF data like authentication data) whose disclosure or modification can compromise the security of a cryptographic module.
Digital signature	The result of an (asymmetric) signature-creation algorithm applied to the original data using a signature-creation key. The digital signature may contain or be appended to the original data.
Message authentication code (MAC)	The result of a (symmetric) message authentication algorithm applied to the original data using a message authentication key. The MAC will be appended to the original data.

Critical security parameters (CSP) have at least the security attributes

(1) **Identity of the CSP** that uniquely identify the CSP,

(2) **CSP usage type** identifying the purpose and methods of use of the CSP,

(3) **CSP access control rules**.

The CSP access control rules may restrict the access for operation like import or export of the key.

Cryptographic keys have at least the security attributes

(1) **Identity of the key** that uniquely identify the key,

(2) **Key entity**, i.e. the identity of the entity this key is assigned to,

(3) **Key type**, i.e. secret key, private key, public key,

(4) **Key usage type**, i.e. the cryptographic algorithms a key can be used for,

(5) **Key access control rules**, and

(6) **Key validity time period**, i.e. the time period for operational use of the key.

The security attribute "key usage type" shall identify the cryptographic algorithm the key is intended to be used and may contain information about the rang of this key in a key hierarchy, and other information. The security attribute "Key access control rules" restricts the access for operation like import or export of the key. The security attribute "key validity time period" restricts the time of operational use of the key; the key must not be used before or after this time slot.

Cryptographic key components have at least the security attributes

(1) **Identity of the key component** that uniquely identify the key component,

(2) **Key entity**, i.e. the identity of the key the key component belongs to,

(3) **Key entry method**, i.e. the method the key component is used for

Furthermore cryptographic keys, key components and CSP may be distinguished as

- Operational if they are used to protect user data,
- Maintenance if they are used for maintenance of the TOE by maintenance personal only.

Note that data used internally by known answer self test of the TOE instead of cryptographic keys are seen neither as operational nor as maintenance keys (CSP).

Operations

The following operations are defined in the context of this protection profile. The security target conform to this protection profile may use all or a subset of them depending on the implemented Endorsed cryptographic security functions.

Operation	**Description**
Decryption	Processes a decryption algorithm to the ciphertext data using the decryption key and returns the corresponding plaintext data
Encryption	Processes a encryption algorithm to the plaintext data using the encryption key and returns the corresponding ciphertext data
Export of key	output of cryptographic keys in protected form
Export of protected data	Output of user data with or without security attributes to the black area of the IT system protected in confidentiality or integrity or both by cryptographic security functions of the TOE
Export of unprotected data	Output of user data with or without security attributes to the red area of the IT system cryptographically protected by cryptographic security functions of the TOE
Import of key	input of cryptographic keys in protected form
Import of protected data	Input of user data with or without security attributes from the black area of the IT system where the cryptographic security functions of the TOE support the protected in confidentiality by decryption or in integrity by detection modification or verification of data origin

Operation	Description
Import of unprotected data	Input of user data with or without security attributes to the red area of the IT system cryptographically unprotected by cryptographic security functions of the TOE
MAC calculation	Processes a (symmetric) MAC algorithm to the original data using the secret message authentication key and returns the corresponding Message Authentication Code
MAC verification	Processes a (symmetric) MAC algorithm to the presented user data and MAC using the secret message authentication key and returns the result of checking whether the user data, the MAC and the key fit together (integrity confirmed) or not (integrity not confirmed)
Signature-creation	Processes a (asymmetric) signature-creation algorithm to the original data using the private signature-creation key of the signatory and returns the corresponding digital signature
Signature-verification	Processes a (asymmetric) signature-verification algorithm to the signed data and the digital signature using the public key and returns the result of checking whether the original data, the electronic signature and the public key fit together (integrity confirmed) or not (integrity not confirmed)
Use of key	Use of the cryptographic key by a cryptographic algorithm as key parameter[4]

3.2 Assumptions

A.User_Data Protection of user data by the IT system

The TOE environment uses the TOE for cryptographic protection of user data for transmission over channels or storage in media, which are not protected against access by unauthorised users. The TOE environment provides cryptographically unprotected user data to the TOE and identifies protection in confidentiality or integrity or both to be provided by the TOE.

A.Data_Sep Separation of cryptographically protected and unprotected data

The TOE environment separates the cryptographically unprotected data from the cryptographically protected user data in the IT system.

A.Key_Generation Key generation and import to the cryptographic module

Cryptographic keys generated by the IT environment and imported into the TOE are cryptographically strong for the intended key usage and have secure security attributes.

A.Availability Availability of keys

The TOE environment ensures the availability of cryptographic keys, key components, CSP and key material.

[4] E.g. if an encryption key *A* is encrypted with a key-encryption key *B* than the *B* is "used as key" (not *A*).

3.3 Threats

The cryptographic modules protect user data as primary assets by means of cryptographic functions. The cryptographic functions, their keys and CSP itself are object of attacks. These attacks are described here.

T.Compro_CSP Compromise of confidential CSP

An attacker with low attack potential may compromise confidential CSP like secret keys, private keys or confidential authentication data, which enables attacks against the confidentiality or integrity of user data protected by these CSPs or the TSF using these CSPs as TSF data.

T.Modif_CSP Modification of integrity sensitive CSP

An attacker with low attack potential may modify integrity sensitive CSP like permanent stored public keys and therefore compromise the confidentiality or integrity of user data protected by these CSPs or the TSF using these CSPs as TSF data.

T.Abuse_Func Abuse of function

An attacker with low attack potential may use TOE functions intended for installation, configuration or maintenance of the TOE which shall not be used for operational cryptographic keys or user data in order (i) to disclose or manipulate operational CSP or user data, or (ii) to enable attacks against the integrity or confidentiality of operational CSP or user data by (iia) manipulating (explore, bypass, deactivate or change) security features or functions of the TOE or (iib) disclosing or manipulating TSF Data.

T.Inf_Leakage Information leakage

An attacker with low attack potential may observe and analyse any energy consumed or emitted through the cryptographic boundary (i.e. including the external interfaces) of the TOE to get internal secrets (especially secret or private cryptographic keys) or confidential user data not intended for export. The information leakage may be inherent in the normal operation or caused by the attacker.

T.Malfunction Malfunction of TSF

An attacker with low attack potential may use a malfunction of the hardware or software, which is accidental or deliberated by applying environmental stress or perturbation, in order to deactivate, modify, or circumvent security functions of the TOE to enable attacks against the integrity or confidentiality of the User data or the CSP.

T.Physical_Tamper Physical tampering

An attacker with low attack potential may tamper the cryptographic module to get secrets, to modify data on whose integrity the TSF relies, or to corrupt or de-activate the TSF inside the cryptographic boundary to violate the integrity or confidentiality of the User data, the CSP or the TSF data.

T.Masquerade Masquerade authorized data source or receiver

An attacker with high attack potential may masquerade as an authorized data source or receiver to perform operations that will be attributed to the authorized user or may gain undetected access to cryptographic module causing potential violations of integrity or confidentiality of the User data, the CSP or the TSF data.

3.4 Organisational Security Policies

OSP.User_Data_Prot Protection of user data by cryptographic functions

The cryptographic module will be used to protect the confidentiality or integrity or both of information represented by user data which may be get known or modified by an attacker. The IT system will ensure the availability of the user data and the cryptographic keys outside the cryptographic module.

OSP.Resist_Low Resistance against low attack potential

The TOE shall resist attacks with low attack potential.

OSP.I&A Identification and authentication of users

All users must be identified and authenticated prior to accessing any controlled resources with the exception of read access to public objects and cryptographic operations with public keys.

OSP.Access Access control of TOE functions

The TOE must limit the extent of each user’s abilities to use the TOE functions in accordance with the TSP.

OSP.Roles Roles

The authorized administrator, cryptographic administrator and end-users shall have separate and distinct roles associated with them. If the TOE provides maintenance functionality the maintenance personal shall have distinct roles associated with them and separate from other roles.

OSP.Endorsed_Crypto Endorsed cryptographic functions

The TOE shall implement Endorsed cryptographic algorithms and Endorsed cryptographic protocols for the protection of the confidentiality or the integrity or both of the user data according to the organizational security policy OSP.User_Data_Prot and for the cryptographic key management according to the organizational security policy OSP.Key_Man. The cryptographic module must not provide any non-Endorsed cryptographic function.

OSP.Key_Man Cryptographic key management

The CSP, cryptographic keys and cryptographic key components are assigned to cryptographic algorithms and protocols they are intended to be used with and the entities, which are allowed to use them.

OSP.Key_Personal Personal security for cryptographic keys

The cryptographic keys shall be managed in such a way that their integrity and confidentiality cannot be compromised by a single person.

3.5 Security Objectives

3.5.1 Security Objectives for the TOE

O.Red-Black-Sep Red-black separation of the TOE

The TOE shall protect confidential information for export into the black area by encryption of plaintext data and for import into the red area by decryption of ciphertext data. The TOE shall protect integrity sensitive information for export into the black area by calculation of MAC or digital signature on the red data and for import into the red area by verification of MAC or digital signature on black data. The TOE shall separate logical interfaces for red user data, black user data, CSP (including plaintext cryptographic keys and key components) and administrative functions.

O.Endorsed_Crypto Endorsed cryptographic functions

The TOE shall provide Endorsed cryptographic functions and Endorsed cryptographic protocols to protect the user data as required by OSP.User_Data_Prot and for key management.

O.I&A Identification and authentication of users

The TOE shall uniquely identify users and verify the claimed identity of the user before providing access to any controlled resources with the exception of read access to public objects. The security functions for authentication of users shall have strength "high".

O.Roles Roles known to TOE

The TOE shall provide at least the Administrator, the Cryptographic Administrator, and the End-user roles. If the TOE provides maintenance functionality the TOE shall provide Maintenance Personal role.

O.Control_Services Access control for services

The TOE shall restrict the access to its services, depending on the user role, to those services explicitly assigned to this role. Assignment of services to roles shall be either done by explicit action of an Administrator or by default.

O.Control_Keys Access control for cryptographic keys

The TOE shall restrict the access to the keys, key components and other CSP according to their security attributes. Cryptographic keys intended for the use with Endorsed cryptographic functions must not be used by any non-endorsed functions.

O.Key_Export Export of cryptographic keys

The TOE shall export cryptographic keys with their security attributes. The cryptographic keys and their security attributes shall be protected in integrity. The TOE shall ensure the confidentiality of secret and private keys exporting them in encrypted form to authorized entities or manually using split knowledge procedures only.

O.Key_Generation Generation of cryptographic keys by the TOE

The TOE shall generate cryptographic strong keys using Endorsed cryptographic key generation algorithms.

O.Key_Import Import of cryptographic keys

The TOE shall import keys with security attributes and verify their integrity. The TOE shall import secret or private keys in encrypted form or manually using split knowledge procedures only.

O.Key_Management Management of cryptographic keys

The TOE shall securely manage cryptographic keys, cryptographic key components and CSP. The TOE shall associate security attributes of the entity the key is assigned to and of the intended cryptographic use of the key. Assignment of the security attributes to the cryptographic keys, cryptographic key components and CSP shall be either done by explicit action of a Cryptographic Administrator or by default.

O.Key_Destruction Destruction of cryptographic keys

The TOE shall destruct in a secure way the keys cryptographic key components and other CSP on demand of authorized users or when they will not be used any more that no information about these keys is left in the resources storing or handling these objects before destruction.

O.Check_Operation Check for correct operation

The TOE shall perform regular checks to verify that its components operate correctly. This includes integrity checks of TOE software, firmware, internal TSF data and keys during initial start-up, at the request of the authorised user, and at the conditions installation and maintenance.

O.Physical_Protect Physical protection

The TOE shall resist physical attacks with low attack potential.

O.Prevent_Inf_Leakage Prevent leakage of confidential information

The TOE shall prevent information leakage about secret and private keys and confidential TSF data outside the cryptographic boundary and unintended output confidential user information. The TOE shall resist attacks with low attack potential, which are based on information leakage.

3.5.2 Security Objectives for the Environment

OE.Assurance Assurance Security Measures in Development and Manufacturing Environment

The developer and manufacture ensure that the TOE is designed and fabricated so that it requires a combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through attack. The developer provides necessary evaluation evidence that the TOE fulfils its security objectives and is resistant against attack with low attack potential.

OE.Key_Generation Key generation by IT environment

The IT environment shall ensure the cryptographic strength, the confidentiality and integrity of secret and private keys, the integrity and authenticity of public keys and correct security attributes if they are generated outside the TOE and imported into the TOE.

OE.Red-Black-Sep Separation of red and black area of the IT system

The TOE environment protects the user data in the red area of the IT system and controls the exchange data between the red and black area of the IT system according to the IT security policy. It provides the red user data with their security attributes for cryptographic protection to the TOE and receives red user data with their security attributes from the TOE.

OE.Personal Personal security

The Administrator, Cryptographic Administrator, End-user roles, and - if supported by the TOE - the Maintenance role shall be assigned with distinct authorized persons. For manual key import at least two different authorized persons are assigned to cryptographic administrator role.

OE.Key_Availability Availability of cryptographic key and key material

The IT environment shall ensure the availability of the user data, cryptographic keys key components, CSP and key material.

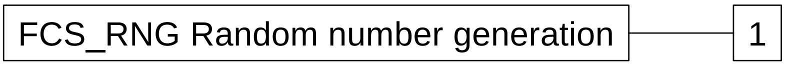
4 Extended Components Definition

4.1 Definition of the Family FCS_RNG

Family behaviour

This family defines requirements for the generation random number where the random numbers are intended to be used for cryptographic purposes. The requirements address the type of the random number generator as defined in AIS 20/31[5] and quality of the random numbers.

Component levelling:



FCS_RNG.1 Generation of random numbers requires that random numbers meet a defined quality metric.

Management: FCS_RNG.1
There are no management activities foreseen.

Audit: FCS_RNG.1
There are no actions defined to be auditable.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.
Dependencies: FPT_TST.1.

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

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

4.2 Definition of the Family FPT_EMSEC

Family behaviour

This family defines requirements to mitigate intelligible emanations.

Component levelling:

[5] New version currently under development

Management: FPT_EMSEC.1

There are no management activities foreseen.

Audit: FPT_EMSEC.1

There are no actions identified that should be auditable if FAU_GEN Audit data generation is included in the PP/ST.

FPT_EMSEC.1 TOE Emanation

Hierarchical to: No other components.

Dependencies: No other components.

FPT_EMSEC.1.1 The TOE shall not emit [*assignment: types of emissions*] in excess of [*assignment: specified limits*] enabling access to [*assignment: list of types of TSF data*] and [*assignment: list of types of user data*].

FPT_EMSEC.1.2 The TSF shall ensure [*assignment: type of users*] are unable to use [*assignment: types of interfaces/ports*] to gain access to [*assignment: list of types of TSF data*] and [*assignment: list of types of user data].*

4.3 Definition of the Security Functional Component FPT_TST.2

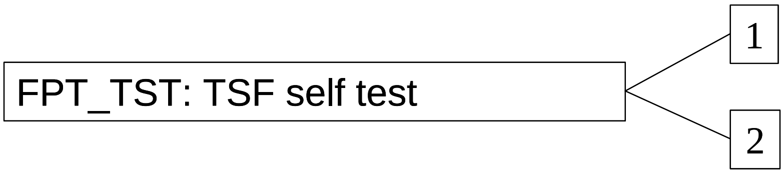
The following addition are made to “TSF self test (FPT_TST)” in Common Criteria, Part 2 to require the self-testing of TSF and of the integrity of the TSF-data and TSF-executable code. FPT_TST.2 requires the behaviour of TSF during self-testing and the actions to be performed by TSF in dependency of the results of the self-testing. This kind of requirements lies beyond FPT_TST.1 defined in Common Criteria, Part 2.

Family behaviour

The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. Examples are interfaces to enforcement functions, and sample arithmetical operations on critical parts of the TOE. These tests can be carried out at start-up, periodically, at the request of the authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self testing are defined in other families.

The requirements of this family are also needed to detect the corruption of TSF executable code (i.e. TSF software) and TSF data by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TSF due to inadequate logical and/or physical protection.

Component levelling:



FPT_TST.1 TSF testing, provides the ability to test the TSF's correct operation. These tests may be performed at start-up, periodically, at the request of the authorised user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and executable code.

FPT_TST.2 TSF self-testing requires self-testing capabilities of the TSF correct operation. These tests must be performed at start-up. Conditional and on demand by a user self-testing may be required. Particular TSF behaviour during self-testing and TSF-actions after self-testing are required.

Management: FPT_TST.2

There are no management activities foreseen.

Audit: FPT_TST.2

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

a) Basic: Execution of the TSF self tests and the results of the tests.

FPT_TST.2 TSF self-testing

Hierarchical to: No other components.

Dependencies: FPT_FLS.1 Failure with preservation of secure state.

FPT_TST.2.1 The TSF shall perform self-testing at power-up to verify the correctness of [assignment: *list of cryptographic algorithms*] and of [assignment: *list of critical TSF*], and to verify the integrity of the TSF-software/firmware.

FPT_TST.2.2 The TSF shall perform self-testing at the conditions [assignment: *list of conditions*] to verify the correctness of [assignment: *list of critical cryptographic algorithms*].

FPT_TST.2.3 The TSF shall perform self-testing at the conditions [assignment: *list of conditions*] to verify the correctness of [assignment: *list of critical TSF*], and to verify the integrity of [assignment: *list of TSF data*].

FPT_TST.2.4 The TSF shall perform self-testing at the conditions [assignment: *list of conditions*] to verify the integrity of [assignment: *list of TSF-objects*].

FPT_TST.2.5 The TSF shall provide [assignment: *list of users*] with the capability to invoke the following self-tests [assignment: *list of self-tests*].

FPT_TST.2.6 During [assignment: *list of self-tests*] the TSF shall [assignment: *list of actions to be performed*].

FPT_TST.2.7 After completion of self-testing the TSF shall [assignment: *list of actions to be performed*].

FPT_TST.2.8 If the self-testing result is fail the TSF shall [assignment: *list of actions to be performed*].

5 Security Requirements

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph 2.1.4 of part 2 of the CC. Each of these operations is used in this PP.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is (i) denoted by the word "refinement" in **bold** text and the added/changed words are in bold text, or (ii) included in text as **bold** text and marked by a footnote. In cases where words from a CC requirement were deleted, a separate attachment indicates the words that were removed.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the PP authors are denoted as *italic* text and the original text of the component is given by a footnote. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made, [selection:], and are *italicized*.

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 and the original text of the component is given by a footnote. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:], 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.

5.1 Security Functional Requirements for the TOE

5.1.1 Cryptographic operation and key management

The TOE shall meet the requirement "Cryptographic key generation (FCS_CKM.1)" as specified below (Common Criteria Part 2).

FCS_CKM.1 Cryptographic key generation

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

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*assignment: cryptographic key generation algorithm*] and specified cryptographic key sizes [*assignment: cryptographic key sizes*] that meet the following: [assignment: *list of **Endorsed** standards*].

Application note 1: The ST writer shall perform the missing operations in the element FCS_CKM.1.1. If the TOE implements more than one key generation method the component shall be iterated to describe all key generation methods under evaluation. The assignment of the standard shall indicate Endorsed algorithms only. All keys used for Endorsed functions shall be generated by Endorsed key generation algorithms. Endorsed key generation algorithms use Endorsed random generators only.

FCS_CKM.2/Import Cryptographic key distribution

Hierarchical to: No other components.

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

FCS_CKM.2.1/Import The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method *key entry*[6]*##* that meets the following: [assignment: *list of* ***Endorsed*** *standards*][7].

Refinement

The key entry shall be performed using either manual or electronic methods.

Manually-entered keys shall be verified for accuracy of the input into the TOE.

Secret and private keys established using manual methods shall be entered either

(1) in encrypted form or

(2) using split knowledge procedures.

If split knowledge procedures are used:

(1) At least two key components shall be required to reconstruct the original cryptographic key,

(2) if knowledge of n key components is required to reconstruct the original key, then knowledge of any n-1 key components provides no information about the original key other than the length.

All secret or private keys that are imported into the TOE in encrypted form shall be encrypted and integrity protected using an Endorsed cryptographic algorithm. All public keys electronically entered into the TOE shall be integrity protected using an Endorsed cryptographic algorithm.

Application note 2: The ST writer shall perform the missing operations in the element FCS_CKM.2.1/Import. The ST writer shall describe all methods of key import provided by the TOE. If the TOE implements more than one method of key import the component should be iterated. The assignment of the standards shall assign Endorsed algorithms only. Manual key input may be used e.g. for secret transport keys for symmetric encryption of keys.

FCS_CKM.2/Export Cryptographic key distribution

Hierarchical to: No other components.

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

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

[7] [assignment: *list of standards*].

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

FCS_CKM.2.1/Export The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method *key export*[8] that meets the following: [assignment: *list of* ***Endorsed*** *standards*][9].

Refinement

The key export shall be performed using either manual or electronic key export methods.

Key components exported for manual key entry method shall support the verification for accuracy of the key material. Secret and private keys exported for manual key entry method shall be exported either

(1) in encrypted form or

(2) using split knowledge procedures.

If split knowledge procedures are used:

(1) at least two key components shall be required to reconstruct the original cryptographic key,

(2) if knowledge of n key components is required to reconstruct the original key, then knowledge of any n-1 key components provides no information about the original key other than the length.

All secret or private keys exported in encrypted form by the TOE shall be encrypted and integrity protected using an Endorsed cryptographic algorithm. All public keys exported for electronic key entry method hall be integrity protected using an Endorsed cryptographic algorithm.

Application note 3: The ST writer shall perform the missing operations in the element FCS_CKM.2.1/Export. The ST writer shall describe **all** methods of key export provided by the TOE. The assignment of the standards shall assign Endorsed algorithms only. If the TOE implements more than one method of key import the component should be iterated.

FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FTP_ITC.1.2 The TSF shall permit [selection: *the TSF, the remote trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for *electronic key distribution according to FCS_CKM.2/Import and FCS_CKM.2/Export*[10].

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

[9] [assignment: *list of standards*].

Application note 4: The ST writer shall perform the missing operation in the element FTP_ITC.1.2. The trusted channel for key import and key export will be established for electronic key distribution.

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 [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of **Endorsed** standards*].

Application note 5: The ST writer shall perform the missing operations in the element FCS_CKM.4.1. If the TOE implements more than one key destruction method the component should be iterated. The assignment of the standards shall assign Endorsed algorithms only.

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

MT_MSA.2 Secure security attributes

FCS_COP.1.1 The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of **Endorsed** standards*].

Application note 6: The ST writer shall perform the missing operations in the element FCS_COP.1.1. The assignment of the standards shall assign Endorsed algorithms only. If the TOE implements more than one cryptographic operation the component shall be iterated.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: FPT_TST.1 TSF testing.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid*] random number generator that meet [selection: *Endorsed RNG class*] [11].

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

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

[11] [assignment: *list of security capabilities*]

Application note 7: The ST writer shall perform the missing operations in the elements FCS_RNG.1.1 and FCS_RNG.1.2. The ST writer shall describe the requirements for all RNG used in the TOE by FCS_RNG.1, possibly by iterations. Endorsed functions use only Endorsed RNG according to FCS_RNG.1. A physical random number generator (RNG) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (key strokes, mouse movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs e.g. a physical RNG with cryptographic post-processing. Endorsed functions use only Endorsed RNG according to FCS_RNG.1. The security capabilities for the assignment in FCS_RNG.1.1 are provided by the Endorsed RNG class. If the seed of a DRNG is entered during key generation, shall be entered as key according with FCS_CKM.2. The quality metric of the random numbers should be chosen depending on the RNG type and the intended application of the random numbers. E.g. a DRNG used to generate key pairs for qualified electronic signatures shall have to be seeded with minimum 100 bit Min-entropy and for AES keys 128 bit Min-entropy.

5.1.2 User I&A

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:

(1) *Identity,*

(2) *Role,*

(3) *Reference authentication data,*

(4) *[assignment: list of additional security attributes]*[12].

Application note 8: The element FIA_ATD.1.1 contains an assignment of the list of security attributes in the bullet (1), (2) and (3) and allows the ST writer to add none or an additional list of security attributes in bullet (4).

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1 The TSF shall allow

(1) Self test according to FPT_TST.2,

(2) [assignment: list of cryptographic operations with public keys]

(3) [assignment: list of other TSF-mediated actions][13]

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

[12] [assignment: *list of security attributes*]

[13] [assignment: *list of TSF mediated actions*]

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

Application note 9: The ST writer shall perform the missing operation in the element FIA_UID.1.1 by adding other TSF mediated actions or none of them if the Unidentified User is not allowed to run other TSF mediated actions than self test.

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.2,*

(2) *[assignment: list of cryptographic operations with public keys]*

(3) *Identification according to FIA_UID.1,*

(4) *Selection of [selection: a role, a set of role],*

(5) *[assignment: list of other TSF mediated actions]*[14]

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

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

Application note 10: The ST writer shall perform the missing operations in the element FIA_UAU.1.1. The list of TSF mediated actions defines the rights assigned to the role Unauthenticated User. Note a role is "a predefined set of rules establishing the allowed interactions between a user and the TOE" (CC part 1, para. 45). The selection in the fourth bullet allows the ST writer to describe how the user may take the role or the roles for the user session. The selection in the fifth bullet allows the ST writer to add other TSF mediated actions or none of them. The TSF for authentication of the users shall have a high strength of function as required by AVA_SOF.1. For each attempt to use the authentication mechanism the probability shall be less or equal than 10^{-6} that a random attempt will succeed or a false acceptance will occur. For multiple attempts to use the authentication mechanism the probability shall be or equal less than $3 \cdot 10^{-6}$ that a random attempt will succeed or a false acceptance will occur.

FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

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

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

(2) *power on or reset,*

(3) [assignment: *list of other conditions under which re-authentication is required*][15].

[14] [assignment: *list of TSF mediated actions*]

[15] [assignment: *list of conditions under which re-authentication is required*]

Application note 11: The ST writer shall perform the missing operation in the element FIA_UAU.6.1 by adding other conditions or none of them in the third bullet. The use may select a role or a set of roles (if supported by the TOE cf. selection in bullet (3) of the element FIA_UAU.1.1). If the TOE support the authentication of a user for a set of roles (cf. to FIA_UAU.1.1), the user is authorized the role or the set of roles and successful authenticated the TSF may bind subjects to the claimed roles. The user may change the role without re-authentication within this set of roles he is authenticated for.

FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UAU.7.1 The TSF shall provide only [assignment: *list of feedback*] to the user while the authentication is in progress.

Application note 12: The ST writer shall perform the missing operation in the element FIA_UAU.7.1. The feedback provided to the user must not include (i) any information about the verification authentication data or (ii) any information about failure or success of the authentication attempt before the authentication procedure is finished. The feedback may indicate the left authentication attempts for the selected identification. The feedback after the defined number of unsuccessful authentication attempts has been met or surpassed shall be described in the element FIA_AFL.1.2.

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,

(3) *[assignment: list of additional user security attributes]*[16].

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*[17].

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) the subject attribute Role shall be changed from Unidentified user to Unauthenticated user after successful identification;

(2) after successful authentication the subject attribute Role shall be changed from Unauthenticated User to a role that the user has selected for the authentication session if the user is authorized for this role;

[16] [assignment: *list of user security attributes*]

[17] [assignment: *rules for the initial association of attributes*]

(3) *after successful re-authentication of the user the subject attribute Role shall be changed to a role that the user has selected for the authentication session if the user is authorized for this role;*

(4) *[assignment: additional rules for the changing of attributes]*[18].

Application note 13: The ST writer shall perform the missing operation in the elements FIA_USB.1.1 by adding additional security attributes or none of them in the third bullet. The ST writer shall perform the missing operation in the elements FIA_USB.1.3 by adding additional rules or none of them in the fourth bullet. The authentication session is the time between the successful authentication and next re-authentication or logout of the user.

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when [selection: *[assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]*] unsuccessful authentication attempts occur related to [assignment: *list of authentication events*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [assignment: *list of actions*].

5.1.3 Protection of user data

FDP_ACC.2/Key_Man Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1/Key_Man The TSF shall enforce the *Key Management SFP* [19] on

(1) *all cryptographic keys, key components, CSP;*

(2) *all user subjects* [20]

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/Key_Man The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

FDP_ACF.1/Key_Man Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/Key_Man The TSF shall enforce the *Key Management SFP* [21] to objects based on the following:

[18] [assignment: *rules for the changing of attributes*]

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

[20] [assignment: *list of subjects and objects*]

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

(1) Subjects with security attributes: Identity of the user the subject is bind to, Role of this user;

(2) Objects

(a) Cryptographic keys with security attributes: Identity of the key, Key entity, Key type, Key usage type, Key access control rules, Key validity time period;

(b) Key components with security attributes: Identity of the key component, Key entity, Key entry method,

(c) CSP with security attributes: Identity of the CSP, CSP usage type, CSP access control rules[22].

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

(1) Subject in crypto officer role is allowed to import encrypted secret and private keys if the security attribute Key access control rules of the key allows import;

(2) Subject in crypto officer role is allowed to import one key component of a key with the key entry method assigned to the key component;

(3) Subject in crypto officer role is allowed to import CSP,

(4) Subject in crypto officer role is allowed to export encrypted secret or private keys if the security attribute Key access control rules of the key allows export;

(5) Subject in crypto officer role is allowed to export one key component of a key with the key entry method assigned to the key component;

(6) Subject in crypto officer role is allowed to export CSP if the security attribute CSP access control rules of the CSP allows export;

(7) Subject in crypto officer role is allowed to destruct cryptographic keys, cryptographic key components and CSP;

(8) [assignment: other rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects][23].

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

(1) Subjects in Maintenance role are allowed to import and destruct maintenance cryptographic keys, key components and CSP;

(2) [assignment: *additional rules, based on security attributes, that explicitly authorise access of subjects to objects*].

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

(1) Subject in crypto officer role is not allowed to import a key component if the same subject or an other subject with the same Identity of the user

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

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

already input a key component with a different Identity and the same Key entity;

(2) *Subject in crypto officer role is not allowed to export a key component if the same subject or an other subject with the same Identity of the user already export a key component with a different Identity and the same Key entity;*

(3) *Subjects with other roles than crypto officer rule are not allowed to input operational public root key;*

(4) *Subjects with other roles than crypto officer rule are not allowed to input permanent stored operational secret keys, private keys, key components and CSP;*

(5) *No subject is allowed to import or export secret key or private keys in plaintext;*

(6) *No subject is allowed to use keys by operation other than identified in Key usage type and the Key access control rules;*

(7) *[assignment: other rules, based on security attributes, that explicitly deny access of subjects to objects]*[24].

Application note 14: The ST writer shall perform the missing operation in the element FDP_ACF.1.2, FDP_ACF.1.3 and FDP_ACF.1.4. The operation in the element FDP_ACF.1.2 shall describe

(1) The Key access control rules protecting secret keys, private keys, and CSPs within the cryptographic module from unauthorized disclosure, modification, and substitution,

(2) The Key access control rules protecting public keys within the cryptographic module against unauthorized modification and substitution.

The other rules in the element FDP_ACF.1.2 may address e.g. the export of public keys. Note if a subject, an object or an operation identified in any rule in the component FDP_ACF.1/Key_Man is not supported by the TOE the access rule is fulfilled obviously. (End of Application note.)

FDP_ACC.2/Oper Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1/Oper The TSF shall enforce the *Cryptographic Operation SFP* [25] on

(1) *operational cryptographic keys, CSP,*

(2) *plaintext data, ciphertext data, original data;*

(3) *all user subjects* [26]

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/Oper The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

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

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

[26] [assignment: *list of subjects and objects*]

FDP_ACF.1/Oper Security attribute based access control

Hierarchical to: No other components.

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

FDP_ACF.1.1/Oper The TSF shall enforce the *Cryptographic Operation SFP* [27] to objects based on the following:

(1) *Subjects with security attributes: Identity of the user the subject is bind to, Role of this user;*

(2) *Objects*

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

(b) *Operational CSP with security attributes: Identity of the CSP, CSP usage type, CSP access control rules,*

(c) *plaintext data, ciphertext data, original data*[28].

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

(1) *Subject in Unidentified-user role is allowed to perform cryptographic operation with public keys in accordance with the security attributes of the used cryptographic keys and CSP;*

(2) *Subject in Unauthenticated-user role is allowed to perform cryptographic operation with public keys in accordance with the security attributes of the used cryptographic keys and CSP;*

(3) *Subject in End-user role is allowed to perform cryptographic operation with public, private, and secret keys in accordance with the security attributes of the used cryptographic keys and CSP;*

(4) *[assignment: other rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]*[29].

FDP_ACF.1.3/Oper The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].

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

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

(2) *No subject is allowed to use CSP by cryptographic operation other than identified in the security attributes CSP usage type and the CSP access control rules;*

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

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

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

(3) *[assignment: other rules, based on security attributes, that explicitly deny access of subjects to objects]*[30].

Application note 15: The ST writer shall perform the missing operation in the element FDP_ACF.1.2/Oper, FDP_ACF.1.3/Oper and FDP_ACF.1.4/Oper.

FDP_ACC.2/Mode_Trans Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1/Mode_Trans The TSF shall enforce the *Mode transition SFP* [31] on *all subjects and the mode variable* [32] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/Mode_Trans The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

Application note 16: The mode variable defines the current mode of the TOE in the finite state model (cf. ADV_SPM.1 for more details). TOE modes of operation define a set of functionality available within the mode. E.g. in User mode the data interfaces are open for encryption/ decryption of user data with operational keys but key management functions are blocked. In Crypto officer mode key management functions are available but the data interfaces for the encryption/ decryption of user data with operational key are blocked.

FDP_ACF.1/Mode_Trans Security attribute based access control

Hierarchical to: No other components.

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

FDP_ACF.1.1/Mode_Trans The TSF shall enforce the *Mode transition SFP* [33] to objects based on the following: *all subjects and the mode variable*[34].

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

(1) *the subject in Crypto officer role is allowed to change the mode variable to a Crypto officer mode, Key/CSP entry mode, User mode, and Self-test mode;*

(2) *the subject in User role is allowed to change the mode variable to User mode;*

(3) *the subject in the Maintenance role is allowed to change the mode variable to a Maintenance mode after destruction of all operational secret and private keys and unprotected CSP,*

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

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

[32] [assignment: *list of subjects and objects*]

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

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

(4) *the subject in the Maintenance role is allowed to change the mode variable from a Maintenance mode to other value only after destruction of all maintenance key and CSP* [35].

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

(1) *the TOE shall enter automatically the Error mode from any mode of operation except Power-off mode and Maintenance mode, when failure listed in FPT_FLS.1 occur,*

(2) *[assignment: additional rules, based on security attributes, that explicitly authorise access of subjects to objects]*[36].

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

(1) *Subjects in other roles than the Crypto officer are not allowed to change the mode variable to a Crypto officer mode or a Key/CSP entry mode;*

(2) *Subjects in other roles than the Maintenance role are not allowed to change the mode variable to a Maintenance mode;*

(3) *[assignment: other rules, based on security attributes, that explicitly deny access of subjects to objects]* [37].

Application note 17: If the TOE does not provide any maintenance functionality the Maintenance modes do not exist and the Maintenance Personal Role is superfluous (cf. to FMT_SMR.2). In this case the rules (2) and (3) in FDP_ACF.1.2/Mode_Trans, (1) in FDP_ACF.1.2/Mode_Trans and (2) in FDP_ACF.1.4/Mode_Trans are obviously fulfilled.

FDP_ITC.2 Import 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]
[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 The TSF shall enforce the *Key Management SFP and Red-black separation SFP* [38] when importing user data, controlled under the SFP, from outside of the TSC.

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

FDP_ITC.2.3 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 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.

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

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

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

[38] [assignment: *access control SFP and/or information flow control SFP*]

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC

(1) *keys shall be imported with the security attributes Key identity, Key entity, Key type, Key usage type and Key validity time period;*

(2) *key components shall be imported with the security attributes Identity of the Key, Key entity, Key entry method;*

(3) *CSP shall be imported with security attributes Identity of the CSP and CSP usage type;*

(4) *all secret and private keys imported into the TSC shall be encrypted or entered using split knowledge procedures using an Endorsed algorithm*[39].

Application note 18: All secret and private keys entered into the TOE and used by an Endorsed function shall be imported in encrypted form or by split knowledge procedures (cf. FCS_CKM.2/Import).

FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

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

FDP_ETC.2.1 The TSF shall enforce the *Key Management SFP and Red-black separation SFP* [40] when exporting user data, controlled under the SFP(s), outside of the TSC.

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

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

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

(1) *keys shall be exported with the security attributes Key identity, Key entity, Key type, Key usage type and Key validity time period;*

(2) *secret and private keys exported in encrypted form shall be exported with additional security attribute: Identity of the key encryption key under which they are encrypted;*

(3) *key components shall be exported with the security attributes Identity of the Key component, Key entity, Key entry method;*

(4) *CSP shall be exported with security attributes Identity of the CSP and CSP usage type;*

(5) *all secret and private keys exported from the TSC shall be encrypted or protected by split-knowledge procedure using an Endorsed algorithm* [41].

FDP_UCT.1 Basic data exchange confidentiality

Hierarchical to: No other components.

[39] [assignment: *additional importation control rules*]

[40] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

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

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

FDP_UCT.1.1 The TSF shall enforce the *Red-black separation SFP*[42] **by providing the ability** to *transmit and receive*[43] objects in a manner protected from unauthorised disclosure.

Application note 19: The element FDP_UCT.1 was refined by substituting “the TSF shall enforce … to be able to” by “the TSF shall enforce… **by providing the ability to**” to ensure the confidentiality of user data when it is transferred using an external channel between distinct TOEs or users on distinct TOEs.

FDP_UIT.1 Data exchange integrity

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]

FDP_UIT.1.1 The TSF shall enforce the *Red-black separation SFP*[44] to be able to *transmit and receive*[45] user data in a manner protected from [selection: *modification, deletion, insertion, replay]* errors.

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

FDP_RIP.2 Full residual information protection

Hierarchical to: FDP_RIP.1 Subset residual information protection

Dependencies: No dependencies.

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] all objects.

5.1.4 Management of TSF and protection of TSF data

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 security management functions:

[42] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[43] [selection: *transmit, receive*]

[44] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[45] [selection: *transmit, receive*]

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

(2) *management of Authentication Reference Data (FMT_MTD.1/Admin, FMT_MTD.1/User),*

(3) *management of security attributes of cryptographic keys, cryptographic key components and CSP (FMT_MSA.1/Key_Man_1, FMT_MSA.1/Key_Man_2, FMT_MSA.2, FMT_MSA.3),*

(4) *[assignment: list additional of security management functions to be provided by the TSF]*[46].

FMT_SMR.2 Restrictions on security roles

Hierarchical to: FMT_SMR.1 Security roles

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.2.1 The TSF shall maintain the roles: *End User Role, Crypto Officer Role, Administrator Role, Unidentified User Role, Unauthenticated User Role, [assignment: other roles]*[47].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

1. *Any user identity assigned to the Administrator Role must not be assigned to the End-user Role or the Crypto Officer Role,*
2. *Any user identity assigned to the Crypto Officer Role must not be assigned to the End-user Role or the Administrator Role,*
3. [assignment: *other conditions for the different roles*][48]

are satisfied.

Refinement:

If the TOE provides maintenance functionality the TOE shall

(1) maintain the Maintenance Personal Role,

(2) any user identity assigned to the Administrator Role or Crypto Officer Role must not be assigned to the Maintenance Personal Role.

Application note 20: The ST writer may introduce other roles depending on the TOE life cycle e.g. Personalization Agent Role as sub-set of the Administrator Role. If the TOE does not provide any maintenance functionality the Maintenance Personal Role is superfluous.

FMT_MOF.1/CO Management of security functions behaviour

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1/CO The TSF shall restrict the ability to [selection: *determine the behaviour of, disable, enable, modify the behaviour of*] the functions [assignment: *list of functions*] to *Crypto Officer*[49].

[46] [assignment: *list of security management functions to be provided by the TSF*]

[47] [assignment: *authorised identified roles*]

[48] [assignment: *conditions for the different roles*]

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

Refinement: If bypass mode is supported by the TOE then the TSF shall indicate through the Status output interface/port when the TOE is in bypass mode.

Application note 21: The ST writer shall perform the missing operation in the element FMT_MOF.1.1/CO according with the management of security functions behaviour supported by the TOE. The operation should address management functions like these

(1) disabling the encryption TSF when entering the bypass mode,

(2) enabling the encryption TSF when leaving the bypass mode,

(3) temporarily enabling or disabling of cryptographic functions e.g. signature-creation.

Application note 22: If maintenance mode is supported by the TOE than additional requirements shall be described by an iteration of the component FMT_MOF.1/CO because the mode transition into the error mode may be caused automatically by hard errors or by users in the maintenance role. Maintenance should include requirements like these:

(4) temporarily disabling cryptographic operation with keys by destruction of plaintext operational keys and CSP when entering the maintenance mode until other keys are imported or generated,

(5) temporarily disabling cryptographic operation with keys by destruction of maintenance keys and CSP when leaving the maintenance mode until operational keys are imported or generated.

FMT_MTD.1/Admin Management of TSF data

Hierarchical to: No other components.

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

FMT_MTD.1.1/Admin The TSF shall restrict the ability to *create, clear and delete*[50] the *Reference Authentication Data* [51] to *Administrator*[52].

FMT_MTD.1/User Management of TSF data

Hierarchical to: No other components.

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

FMT_MTD.1.1/User The TSF shall restrict the ability to *modify*[53] the *Reference Authentication Data*[54] to *the authorized user for their own Reference Authentication Data*[55].

FMT_MSA.1/Key_Man_1 Management of security attributes

Hierarchical to: No other components.

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

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

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

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

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

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

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/Key_Man_1 The TSF shall enforce the *Key Management SFP* [56] to restrict the ability to *change_default and query*[57] the security attributes *Identity of the key, Key entity, Key type of the key, Key usage type, Key access control rules, Key validity time period, Identity of the key component, Key entity of the key component, Key entry method, Identity of the CSP, CSP usage type, CSP access control rules*[58] to *Crypto Officer* [59].

FMT_MSA.1/Key_Man_2 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/Key_Man_2 The TSF shall enforce the *Key Management SFP* [60] to restrict the ability to *modify or delete*[61] the security attributes *Identity of the key, Key entity of the key, Key type, Key usage type, Key access control rules, Key validity time period, Identity of the key component, Key entity of the key component, Key entry method, Identity of the CSP, CSP usage type, CSP access control rules* [62] to *none* [63].

Application note 23: The ST writer may define additional management of security attributes consistent with this component by iteration.

FMT_MSA.2 Secure security attributes

Hierarchical to: No other components.

Dependencies: ADV_SPM.1 Informal TOE security policy model
[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.

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

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

[58] [assignment: *list of security attributes*]

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

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

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

[62] [assignment: *list of security attributes*]

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

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the *Key Management SFP, Cryptographic Operation SFP and Mode_Trans SFP*[64] to provide *restrictive*[65] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *Crypto Officer* [66] to specify alternative initial values to override the default values when an object or information is created.

5.1.5 TSF protection

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Application note 24: The reliable time stamp is used to enforce the Key validity time period defined for a key according to FDP_ACF.1/Oper.

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 *security attributes of cryptographic keys, key components and CSP*[67] when shared between the TSF and another trusted IT product.

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

(1) the TOE reports about conflicts between the Identity of the key of stored cryptographic keys and cryptographic keys to be imported,

(2) the TOE does not change the security attributes Identity of the key, Key entity of the key, Key type, Key usage type and Key validity time period of keys being imported or exported,

(3) the TOE reports about conflicts between the Identity of cryptographic key components of stored key components and cryptographic key components to be imported,

(4) the TOE does not change the security attributes Identity of the key component, Key entity, Key entry method of components keys being imported,

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

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

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

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

(5) *the TOE reports about conflicts between the Identity of the CSP of stored CSP and CSP to imported,*

(6) *the TOE does not change the security attributes Identity of the CSP and CSP usage type of CSP being imported or exported* [68]

when interpreting the TSF data from another trusted IT product.

FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

Dependencies: ADV_SPM.1 Informal TOE security policy model

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: *self test fails*[69].

Refinement:

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

FPT_EMSEC.1 TOE Emanation

Hierarchical to: No other components.

Dependencies: No other components.

FPT_EMSEC.1.1 The TOE shall not emit [assignment: *types of emissions*] in excess of [assignment: *specified limits*] enabling access to

(1) confidential authentication data,

(2) [assignment: list of types of other TSF data][70]

and

(1) "red data" containing confidential information,

(2) plaintext cryptographic secret or private key,

(3) cryptographic key components,

(4) confidential CSP,

(5) [assignment: list of types of other user data][71].

FPT_EMSEC.1.2 The TSF shall ensure [assignment: *type of users*] are unable to use *any interfaces or port*[72] to gain access to

(1) confidential authentication data (except the authentication interface/port during authentication process of the user),

(2) [assignment: list of types of other TSF data][73]

and

(1) "red data" containing confidential information (except the red data input and output interface/port),

(2) plaintext cryptographic secret or private key,

(3) cryptographic key components (except key interface during import of the

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

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

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

[71] [assignment: *list of types of user data*]

cryptographic key component),

(4) *confidential CSP (except key interface during import of the confidential CSP),*

(5) *[assignment: list of types of other user data]*[74].

Application note 25: The ST writer shall perform the missing operation in the elements FPT_EMSEC.1.1 and FPT_EMSEC.1.2. The types of emanation shall include all forms of side channels (power consumption, electromagnetic emanation, and timing) emitted by the TOE that may contain confidential information about the listed assets. The limits shall be specified to prevent attacks through analysis of the emissions in the intended operational environment. The exceptions in element FPT_EMSEC.1.2 comprise the intended use of these interfaces respective ports. E.g. the data input and output interfaces for "red data" contain the confidential information about confidential red data but the data input and output interfaces for "black data" must not contain confidential information about the corresponding "red data" for user not knowing the decryption key. They must not provide information about the plaintext of any cryptographic secret or private key without any exception of interfaces or ports (e.g. through a side channel analysis). If a cryptographic secret or private key is exported in encrypted form the information about the plaintext depends on knowledge of the key encryption key.

FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

Refinement: The TOE shall separate physically or logically the interfaces for red user data, black user data, CSP (including plaintext cryptographic keys and cryptographic key components) and administrative functions. The data output shall be disabled while performing (i) key generation and manual key entry for the communication through this data port, (ii) self-tests, (iii) software loading and zeroization.

FPT_TST.1 TSF testing

Hierarchical to: No other components.

Dependencies: FPT_AMT.1 Abstract machine testing

[72] [assignment: *types of interfaces/ports*]

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

[74] [assignment: *list of types of user data*]

FPT_TST.1.1 The TSF shall run a suite of self tests [selection: *during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self test should occur]*] to demonstrate the correct operation of [selection: *[assignment: parts of TSF], the TSF*][75].

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of [selection: *[assignment: parts of TSF], TSF data*].

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

FPT_TST.2 TSF self-testing

Hierarchical to: No other components.

Dependencies: FPT_FLS.1 Failure with preservation of secure state.

FPT_TST.2.1 The TSF shall perform self-testing at power-up to verify the correctness of [assignment: *list of cryptographic algorithms*] and of [assignment: *list of critical TSF*], and to verify the integrity of the TSF-software/firmware.

FPT_TST.2.2 The TSF shall perform self-testing at the conditions [assignment: *list of conditions*] to verify the correctness of [assignment: *list of critical cryptographic algorithms*].

FPT_TST.2.3 The TSF shall perform self-testing at the conditions [assignment: *list of conditions*] to verify the correctness of [assignment: *list of critical TSF*], and to verify the integrity of [assignment: *list of TSF data*].

FPT_TST.2.4 The TSF shall perform self-testing at the conditions [assignment: *list of conditions*] to verify the integrity of [assignment: *list of TSF-objects*].

FPT_TST.2.5 The TSF shall provide [assignment: *list of users*] with the capability to invoke the following self-tests [assignment: *list of self-tests*].

FPT_TST.2.6 During *initial start-up self-test, power-up self-test, self-test at the request of the authorised user [assignment: other self-tests]*[76] the TSF shall *inhibit all output via the data interfaces/ports, and [assignment: list of additional actions to be performed]*[77].

FPT_TST.2.7 After completion of self-testing the TSF shall *output the results of the self-tests via the status output interface/port, and [assignment: list of additional actions to be performed]*[78].

FPT_TST.2.8 If the self-testing result is fail the TSF shall *enter a secure state (see FPT_FLS.1) and output an error indicator via the status output interface/port, and [assignment: list of additional actions to be performed]*[79].

Refinement:

[75] Note the misprint in the element FPT_TST.1.1 [2]. The text here fits to the Final interpretation 056.

[76] [assignment: *list of self-tests*]

[77] [assignment: *list of actions to be performed*]

[78] [assignment: *list of actions to be performed*]

[79] [assignment: *list of actions to be performed*]

A *start-up test* shall be performed when the TOE is powered up (after being powered off) or on reset. A *List of cryptographic algorithms* shall include all Endorsed cryptographic algorithms employed by the TOE.

In order to *verify the correctness* of cryptographic algorithms self-testing shall perform a known answer or a pair-wise consistency test. If the TOE module includes two independent implementations of the same cryptographic algorithm, then the outputs of two implementations shall be compared.

In order to *verify the integrity of the TSF-software/firmware* a self-testing using an Endorsed error detection code (EDC) or Endorsed authentication technique shall be applied.

The *self-testing at the conditions* shall cover, if applicable, the following conditions: i) when a critical cryptographic algorithm or critical TSF operation is invoked, ii) pair-wise consistency test for newly generated asymmetric key-pairs, iii) on software/firmware load test, iv) on manual key entry, and v) and on bypass events.

If the TOE provides *generation of public/private key pairs*, then the following pair-wise consistency tests for public and private keys shall be performed. If the keys are used to perform an Endorsed key transport method, then the public key shall encrypt a plaintext value. The resulting ciphertext value shall be compared to the original plaintext value. If the two values are equal, then the test shall fail. If the two values differ, then the private key shall be used to decrypt the ciphertext and the resulting value shall be compared to the original plaintext value. If the two values are not equal, the test shall fail. If the keys are used to perform the calculation and verification of digital signatures, then the consistency of the keys shall be tested by the calculation and verification of a digital signature. If the digital signature cannot be verified, the test shall fail.

If *manual import of cryptographic keys or key components* into the TOE is supported, then the following manual key entry tests shall be performed. The cryptographic key or key components shall have an Endorsed EDC applied, or shall be entered using duplicate entries. If the EDC cannot be verified, or the duplicate entries do not match, the test shall fail.

If *load of software or firmware into the TOE* is supported, then the following software/firmware load tests shall be performed. An Endorsed authentication technique shall be applied to all validated software and firmware components when the components are externally loaded into the TOE. The calculated result shall be compared with a previously generated result. If the calculated result does not equal the previously generated result, the software/firmware integrity test shall fail.

If the TOE implements a *bypass capability* where the services may be provided without cryptographic processing, then the following bypass tests shall be performed to ensure that a single point of failure of TOE components will not result in the unintentional output of plaintext. The TSF shall test for the correct operation of the services providing cryptographic processing when a switch takes place between an exclusive bypass service and an exclusive cryptographic service. If the TOE can automatically alternate between a bypass service and a cryptographic service, providing some services with cryptographic processing and some services without cryptographic processing, then the TSF shall test for the correct operation of the services providing cryptographic processing when the mechanism governing the switching procedure is modified.

(End of refinement.)

Application note 26: A cryptographic algorithm shall have an independent known-answer self-test or the known-answer self-test shall be included with the associated cryptographic algorithm self-test. If the calculated output does not equal the known answer, the known-answer self-test shall fail. If a known-answer self-test is not appropriate because the output of the cryptographic algorithms vary for a given set of inputs (e.g., a digital signature generated by means of the Digital Signature Algorithm [7]) it shall be tested using a known-answer test or using the inverse cryptographic function (e.g., a digital signature is verified). Random number generators shall implement statistical or other appropriate tests.

FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1 The TSF shall resist *physical manipulation and probing*[80] to the *TSF*[81] by responding automatically such that the TSP is not violated.

Refinement:

If the TOE is a single-chip cryptographic module the TOE shall resist physical manipulation and probing at any time. If the TOE is a multiple-chip cryptographic module the TOE shall contain tamper response circuitry, which shall immediately destruct all plaintext secret and private keys and CSPs upon the detection of physical tampering.

Application note 27:

The TOE should implement specific security mechanisms to resist the physical tampering scenarios with low attack potential.

For single-chip TOE the supporting documents for smart cards and similar devices apply to the TOE for resistance physical to physical attacks. The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, "automatic response" means here (i) assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.

The requirement for automated response of the TOE is fulfilled due to physical protection mechanisms of the TOE if a physical tampering attack causes serious damage to the TOE such that the TOE will not function and will not compromise any internal secret (i.e. confidential CSP like secret or private cryptographic keys or confidential TSF data).

If the TOE contains circuitry for implementing physical attack response (e.g. destruction of keys), than this circuitry shall remain operational as long as plaintext cryptographic keys, cryptographic key components and CPSs are contained within the TOE. A tamper detection envelope may be e.g., a flexible mylar printed circuit with a serpentine geometric pattern of conductors.

(End of Application note.)

[80] [assignment: *physical tampering scenarios*]

[81] [assignment: *list of TSF devices/elements*]

5.2 Security Assurance Requirements for the TOE

EAL4. Package assurance requirements are:

Development activities (Class ADV)
- Functional Specification (Component ADV_FSP.2 Fully defined external interfaces)
- Informal TOE security policy model (Component ADV_SPM.1)
- High-Level Design (Component ADV_HLD.2)
- Low-Level Design (Component ADV_LLD.1)
- Implementation Representation (Component ADV_IMP.1)
- Representation Correspondence (Component ADV_RCR.1)

Tests activities (Class ATE)
- Coverage (Component ATE_COV.2)
- Depth (Component ATE_DPT.1)
- Functional Tests (Component ATE_FUN.1)
- Independent Testing (Component ATE_IND.2)

Delivery and operation activities (Class ADO)
- Delivery (Component ADO_DEL.2)
- Installation, generation, and start-up (Component ADO_IGS.1)

Guidance documents activities (Class AGD)
- Administrator Guidance (Component AGD_ADM.1)
- User guidance (Component AGD_USR.1)

Configuration management activities (Class ACM)
- CM automation (Component ACM_AUT.1)
- CM Capabilities (Component ACM_CAP.4)
- CM Scope (Component ACM_SCP.2)

Life cycle support activities (Class ALC)
- Development Security (Component ALC_DVS.1)
- Life Cycle Definition (Component ALC_LCD.1)
- Tools and Techniques (Component ALC_TAT.1)

Vulnerability assessment activities (Class AVA)
- Validation of analysis (Component AVA_MSU.2)
- Strength of TOE Security Functions (Component AVA_SOF.1)
- Vulnerability Analysis (Component AVA_VLA.2)

The minimum strength of function is SOF-high. This protection profile contains security functional requirement for which an explicit strength of function claim is required: FIA_UAU.1 and FCS_RNG.1.

5.2.1 Development activities (Class ADV)

5.2.1.1 ADV_FSP.2 Fully defined external interfaces

Dependencies: ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_FSP.2.1D The developer shall provide a functional specification.

Content and presentation of evidence elements:

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

Evaluator action elements:

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

Refinement:

The *functional specification* shall *describe of all details of all effects.* It shall also specify as minimum the normal voltage and temperature operating ranges of the cryptographic module.

The *functional specification* shall describe the interface indicating the selection of an Endorsed mode of operation and the interfaces for user data and TSF data as Endorsed modes of operation.

The *functional specification* shall identify the logical interfaces and physical ports as of the following types ("input" and "output" are indicated from the perspective of the module):

- **Data input interface/port: All data (except control data entered via the control input interface) that is input to and processed by the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and status information from another entities),**
- **Data output interface/port: All data (except status data output via the status output interface) that is output from the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and control information for another entity). All data output via the data output interface shall be inhibited when the TOE is in an error mode or in start-up (power on) self-test mode,**
- **Control input interface/port: All input commands, signals, and control data (including function calls and manual controls such as switches, buttons, and keyboards) used to control the operation of a cryptographic module shall enter via the "control input" interface.**
- **Status output interface/port: All input commands, signals, and control data (including calls and manual controls such as switches, buttons, and keyboards) used to control the operation of the cryptographic module),**
- **Power interface/port: all external electrical power supply.**

Application note 28: Note the TOE shall separate logically the interfaces for red user data, black user data, CSP (including plaintext cryptographic keys and cryptographic key components) and administrative functions according to refinement of FPT_SEP.1. The functional specification shall describe this logical separation according to ADV_FSP.2.3C, ADV_FSP.2.4C and ADV_FSP.2.5C.

5.2.1.2 High-Level Design (Component ADV_HLD.2)

ADV_HLD.2 Security enforcing high-level design

Dependencies: ADV_FSP.1 Informal functional specification ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

ADV_HLD.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

Refinement

The *high level design* shall identify the subsystem with the interface providing the physical port for the import of secret key, private keys and key component. This subsystem and all subsystem which transfer or store any secret key, private keys and key module shall be TSP-enforcing.

5.2.1.3 Low-Level Design (Component ADV_LLD.1)

ADV_LLD.1 Descriptive low-level design

Dependencies: ADV_HLD.2 Security enforcing high-level design
ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

Content and presentation of evidence elements:

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV_LLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_LLD.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

Refinement to ADV_LLD.1.6C:

The *low-level design* shall specify the key storage methods employed by the TOE.

The *low-level design* shall specify methods to destruct all plaintext secret and private cryptographic keys, key components and CSPs within the module.

The *low-level design* shall identify the modules with the interface providing the physical port for the import of secret key, private keys and key component. This module and all modules which transfer or store any secret key, private keys and key components shall be TSP-enforcing.

The *low-level design* shall describe the ventilation physical design approach if applicable. This description shall demonstrate that if the TOE (hardware) contains ventilation holes or slits, then the holes or slits shall be constructed in a manner that prevents undetected physical probing inside the enclosure.

The *low-level design* shall describe the physical enclosure of the TOE. This description shall demonstrate that the enclosure is production grade. The demonstration must either

show that an enclosure of the same material has been used commercially, or provide data to show that it is equivalent to a commercial product.

Application note 29: Construction may prevent undetected physical probing inside the enclosure by means e.g., require at least one 90 degree bend or obstruction with a substantial blocking material.

5.2.1.4 Implementation Representation (Component ADV_IMP.1)

ADV_IMP.1 Subset of the implementation of the TSF

Dependencies: ADV_LLD.1 Descriptive low-level design
ADV_RCR.1 Informal correspondence demonstration
ALC_TAT.1 Well-defined development tools

Developer action elements:

ADV_IMP.1.1D The developer shall provide the implementation representation for the entire TSF.

Content and presentation of evidence elements:

ADV_IMP.1.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be internally consistent.

Evaluator action elements:

ADV_IMP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_IMP.1.2E The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

Refinement:

The *implementation representation* for all software and firmware of the TOE shall be done in a high-level language. The exceptional limited usage of low-level language (e.g., assembly language or microcode) is allowed if essential to the performance of the TOE or when a high-level language is not available. The *implementation representation* for all hardware components of the TOE within the cryptographic module shall be done in a high-level specification language. The source code of implementation representation for each hardware, software, and firmware component (of the TOE) shall be annotated with comments that specify the preconditions required upon entry into the component (of the TOE), function, or procedure in order to execute correctly and the post-conditions expected to be true when execution of the component (of the TOE), function, or procedure is complete.

5.2.1.5 Informal TOE security policy model (Component ADV_SPM.1)

ADV_SPM.1 Informal TOE security policy model

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements:

ADV_SPM.1.1C The TSP model shall be informal**.**

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

Evaluator action elements:

ADV_SPM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement for ADV_SPM.1.2C:

1. **The security policy model shall contain a Finite state model.**
2. **The Finite state model of the TOE shall describe at least the following modes**
   (1) **Power on/off modes**
   (2) **Crypto officer modes**
   (3) **Key/CSP entry modes**
   (4) **User modes**
   (5) **Self-test modes**
   (6) **Error modes**
   (7) **Bypass modes if exist any**
   (8) **Maintenance modes if TOE provides maintenance functionality.**
3. **The Finite state model of the TOE shall describe the mode transition in terms of the input and internal events and internal conditions that cause transitions from one mode to another and the output events resulting from transitions from one mode to another.**
4. **If bypass mode exist the Finite state model shall show, that for all transitions into any bypass mode, two independent internal actions are required for the transition into each bypass mode.**
5. **If maintenance mode exists the mode transition entering or exiting maintenance mode shall destruct all plaintext secret and private keys and unprotected CSPs.**

Refinement for Evaluator action element ADV_SPM.1.1E:
The evaluator shall confirm that the finite state model is consistent with the TSF presentation in the functional specification, low level design, TSF implementation, guidance documentation and evaluator tests.

Application note 30: The term "finite state model" describes finite set of states related to the modes of operation of the cryptographic module, and the state transition in the model in terms of internal actions and conditions for changing the modes of operation of the cryptographic module. The term "mode" for the states in the model is used according to the mode addressed in FDP_ACC.2/Mode_Trans and FDP_ACF.1/Mode_Trans.

5.2.1.6 Representation Correspondence (Component ADV_RCR.1)

ADV_RCR.1 Informal correspondence demonstration

Dependencies: No dependencies.

Developer action elements:

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2 Test activities (Class ATE)

5.2.2.1 Coverage (Component ATE_COV.2)

ATE_COV.2 Analysis of coverage

Dependencies: ADV_FSP.1 Informal functional specification
ATE_FUN.1 Functional testing

Developer action elements:

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

Content and presentation of evidence elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

Evaluator action elements:

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 Depth (Component ATE_DPT.1)

ATE_DPT.1 Testing: high-level design

Dependencies: ADV_HLD.1 Descriptive high-level design
ATE_FUN.1 Functional testing

Developer action elements:

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements:

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

Evaluator action elements:

ATE_DPT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.3 Functional Tests (Component ATE_FUN.1)

ATE_FUN.1 Functional testing

Dependencies: No dependencies.

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.4 Independent Testing (Component ATE_IND.2)

ATE_IND.2 Independent testing - sample

Dependencies: ADV_FSP.1 Informal functional specification
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance
ATE_FUN.1 Functional testing

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.2.3 Delivery and operation activities (Class ADO)

5.2.3.1 Delivery (Component ADO_DEL.2)

ADO_DEL.2 Detection of modification

Dependencies: ACM_CAP.3 Authorisation controls

Developer action elements:

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Evaluator action elements:

ADO_DEL.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 Installation, generation, and start-up (Component ADO_IGS.1)

ADO_IGS.1 Installation, generation, and start-up procedures

Dependencies: AGD_ADM.1 Administrator guidance

Developer action elements:

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

Evaluator action elements:

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.2.4 Guidance documents activities (Class AGD)

5.2.4.1 Administrator Guidance (Component AGD_ADM.1)

AGD_ADM.1 Administrator guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

Content and presentation of evidence elements:

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4.2 User guidance (Component AGD_USR.1)

AGD_USR.1 User guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement:

The guidance documentation shall describe how the user is able to determine when an Endorsed mode of operation is selected and what the current status of the cryptographic module is.

The guidance documentation shall describe how the user is able to initiate and is informed about the result of the self-tests as specified in FPT_TST.1.

5.2.5 Configuration management activities (Class ACM)

5.2.5.1 CM automation (Component ACM_AUT.1)

ACM_AUT.1 Partial CM automation

Dependencies: ACM_CAP.3 Authorisation controls

Developer action elements:

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

Content and presentation of evidence elements:

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

Evaluator action elements:

ACM_AUT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5.2 CM Capabilities (Component ACM_CAP.4)

ACM_CAP.4 Generation support and acceptance procedures

Dependencies: ALC_DVS.1 Identification of security measures

Developer action elements:

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labelled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

ACM_CAP.4.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.6C The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.

ACM_CAP.4.7C The CM system shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.8C The CM plan shall describe how the CM system is used.

ACM_CAP.4.9C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.10C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.11C The CM system shall provide measures such that only authorised changes are made to the configuration items.

ACM_CAP.4.12C The CM system shall support the generation of the TOE.

ACM_CAP.4.13C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

Evaluator action elements:

ACM_CAP.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5.3 CM Scope (Component ACM_SCP.2)

ACM_SCP.2 Problem tracking CM coverage

Dependencies: ACM_CAP.3 Authorisation controls

Developer action elements:

ACM_SCP.2.1D The developer shall provide a list of configuration items for the TOE.

Content and presentation of evidence elements:

ACM_SCP.2.1C The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

Evaluator action elements:

ACM_SCP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement:

The *list of configuration items* shall include a list all physical components of the hardware together with the information about the production-grade of these components.

5.2.6 Life cycle support activities (Class ALC)

5.2.6.1 Development Security (Component ALC_DVS.1)

ALC_DVS.1 Identification of security measures

Dependencies: No dependencies.

Developer action elements:

ALC_DVS.1.1D The developer shall produce development security documentation.

Content and presentation of evidence elements:

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements:

ALC_DVS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

5.2.6.2 Life Cycle Definition (Component ALC_LCD.1)

ALC_LCD.1 Developer defined life-cycle model

Dependencies: No dependencies.

Developer action elements:

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

Content and presentation of evidence elements:

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

Evaluator action elements:

ALC_LCD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6.3 Tools and Techniques (Component ALC_TAT.1)

ALC_TAT.1 Well-defined development tools

Dependencies: ADV_IMP.1 Subset of the implementation of the TSF

Developer action elements:

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

Content and presentation of evidence elements:

ALC_TAT.1.1C All development tools used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

Evaluator action elements:

ALC_TAT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.7 Vulnerability assessment activities (Class AVA)

5.2.7.1 Misuse (Component AVA_MSU.2)

AVA_MSU.2 Validation of analysis

Dependencies: ADO_IGS.1 Installation, generation, and start-up procedures

ADV_FSP.1 Informal functional specification

AGD_ADM.1 Administrator guidance

AGD_USR.1 User guidance

Developer action elements:

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

Content and presentation of evidence elements:

AVA_MSU.2.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

AVA_MSU.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.2.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

5.2.7.2 Strength of TOE Security Functions (Component AVA_SOF.1)

AVA_SOF.1 Strength of TOE security function evaluation

Dependencies: ADV_FSP.1 Informal functional specification

ADV_HLD.1 Descriptive high-level design

Developer action elements:

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

5.2.7.3 Vulnerability Analysis (Component AVA_VLA.2)

AVA_VLA.2 Independent vulnerability analysis

Dependencies: ADV_FSP.1 Informal functional specification
ADV_HLD.2 Security enforcing high-level design
ADV_IMP.1 Subset of the implementation of the TSF
ADV_LLD.1 Descriptive low-level design
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance

Developer action elements:

AVA_VLA.2.1D The developer shall perform a vulnerability analysis.

AVA_VLA.2.2D The developer shall provide vulnerability analysis documentation.

Content and presentation of evidence elements:

AVA_VLA.2.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.2.2C The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.2.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.2.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

Evaluator action elements:

AVA_VLA.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.2.2E The evaluator *shall conduct* penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.2.3E The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.2.4E The evaluator *shall perform* independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.2.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low attack potential.

Refinement

The *vulnerability analysis* shall specify and analyse all security-related information, including secret and private cryptographic keys (both plaintext and encrypted), key components, CSPs, authentication data, and other protected information whose disclosure or modification can compromise the security of the TOE.

5.3 Security Requirements for the IT environment

This protection profile does not describe security requirements for the IT environment.

6 PP Application Notes

The application notes are included in the text above.

7 Rationales

7.1 Security Objectives Rationale

The following table provides an overview for security objectives coverage.

	OSP.User_Data_Prot	OSP.Resist_Low	OSP.I&A	OSP.Access	OSP.Roles	OSP.Endorsed_Crypto	OSP.Key_Man	OSP.Key_Personal	T. Compro_CSP	T.Modif_CSP	T.Abuse_Func	T.Physical_Tamper	T.Inf_Leakge	T.Malfunction	T.Masquerade	A.User_Data	A.Data_Sep	A.Key_Generation	A.Availability
O.I&A		x	x					x							x				
O.Control_Services				x							x								
O.Control_Keys				x				x	x	x					x				
O.Roles				x	x						x								
O.Key_Export							x	x	x	x									
O.Key_Generation							x		x										
O.Key_Import							x	x	x	x									
O.Key_Management							x	x	x	x									
O.Key_Destruction							x		x										
O.Red-Black-Sep									x						x				
O.Check_Operation										x				x					
O.Endorsed_Crypto	x					x													
O.Physical_Protect		x										x							
O.Prevent_Inf_Leakage		x							x				x						
OE.Assurance		x																	
OE.Key_Generation							x		x									x	
OE.Red-Black-Sep																x	x		
OE.Personal					x			x			x								
OE.Key_Availabilty	x																		x

Table 1: Security Objective Rationale

The organisational security policy **OSP.User_Data_Prot** “Protection of user data by cryptographic functions” addresses the protection of the confidentiality or integrity or both of information represented by user data of the IT-system to be provided by the cryptographic module and the protection of availability of user data by the IT system. The security objective O.Endorsed_Crypto ensures that TOE provides Endorsed cryptographic functions to protect the user data as required by OSP.User_Data_Prot. The security objective for the IT environment OE.Key_Availabilty ensures that IT system protects the availability of the user data and the cryptographic keys outside the cryptographic module.

The organisational security policy **OSP.Resist_Low** “Resistance against low attack potential” requires the TOE to resist attacks with low attack potential. This is ensured by the security objective for the development environment OE.Assurance (cf. to last sentence). The security objectives O.I&A, O.Physical_Protect and O.Prevent_Inf_Leakage address directly the resistance against attacks with low attack potential.

The organisational security policy **OSP.I&A** “Identification and authentication of users” addresses identification and authentication of all users prior to accessing any controlled resources with the exception of read access to public objects and cryptographic operations with public keys. This is directly ensured by the security objective O.I&A.

The organisational security policy **OSP.Access** “Access control of TOE functions” addresses the limitation of the extent of each user’s abilities to use the TOE functions in accordance with the TSP. The security objective O.Control_Services requires that the TOE shall restrict the access to its services, depending on the user role, to those services explicitly assigned to this role which are provided according to the security objective O.Roles. The security objective O.Control_Keys limits user’s ability to use the TOE functions to ensure the cryptographic security as part of the TSP.

The organisational security policy **OSP.Roles** “Roles” addresses separate and distinct roles for authorized administrator, cryptographic administrator and end-users. The security objective O.Roles requires the TOE to implement them and the security objective OE.Personal requires the IT environment to use them.

The organisational security policy **OSP.Endorsed_Crypto** “Endorsed cryptographic functions” addresses the implementation of Endorsed cryptographic algorithms and Endorsed cryptographic protocols for the protection of the confidentiality or the integrity or both of the user data according to the organizational security policy OSP.User_Data_Prot and for the key management. This is ensured generally by the security objective O.Endorsed_Crypto.

The security objective **OSP.Key_Man** “Cryptographic key management” requires to manage and use the cryptographic keys as they are assigned to the entities, cryptographic algorithms and protocols. This OSP is implemented generally by the security objectives for the TOE O.Key_Management for secure key management and specifically for critical processes over the key life cycle by the security objectives O.Key_Generation, O.Key_Import, O.Key_Export and O.Key_Destruction. OE.Key_Generation ensures the cryptographic strength, the confidentiality and integrity of secret and private keys, the integrity and authenticity of public keys and correct security attributes if they are generated outside the TOE and imported into the TOE.

The organisational security policy **OSP.Key_Personal** “Personal security for cryptographic keys” addresses key management in a way that the integrity and confidentiality of key can not be compromised by a single person. This OSP is implemented generally by the security objectives O.Key_Management and O.Control_Keys for secure key management and use. Furthermore for critical processes, the security objectives O.Key_Import, O.Key_Export and O.Control_Keys enforce secure key import, key export and key usage. O.I&A ensures that the

TOE uniquely identifies users and verifies the claimed identity of the user before providing access. OE.Personal requires assignment of roles to distinct authorized persons and that for manual key import at least two different authorized persons are assigned to cryptographic administrator role.

The threat **T.Compro_CSP** “Compromise of CSP” addresses the compromise of confidential CSP which enables attacks against the confidentiality or integrity of user data and TSF data protected by these CSPs. The security objective O.Control_Keys requires the TOE to restrict the access to the keys, key components and CSP according to their security attributes. The security objective O.Key_Management ensures these security attributes are managed securely. The security objective O.Key_Export and O.Key_Import require the protection of secret or private keys in encrypted form or using split knowledge procedures for their export and import. The security objective O.Key_Generation requires the TOE and the OE.Key_Generation requires the environment to generate cryptographic strong keys. O.Key_Destruction requires the secure destruction on demand of user. The security objective O.Red-Black-Sep requires protecting the confidentiality of CSP by logical separation of interfaces for CSP from other interfaces. The security objective O.Prevent_Inf_Leakage requires the TOE to prevent information leakage about secret and private keys and confidential TSF data outside the cryptographic boundary.

The threat **T.Modif_CSP** “Modification of integrity sensitive CSP” addresses the modification of the integrity sensitive CSP which enables attacks against the confidentiality or integrity of user data or the TSF protected by these CSPs . The security objective O.Control_Keys requires the TOE to restrict the access to the keys, key components and CSP according to their security attributes. The security objective O.Key_Management ensures these security attributes are managed securely. The security objective O.Key_Export and O.Key_Import require the protection of the integrity keys during their export and import. The security objective O.Check_Operation requires verification the integrity of CSP.

The threat **T.Abuse_Func** “Abuse of function” addresses the misuse of TOE functions intended for installation, configuration or maintenance which shall not be used for operational cryptographic keys or user data. This is ensured by the security objective O.Control_Services that restricts the access to TOE services, depending on the user role, to those services explicitly assigned to this role. The security objective O.Roles requires the TOE to provide at least the Administrator, the Cryptographic Administrator, the End-user roles, and Maintenance Personal if the TOE supports maintenance functionality. The Administrator, Cryptographic Administrator, End-user roles and Maintenance Personal if the TOE supports maintenance functionality, will be assigned to authorized distinct persons according to the security objective for the IT environment OE.Personal.

The threat **T.Inf_Leakage** “Information leakage” describes that an attacker may observe and analyse any energy consumed or emitted through the cryptographic boundary (i.e. including the external interfaces) of the TOE to get internal secrets or confidential user data not intended for export. The protection against this threat is directly required by the security objective O.Prevent_Inf_Leakage.

The threat **T.Malfunction** “Malfunction of TSF” describes the use of a malfunction of the hardware or software in order to deactivate, modify, or circumvent security functions of the TOE to enable attacks against the integrity or confidentiality of the User data or the CSP. The security objective O.Check_Operation prevents this threat by regular checks verifying that TOE components operate correctly.

The threat **T.Physical_Tamper** “Physical tampering” describes tampering the cryptographic module to get secrets, to modify data on whose integrity the TSF relies, or to corrupt or deactivate the TSF inside the cryptographic boundary, which is directly addressed by the security objective O.Physical_Protect.

The threat **T.Masquerade** “Masquerade authorized data source or receiver” describes that an attacker may masquerade as an authorized data source or receiver to perform operations that will be attributed to the authorized user or gains undetected access to cryptographic module causing potential violations of integrity, or confidentiality. The security objective O.I&A requires the TOE to identify and authenticate the user before providing access to any controlled resources with the exception of public objects. The security objective O.I&A requires the security functions for authentication of users to have strength “high” to cover attacks with high attack potential as described in T.Masquerade. The security objective O.Control_Keys restricts the access to the keys, key components and other CSP according to their security attributes (including Key entity). Furthermore the security objective O.Red-Black-Sep requires the TOE to protect integrity sensitive information by verification of black data for import into the red area.

The assumptions **A.User_Data** “Protection of user data by the IT system” and **A.Data_Sep** “Separation of cryptographically protected and unprotected data “ are covered by the security objective for the IT environment OE.Red-Black-Sep “Separation of red and black area of the IT system” dealing with protection of the user data in the red area of the IT system, their security attributes for cryptographic protection to the TOE and the control the exchange data between the red and black area of the IT system according to the IT security policy.

The assumption **A.Key_Generation** “Key generation and import to the cryptographic module” deals with the cryptographic strength and secure security attributes of cryptographic keys generated by the IT environment and imported into the TOE. This assumption is directly and completely covered by the security objective for the IT environment OE.Key_Generation.

The assumption **A.Availability** “Availability of keys” describes that the IT environment ensures the availability of cryptographic keys and key material as ensured by the security objective for the IT environment OE.Key_Availabilty.

7.2 Security Requirements Rationale

7.2.1 Security Functional Requirements Rationale

The following table provides an overview on how the TOE security functional requirements cover the TOE security objectives.

	O.Red-Black-Sep	O.Endorsed_Crypto	O.I&A	O.Control_Services	O.Control_Keys	O.Roles	O.Key_Export	O.Key_Generation	O.Key_Import	O.Key_Management	O.Key_Destruction	O.Check_Operation	O.Physical_Protect	O.Prevent_Inf_Leakage
FCS_CKM.1		x						x		x				
FCS_CKM.2/Import		x							x	x				
FCS_CKM.2/Export		x					x			x				
FCS_CKM.4		x								x	x			
FTP_ITC.1							x		x	x				
FCS_COP.1	x	x												
FCS_RNG.1		x						x						
FIA_ATD.1			x											
FIA_UID.1			x											
FIA_UAU.1			x											
FIA_UAU.6			x											
FIA_UAU.7			x											
FIA_USB.1			x											
FIA_AFL.1			x											
FDP_ACC.2/Key_Man				x	x					x				
FDP_ACF.1/Key_Man				x	x					x	x			
FDP_ACC.2/Oper				x	x									
FDP_ACF.1/Oper				x	x									
FDP_ACC.2/Mode_Trans				x										
FDP_ACF.1/Mode_Trans				x	x									
FDP_ITC.2		x							x	x				
FDP_ETC.2		x					x			x				
FDP_UCT.1	x						x		x	x				
FDP_UIT.1	x						x		x	x				
FDP_RIP.2														x
FPT_STM.1					x									
FMT_SMF.1				x						x				
FMT_SMR.2				x		x				x				
FMT_MOF.1/CO				x										
FMT_MTD.1/Admin			x											
FMT_MTD.1/User			x											
FMT_MSA.1/Key_Man_1				x						x				
FMT_MSA.1/Key_Man_2				x						x				
FMT_MSA.2				x	x					x				

	O.Red-Black-Sep	O.Endorsed_Crypto	O.I&A	O.Control_Services	O.Control_Keys	O.Roles	O.Key_Export	O.Key_Generation	O.Key_Import	O.Key_Management	O.Key_Destruction	O.Check_Operation	O.Physical_Protect	O.Prevent_Inf_Leakage
FMT_MSA.3				x				x		x				
FPT_TDC.1							x		x	x				
FPT_FLS.1												x		
FPT_EMSEC. 1	x													x
FPT_PHP.3													x	
FPT_RVM.1	x													x
FPT_SEP.1	x													
FPT_TST.1												x		
FPT_TST.2												x		

Table 2: Coverage of Security Objective for the TOE by SFR

The security objective **O.Red-Black-Sep** “Red-black separation of the TOE” is provided by the following SFR:

- FCS_COP.1 requires the necessary cryptographic operations needed for encryption, decryption of data containing confidential information and integrity protection for data containing integrity sensitive information.
- FDP_UCT.1 addresses the protection of the data containing confidential information during data exchange.
- FDP_UIT.1 addresses the protection of the data containing integrity sensitive information during data exchange.
- FPT_SEP.1 requires the separation of logical and physical interfaces.
- FPT_EMSEC.1 requires protection of confidential information against emanation.
- FPT_RVM.1 requires that TSF are invoked and succeed before each function within the TSC is allowed to proceed.

The security objective **O.Endorsed_Crypto “**Endorsed cryptographic functions“ requires the TOE to provide Endorsed cryptographic functions and Endorsed cryptographic protocols to protect the user data as required by OSP.User_Data_Prot and for key management. This security objective is provided by the SFR FCS_CKM.1, FCS_CKM.2/Import, FCS_CKM.2/Export, FCS_CKM.4, FCS_COP.1 and FCS_RNG.1, which require meeting Endorsed standards for cryptographic functions. FDP_ITC.2 and FDP_ETC.2 enforce the use of Endorsed cryptographic functions for import and export of confidential cryptographic keys.

The security objective **O.I&A** “Identification and authentication of users” requires the TOE to identify uniquely users and to verify the claimed identity of the user before providing access to any controlled resources with the exception of read access to public objects. This security objective is provided by the following SFR:

- FIA_UID.1 allows unidentified users to run self test of the TOE only and requires identification before any other TSF mediated action.
- FIA_UAU.1 allows unauthenticated users to run self test of the TOE, identification according FIA_UID.1 and selection of a claimed role and requires authentication before any other TSF mediated action.
- FIA_UAU.6 requires re-authentication after start-up of the TOE and if the user changes the role after authentication.
- FIA_UAU.7 requires limitation of the feedback to the user while authentication is in progress.
- FIA_AFL.1 requires detection and reaction to unsuccessful authentication attempts.
- FIA_ATD.1 requires maintaining security attributes to individual users including Identity, Role and Reference authentication data as prerequisite for identification and authentication of authorized users..
- FIA_USB.1 requires associating the identity and the role with the subjects acting for the authenticated user.
- FMT_MTD.1/Admin restricts the creation, clearing and deletion of Authentication Reference Data to the role Administrator.
- FMT_MTD.1/User restricts the ability to modify the Reference authentication data to the user to which this security attribute belongs.

The security objective **O.Roles "**Roles known to TOE" is implemented by the SFR FMT_SMR.2 which requires the TOE to provide at least the Administrator, the Cryptographic Administrator, the End-user roles, *Unidentified User Role, Unauthenticated User Role* and the Maintenance Personal if TOE supports maintenance functionality.

The security objective **O.Control_Services "**Access control for services" requires the TOE to restrict the access to its services, depending on the user role, to those services explicitly assigned to the role. Assignment of services to roles shall be either done by explicit action of an Administrator or by default. This security objective is provided by the following SFR:

- FDP_ACC.2/Key_Man and FDP_ACF.1/Key_Man require access control to the key management services of the TOE,
- FDP_ACC.2/Oper and FDP_ACF.1/Oper require access control to the cryptographic operation services of the TOE,
- FDP_ACC.2/Mode_Trans and FDP_ACF.1/Mode_Trans require access control to the operational modes of the TOE which limit the available services.
- FMT_SMF.1 lists the security management functions including the management of TSF behaviour FMT_MOF.1.
- FMT_SMR.2 describing the minimum list of roles and restrictions to these roles.
- FMT_MOF.1/CO limits the management of TSF behaviour to the users in the Crypto officer role.
- FMT_MSA.1/Key_Man_1 and FMT_MSA.1/Key_Man_2 require limitation to the management of security attributes of cryptographic keys, key components and CSP describing the available services for these objects.
- FMT_MSA.2 and FMT_MSA.3 describe additional requirements to the management of security attributes to enforce the access control SFP for FDP_ACF.1/Key_Man, FDP_ACF.1/Oper and FDP_ACF.1/Mode_Trans.

The security objective **O.Control_Keys** "Access control for cryptographic keys" requires the TOE to restrict the access to the keys, key components and other CSP according to their security attributes. This security objective is provided by the following SFR:

- FDP_ACC.2/Key_Man and FDP_ACF.1/Key_Man require access control to the key keys, key components and other CSP according to their security attributes,
- FDP_ACC.2/Oper and FDP_ACF.1/Oper require access control to the keys and other CSP of the TOE according to their security attributes,
- FMT_MSA.1/Key_Man_1 and FMT_MSA.1/Key_Man_2 require limitation to the management of security attributes of cryptographic keys, cryptographic key components and CSP describing the access rights, available services and properties for these objects.
- FMT_MSA.2 ensures that only secure values for cryptographic keys, key components and CSP are accepted for security attributes.
- FPT_STM.1 requires the TSF to provide reliable time stamp that is necessary for FDP_ACF.1/Oper to enforce the use of cryptographic keys in the limits of the Key validity time period defined as security attribute of this key.

The SFR FDP_ACF.1/Mode_Trans and the refinement to the SAR ADV_SPM.1 ensures that operational keys and CSP can not be used in maintenance mode and maintenance keys and CSP can not be used outside the operational mode to protect user data.

The security objective **O.Key_Management** "Management of cryptographic keys" requires the TOE to manage securely cryptographic keys, cryptographic key components and CSP. This security objective is provided by the following SFR:

- FCS_CKM.1, FCS_CKM.2/Import, FCS_CKM.2/Export, and FCS_CKM.4 provide the Endorsed cryptographic functions used by key management.
- FTP_ITC.1 provides a trusted channel for key import and export.
- FDP_ACC.2/Key_Man and FDP_ACF.1/Key_Man provide the access control to the key management functions.
- FDP_ITC.2 and FDP_ETC.2 ensure the import and export of cryptographic keys, cryptographic key components and CSP with security attribute, which are associated with these objects for key management.
- FDP_UCT.1 and FDP_UIT.1 requires the TSF to ensure confidentiality and integrity of keys exchanged by import and export of user data including cryptographic keys.
- FMT_SMF.1 list the security management functions and FMT_SMR.2 the roles for key management (i.e. the Crypto officer for operational keys and the Maintenance role for maintenance keys).
- FMT_MSA.1/Key_Man_1, FMT_MSA.1/Key_Man_2 FMT_MSA.2 and FMT_MSA.3 describes the management of security attributes of cryptographic keys, cryptographic key components and CSP.
- FPT_TDC.1 ensures the consistency of the security attributes of cryptographic keys, cryptographic key components and CSP.

The security objective **O.Key_Export** "Export of cryptographic keys" requires the TOE to export keys with their security attributes and protected in integrity. This is provided by the following SFR:

- FCS_CKM.2/Export requires the TSF to distribute keys by export methods meeting Endorsed standards and provides a refinement for keys exported for manual import.

- FTP_ITC.1 requires the TSF to provide a trusted channel of key export.
- FDP_ETC.2 requires the TSF to export keys unambiguously associated with their security attributes.
- FDP_UCT.1 requires the ability to protect confidentiality of exchanged user data which includes cryptographic keys.
- FDP_UIT.1 requires the ability to protect integrity of exchanged user data which includes cryptographic keys.
- FPT_TDC.1 requires to ensure inter-TSF basic TSF data consistency for exported security attributes of cryptographic keys, key components and CSP.

The security objective **O.Key_Generation** “Generation of cryptographic keys by the TOE” requires the TOE to generate cryptographic strong keys using Endorsed cryptographic key generation algorithms. This is provided by the SFR FCS_CKM.1 which requires the use of Endorsed key generation algorithms and FCS_RNG.1 describing requirements for the random number generator needed for key generation. The SFR FMT_MSA.3 requires restrictive values of security attributes for cryptographic keys and limits the ability to specify their initial value to the Crypto officer.

The security objective **O.Key_Import** “Import of cryptographic keys” requires the TOE to import keys with security attributes and verify their integrity. The TOE shall import secret or private keys in encrypted form or manually using split knowledge procedures only. This is provided by the following SFR:

- FCS_CKM.2/Import requires the TSF to distribute by key import methods meeting Endorsed standards and provides a refinement for manually imported keys.
- FTP_ITC.1 requires the TSF to provide a trusted channel of key import.
- FDP_UCT.1 requires the ability to protect confidentiality of exchanged user data which includes cryptographic keys.
- FDP_UIT.1 requires the ability to protect integrity of exchanged user data which includes cryptographic keys.
- FDP_ITC.2 requires the TSF to import keys unambiguously associated with their security attributes.
- FPT_TDC.1 requires to ensure inter-TSF basic TSF data consistency for imported security attributes of cryptographic keys, key components and CSP.

The security objective **O.Key_Destruction** “Destruction of cryptographic keys” requires the TOE to destruct keys cryptographic key components and other CSP on demand of authorized users or when they will not be used any more in a secure way that no information about these keys is left in the resources storing or handling these objects before destruction. This is provided by the following SFR:

- FCS_CKM.4 requires the TSF to provide Endorsed mechanisms for key destruction.
- FDP_ACF.1/Key_Man limits key destruction to users in the Crypto officer role.

The security objective **O.Check_Operation** “Check for correct operation” requires the TOE to perform regular checks to verify that its components operate correctly including integrity checks of TOE software, firmware, internal TSF data and keys. This is provided by the SFR:

- FPT_TST.1 and FPT_TST.2 requiring TSF self tests.
- FPT_FLS.1 requires the TSF to preserve a secure state when self-test fails.

The security objective **O.Physical_Protect** “Physical protection” requires the TOE to unambiguously detect physical tampering at the cryptographic boundary and respond automatically such that the TSP is not violated. Upon the detection of tampering, the TOE shall immediately destruct all plaintext secret and private cryptographic keys and CSPs. This is provided by the SFR FPT_PHP.3.

The security objective **O.Prevent_Inf_Leakage** “Prevent leakage of confidential information” requires the TOE to prevent information leakage about secret and private keys and confidential TSF data outside the cryptographic boundary and unintended output confidential user information. This is provided by the following SFR:

- FDP_RIP.2 requires the TOE to ensure that any previous information content of a resource is made unavailable.
- FPT_RVM.1 ensures that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.
- FPT_EMSEC.1 requires to prevent illicit flow of confidential information through any emanation and the “black data” interface

The security objective for the TOE environment OE.Assurance is provided by the security assurance requirements EAL4.
The security objectives for the TOE environment OE.Key_Generation, OE.Red-Black-Sep, OE.Personal and OE.Key_Availabilty will be provided by technical and organisational security measures. There is no need to specify these security measures on the abstract level of this protection profile.

7.2.2 Dependency Rationale

SFR	Dependencies	Support of the Dependencies
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation], FCS_CKM.4 Cryptographic key destruction, FMT_MSA.2 Secure security attributes	FCS_CKM.2/Export, FCS_COP.1, FCS_CKM.4, FMT_MSA.2
FCS_CKM.2/Export	[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, FMT_MSA.2 Secure security attributes	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2
FCS_CKM.2/Import	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or	FDP_ITC.2, FCS_CKM.1, FCS_CKM.4, FMT_MSA.2

SFR	Dependencies	Support of the Dependencies
	FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction, FMT_MSA.2 Secure security attributes	
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FMT_MSA.2 Secure security attributes	FDP_ITC.2, FCS_CKM.1, FMT_MSA.2
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction, FMT_MSA.2 Secure security attributes	FDP_ITC.2, FCS_CKM.1, FCS_CKM.4, FMT_MSA.2
FCS_RNG.1	FPT_TST.1 TSF testing	FPT_TST.1
FDP_ACC.2/Key_Man	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Key_Man
FDP_ACC.2/Mode_Trans	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Mode_Trans
FDP_ACC.2/Oper	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Oper
FDP_ACF.1/Key_Man	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialization	FDP_ACC.2/Key_Man, FMT_MSA.3
FDP_ACF.1/Mode_Trans	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialization	FDP_ACC.2/Mode_Trans, FMT_MSA.3
FDP_ACF.1/Oper	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialization	FDP_ACC.2/Oper, FMT_MSA.3
FDP_ETC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.2/Key_Man, FDP_ACC.2/Oper (hierarchical to

SFR	**Dependencies**	**Support of the Dependencies**
		FDP_ACC.1)
FDP_ITC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path], FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.2/Key_Man, FDP_ACC.2/Oper (hierarchical to FDP_ACC.1), FTP_ITC.1, FPT_TDC.1,
FDP_UCT.1	[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]	FDP_ACC.2/Oper (hierarchical to FDP_ACC.1), FTP_ITC.1
FDP_UIT.1	[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]	FDP_ACC.2/Oper (hierarchical to FDP_ACC.1), FTP_ITC.1
FDP_RIP.2	No dependencies	n. a.
FIA_AFL.1	FIA_UAU.1 Timing of authentication	FIA_UAU.1
FIA_ATD.1	No dependencies	n. a.
FIA_UAU.1	FIA_UID.1 Timing of identification	FIA_UID.1
FIA_UAU.6	No dependencies	n. a.
FIA_UAU.7	FIA_UAU.1 Timing of authentication	FIA_UAU.1
FIA_UID.1	No dependencies	n. a.
FIA_USB.1	FIA_ATD.1 User attribute definition	FIA_ATD.1
FMT_MOF.1/CO	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	FMT_SMR.2 (hierarchical to FMT_SMR.1), FMT_SMF.1
FMT_MSA.1/Key_Man_1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control],	FDP_ACC.2/Key_Man, FDP_ACC.2/Oper (hierarchical to

SFR	**Dependencies**	**Support of the Dependencies**
	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	FDP_ACC.1), FMT_SMR.2 (hierarchical to FMT_SMR.1),, FMT_SMF.1
FMT_MSA.1/Key_Man_2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	FDP_ACC.2/Key_Man, FDP_ACC.2/Oper (hierarchical to FDP_ACC.1) FMT_SMR.2 (hierarchical to FMT_SMR.1),, FMT_SMF.1
FMT_MSA.2	ADV_SPM.1 Informal TOE security policy model, [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	ADV_SPM.1, FDP_ACC.2 (all iterations, hierarchical to FDP_ACC.1), FMT_MSA.1/Key_Man_1, FMT_MSA.1/Key_Man_2, FMT_SMR.2 (hierarchical to FMT_SMR.1),
FMT_MSA.3	FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles	FMT_MSA.1/Key_Man_1, FMT_MSA.1/Key_Man_2, FMT_SMR.2 (hierarchical to FMT_SMR.1),
FMT_MTD.1/Admin	FMT_SMF.1 Specification of management functions, FMT_SMR.1 Security roles	FMT_SMR.2 (hierarchical to FMT_SMR.1), FMT_SMF.1
FMT_MTD.1/User	FMT_SMF.1 Specification of management functions, FMT_SMR.1 Security roles	FMT_SMR.2 (hierarchical to FMT_SMR.1), FMT_SMF.1
FMT_SMF.1	No dependencies	No dependencies
FMT_SMR.2	FIA_UID.1 Timing of identification	FIA_UID.1
FPT_EMSEC. 1	No dependencies	No dependencies
FPT_FLS.1	ADV_SPM.1	ADV_SPM.1
FPT_PHP.3	No dependencies	No dependencies
FPT_RVM.1	No dependencies	No dependencies
FPT_SEP.1	No dependencies	No dependencies

SFR	Dependencies	Support of the Dependencies
FPT_STM.1	No dependencies	No dependencies
FPT_TDC.1	No dependencies	No dependencies
FPT_TST.1	FPT_AMT.1 Abstract machine testing	See justification 1 for non-satisfied dependencies
FPT_TST.2	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1
FTP_ITC.1	No dependencies	No dependencies

Table 3: Dependencies between the SFR for the TOE

Justification for non-satisfied dependencies between the SFR for TOE:

No. 1: The TOE is physically defined as a set of hardware and software and/or firmware which is contained within the cryptographic boundary and logically defined by the provided security functions depending on the implemented cryptographic algorithms and protocols. There is no need to perform testing to demonstrate the security assumptions made about the underlying abstract machine upon which the TSF relies.

7.2.3 Security Assurance Requirements Rationale

The EAL4 assurance package was chosen to permit a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest level at which it is likely to be economically feasible to retrofit to an existing product line. EAL4 is applicable in those circumstances where developers or users require a low to moderate level of independently assured security in conventional commodity TOEs and are prepared to incur sensitive security specific engineering costs.

The minimal strength of function “high” was selected to ensure resistance against direct attacks on functions based on probabilistic or permutational mechanisms. The SOF requirement applies to all permutational and probabilistic mechanisms especially those implementing FIA_UAU.1 and FCS_RNG.1. This is consistent with the security objective OE.Assurance.

7.2.4 Security Requirements – Mutual Support and Internal Consistency

The following part of the security requirements rationale shows that the set of security requirements for the TOE consisting of the security functional requirements (SFRs) and the security assurance requirements (SARs) together form a mutually supportive and internally consistent whole.

The analysis of the TOE´s security requirements with regard to their mutual support and internal consistency demonstrates:

The dependency analysis in section 7.2.2 Dependency Rationale for the security functional requirements shows that the basis for mutual support and internal consistency between all defined functional requirements is satisfied. All dependencies between the chosen functional components are analyzed, and non-satisfied dependencies are appropriately explained.

The assurance class EAL4 is an established set of mutually supportive and internally consistent assurance requirements. The dependency analysis for the sensitive assurance components in section 7.2.3 Security Assurance Requirements Rationale shows that the assurance requirements are mutually supportive and internally consistent as all (sensitive) dependencies are satisfied and no inconsistency appears.

Inconsistency between functional and assurance requirements could only arise if there are functional-assurance dependencies which are not met, a possibility which has been shown not to arise in sections 7.2.2 Dependency Rationale and 7.2.3 Security Assurance Requirements Rationale. Furthermore, as also discussed in section 7.2.3 Security Assurance Requirements Rationale, the chosen assurance components are adequate for the functionality of the TOE. So the assurance requirements and security functional requirements support each other and there are no inconsistencies between the goals of these two groups of security requirements.

8 Glossary and Acronyms

8.1 Glossary

Term	PP CM (all security levels)
Administrator	An authorized user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given to them.
Authentication interface/port	Data interface respective port used for input of confidential authentication data.
Authentication keys	General term for keys used for authentication of data (i.e. Data authentication keys) or the identity of an entity (i.e. Entity authentication keys)
Authentication reference keys	Private key for proof of their own identity claimed in an asymmetric authentication protocol
Authentication verification keys	Public Key assigned to a claimed identity of an entity for verification of the knowledge of a private key by means asymmetric authentication protocol
Automated key transport	The transport of cryptographic keys, usually in encrypted form, using electronic means such as a computer network (e.g., key transport/agreement protocols).
Black data	Cryptographically protected user data representing user information. If this information needs protection in confidentiality the data shall be encrypted. If this information needs protection in integrity a cryptographic MAC or digital signature shall be associated with this data to detect modification.
Compromise	The unauthorized disclosure, modification, substitution, or use of sensitive data (including plaintext cryptographic keys and other CSPs).
Confidentiality	the property that sensitive information is not disclosed to unauthorized individuals, entities, or processes.
Control input interface/port	Interface respective port intended for all input commands, signals, and control data (including function calls and manual controls such as switches, buttons, and keyboards) used to control the operation of a cryptographic module shall enter via the "control input" interface.
Critical security parameter (CSP)	Security-related information (e.g., secret and private cryptographic keys, and TSF data like authentication data) whose disclosure or modification can compromise the security of a cryptographic module.
Critical TSF	TSF that, upon failure, could lead to (i) the disclosure of secret keys, private keys, or CSPs or (ii) modification of public root keys. Examples of the critical functionality include but are not limited to

Term	PP CM (all security levels)
	random number generation, operation of the cryptographic algorithm, and cryptographic bypass.
Crypto officer	An authorized user who has been granted the authority to perform cryptographic initialization and management functions (including key management) cryptographically unprotected data in the red area of the IT system. These users are expected to use this authority only in the manner prescribed by the guidance given to them. (The "cryptographic administrator" is some times called "crypto officer" in the guidance documentation) (the same as *Cryptographic administrator)*
Cryptographic algorithm	A well-defined computational procedure that takes variable inputs that usually includes a cryptographic key and produces an output, e.g. encryption, decryption, a private or a public operation in a dynamic authentication, signature creation, signature verification, generation of hash value.
Cryptographic boundary	An explicitly defined continuous perimeter that establishes the physical bounds of a cryptographic module and contains all the hardware, software, and/or firmware components of a cryptographic module.
Cryptographic functions	TSF implementing cryptographic algorithms and/or protocols for - encryption and decryption, - signature creation or verification, - calculation of Message Authentication Code, - entity authentication, - key management.
Cryptographic key (key)	A parameter used in conjunction with a cryptographic algorithm that determines - the transformation of plaintext data into ciphertext data, - the transformation of ciphertext data into plaintext data, - a digital signature computed from data, - the verification of a digital signature computed from data, - a Message Authentication Code computed from data, - a proof of the knowledge of a secret, - a verification of the knowledge of a secret or - an exchange agreement of a shared secret.
Cryptographic key component (key component)	A parameter used in conjunction with other key components in an Endorsed security function to form a plaintext cryptographic key by a secret sharing algorithm (e.g. the cryptographic plaintext key is the xor-sum of two key components)
Cryptographic module	The set of hardware, software, and/or firmware that implements Endorsed security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.

Term	PP CM (all security levels)
Cryptographic protocol	A cryptographic algorithm including interaction with an external entity (e.g. key exchange)
Data input interface/port	Interface respective port intended for all data (except control data entered via the control input interface) that is input to and processed by the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and status information from another entities).
Data output interface/port:	Interface respective port intended for all data (except status data output via the status output interface) that is output from the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and control information for another entity).
Data path	The physical or logical route over which data passes; a physical data path may be shared by multiple logical data paths.
Decryption algorithm	Algorithm of decoding a cipher text into the plaintext using a decryption key. The decryption algorithm reproduces the plaintext which where used to calculate the cipher text with the corresponding encryption algorithm and the corresponding encryption key .
Destruction of data	a method of erasing electronically stored data, cryptographic keys, and CSPs by altering or deleting the contents of the data storage to prevent recovery of the data.
Differential power analysis (DPA)	an analysis of the variations of the electrical power consumption of a cryptographic module, using advanced statistical methods and/or other techniques, for the purpose of extracting information correlated to cryptographic keys used in a cryptographic algorithm.
Digital signature	the result of a asymmetric cryptographic transformation of data which, when properly implemented, provides the services of 1. origin authentication, 2. data integrity, and 3. signer non-repudiation.
Electromagnetic compatibility (EMC)	The ability of electronic devices to function satisfactorily in an electromagnetic environment without introducing intolerable electromagnetic disturbances to other devices in that environment.
Electromagnetic emanation analysis (EMEA)	Analysis of electromagnetic emissions from a device, equipment, or system to gain information about its internal secrets or processes
Electromagnetic interference (EMI)	Electromagnetic emissions from a device, equipment, or system that interfere with the normal operation of another device, equipment, or system.
Electronic key entry	The entry of cryptographic keys into a cryptographic module using electronic methods such as a smart card or a key-loading device. (The user of the key may have no knowledge of the value of the key being entered.)
Encrypted key	a cryptographic key that has been encrypted using an Endorsed

Term	PP CM (all security levels)
	security function with a key encrypting key, a PIN, or a password in order to disguise the value of the underlying plaintext key.
Encryption algorithm	Algorithm of processing a plaintext into a ciphertext using a encryption key in a way that decoding of the cipher text into the plain text without knowledge of the corresponding decryption key is computationally infeasible.
End User	An individual or process (subject) acting on behalf of the individual that accesses a cryptographic module in order to obtain cryptographic services.
Endorsed	For this protection profile, endorsed by the certification body for the evaluation of products of an intended type and resistance against attacks with attack potential addressed by the vulnerability analysis component in the security target[82].
Endorsed mode of operation	For this protection profile, a operational mode of the cryptographic module that employs only Endorsed security functions (e.g. installation, start-up, normal operation, maintenance; not to be confused with a specific mode of an Endorsed security function, e.g., DES CBC mode)
Endorsed security function	For this protection profile, a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either a) specified in an Endorsed standard, b) adopted in an Endorsed standard and specified either in an appendix of the Endorsed standard or in a document referenced by the Endorsed standard, or c) specified in the list of Endorsed security functions.
End-user	An individual or a process (subject) acting on behalf of the individual that accesses a cryptographic module in order to obtain cryptographic services.
Environmental failure protection (EFP)	The use of features to protect against a compromise of the security of a cryptographic module due to environmental conditions or fluctuations outside of the module's normal operating range.
Environmental failure testing (EFT)	the use of testing to provide a reasonable assurance that the security of a cryptographic module will not be compromised by environmental conditions or fluctuations outside of the module's normal operating range.
Error detection code (EDC)	a code computed from data and comprised of redundant bits of information designed to detect, but not correct, unintentional changes in the data.
Error mode	Mode of operation when the cryptographic module has encountered an error condition as defined in FPT_FLS.1 (term is used for

[82] Endorsed algorithms and functions could be similar to the list of cryptographic algorithms and parameters published for qualified electronic signatures by the notified body Bundesnetzagentur in Germany [5] or the Approved algorithms published by NIST in the USA.

Term	PP CM (all security levels)
	description of the Mode transition SFP).
Error state	State related to the Error mode in the Finite state model (cf. ADV_SPM.1).
Firmware	The programs and data components of a cryptographic module that are stored in hardware (e.g., ROM, PROM, EPROM, EEPROM or FLASH) and cannot be dynamically written or modified during execution.
Hardware	The physical equipment used to process programs and data.
Hash-based message authentication code (HMAC)	A message authentication code that utilizes a keyed hash.
Information processing	The organisation, manipulation and distribution of information.
Initialization vector (IV)	A vector used in defining the starting point of an encryption process within a cryptographic algorithm.
Input data	Information that is entered into a cryptographic module for the purposes of transformation or computation using an Endorsed security function.
Integrity	The property that sensitive data has not been modified or deleted in an unauthorized and undetected manner.
Internal secrets	Confidential data inside the cryptographic boundary not intended for export (e.g. secret or private plaintext keys, authentication reference data).
IT system	for this protection profile, a IT system using the TOE to protect user data during transmission over or storage on media to which unauthorised user have access to
Key encrypting key	a cryptographic key that is used for the encryption or decryption of other keys.
Key establishment	the process by which cryptographic keys are securely distributed among cryptographic modules using manual transport methods (e.g., key loaders), automated methods (e.g., key transport and/or key agreement protocols), or a combination of automated and manual methods (consists of key transport plus key agreement).
Key interface/port	Data interface respective port used for the input and output of plaintext cryptographic key components and CSPs.
Key loader	a self-contained unit that is capable of storing at least one plaintext or encrypted cryptographic key or key component that can be transferred, upon request, into a cryptographic module.
Key management	the activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.
Key material	any media storing key components or keys for offline key exchange.

Term	PP CM (all security levels)
Key transport	secure transport of cryptographic keys from one cryptographic module to another module.
Key usage type	Type of cryptographic algorithm a key can be used for (e.g. DES encryption, TDES MAC calculation, signature-creation with RSA PKCS#1 v1.5)
Logical external interface	a logical entry or exit point of a cryptographic module that provides access to the module for logical information flows representing physical signals (see also the term “port” for the physical aspects of a logical external interface). In the CC terminology it covers all logical external interfaces of the TOE (direct or indirect interfaces to the TSF or interfaces to the non-TSF portion of the TOE, cf. CEM paragraph 529 for details).
Non-operational CSP	CSP used only for self test (e.g. for known answer tests) and maintenance operation (e.g. to test the operation of the cryptographic module after software update or repairing hardware components). Non-operational must not be used for protection of user the confidentiality or integrity of data by cryptographic operation.
Maintenance mode	Mode of operation for maintaining and servicing a cryptographic module, including physical and logical maintenance testing.
Maintenance state	State related to the Maintenance mode in the Finite state model (cf. ADV_SPM.1).
Manual key entry	the entry of cryptographic keys into a cryptographic module, using devices such as a keyboard.
Manual key transport	a non-electronic means of transporting cryptographic keys.
Message authentication with appendix	A digital signature scheme which requires the message as input to the verification algorithm. The signature is attached to the message.
Message authentication with message recovery	A digital signature scheme with message recovery is a digital signature scheme for which a priori knowledge of the message is not required for the verification algorithm.
Microcode	the elementary processor instructions that correspond to an executable program instruction.
Operating conditions	Any environmental condition being accidental or induced outside of the normal range intended for the TOE may affect the correct operation or compromise of confidential information. These conditions include but are not limit to voltage of power supply, temperature, emanation which TOE environmental conditions.
Operational CSP	CSP used for protection of the confidentiality or integrity of data by cryptographic operation.
Output data	Data containing information that is produced from a cryptographic module.
Password	a string of characters (letters, numbers, and other symbols) used to authenticate an identity or to verify access authorization.

Term	PP CM (all security levels)
Personal identification number (PIN)	an alphanumeric code or password used to authenticate an identity.
Permanent stored keys	Keys remains stored in the TOE after power off or reset.
Physical protection	the safeguarding of a cryptographic module, cryptographic keys, or CSPs using physical means.
Plaintext key	an unencrypted cryptographic key.
Port	a physical input or output interface of a cryptographic module that provides access to the module for physical signals, represented by logical information flows. Physically separated ports do not share the same physical pin or wire. In the CC terminology a port is a physical external interface of the TOE (direct or indirect interface to the TSF or interface to the non-TSF portion of the TOE, cf. CEM paragraph 529 for details).
Power interface/port	Interface respective port providing all external electrical power supply.
Private key	a cryptographic key, used with a public key cryptographic algorithm, that is uniquely associated with an entity and is not made public.
Protection Profile	an implementation-independent set of security requirements for a category of Targets of Evaluation (TOEs) that meet specific consumer needs.
Public key	a cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that may be made public.
Public key (asymmetric) cryptographic algorithm	a cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible.
Public key certificate	a set of data that uniquely identifies an entity, contains the entity's public key, and is digitally signed by a trusted party, thereby binding the public key to the entity.
Random Number Generator	Random Number Generators (RNGs) used for cryptographic applications produce a sequence of zero and one bits that may be combined into sub-sequences or blocks of random numbers. There are three basic classes physical true RNG, non-physical true RNG, and deterministic RNG. A physical true RNG produces output that dependents on some physical random source inside the TOE boundary only. A non-deterministic true RNG gets its entropy from sources from outside the TOE boundary (e.g. by system data like RAM data or system time of a PC, output of API functions etc. or human interaction like key strokes, mouse movement etc.). A deterministic RNG consists of an algorithm that produces a sequence of bits from an initial random value (seed).
Reference	Data known for the claimed identity and used by the TOE to verify

Term	PP CM (all security levels)
authentication data	the verification authentication data provided by an entity in an authentication attempt to prove their identity.
Red data	Cryptographically unprotected user data representing user information which need protection in confidentiality and / or integrity.
Removable cover	a cover designed to permit physical access to the contents of a cryptographic module.
Reset	Action to clear any pending errors or events and to bring a system to normal condition or initial state (e.g. after power-up).
Secret key	a cryptographic key, used with a secret key cryptographic algorithm, that is uniquely associated with one or more entities and should not be made public.
Secret key (symmetric) cryptographic algorithm	a cryptographic algorithm which keys for both encryption and decryption respective MAC calculation and MAC verification are the same of can easily be derived from each other and therefore must be kept secret.
Seed key	a secret value used to initialize a cryptographic function or operation.
Self-test mode	Mode of operation in which the cryptographic module performs initial start-up self-test, self-test at power-up, self-test at the request of the authorised user and may perform other self-tests identified in FPT_TST.2.6.
Self-test state	State related to the Self-test mode in the Finite state model (cf. ADV_SPM.1).
Shutdown	Shutdown of the TOE initiated by the user (may not include reset after detection of error or power-off due to loss of power supply)
Signature-creation key	Private key for the creation of digital signatures
Signature-verification key	Public key for the verification of digital signatures
Simple power analysis (SPA)	a direct analysis of patterns of instruction execution (or execution of individual instructions), obtained through monitoring the variations in electrical power consumption of a cryptographic module, for the purpose of revealing the features and implementations of cryptographic algorithms and subsequently the values of cryptographic keys.
Software	the programs and data components, usually stored on erasable media (e.g., disk), that can be dynamically written and modified during execution.
Split knowledge	a process by which a cryptographic key is split into multiple key components, individually sharing no knowledge of the original key, that can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.

Term	PP CM (all security levels)
Status information	information that is output from a cryptographic module for the purposes of indicating certain operational characteristics or modes of the module.
Status output interface/port	Interface respective port intended for all input commands, signals, and control data (including calls and manual controls such as switches, buttons, and keyboards) used to control the operation of the cryptographic module).
System software	the special software within the cryptographic boundary (e.g., operating system, compilers or utility programs) designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system, and associated programs, and data.
Tamper detection	the automatic determination by a cryptographic module that an attempt has been made to compromise the physical security of the module.
Tamper evidence	the external indication that an attempt has been made to compromise the physical security of a cryptographic module. (The evidence of the tamper attempt should be observable by an user subsequent to the attempt.)
Tamper response	the automatic action taken by a cryptographic module when a tamper detection has occurred (the minimum response action is the desctruction of plaintext keys and CSPs).
Target of Evaluation (TOE)	an information technology product or system and associated administrator and user guidance documentation that is the subject of an evaluation.
TEMPEST	a name referring to the investigation, study, and control of unintentional compromising emanations from telecommunications and automated information systems equipment. Note, TEMPEST is not limited to electromagnetic emanation.
Timing analysis	Analysis of timing behaviour of a device, equipment, or system to gain information about its internal secrets or processes
TOE Security Functions (TSF)	a set of the TOE consisting of all hardware, software, and firmware that must be relied upon for the correct enforcement of the TOE Security Policy.
TOE security functions interface (TSFI)	a set of interfaces, whether interactive (man-machine interface) or machine (machine-machine interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.
TOE Security Policy (TSP)	a set of rules that regulate how assets are managed, protected, and distributed within a Target of Evaluation.
Trusted channel	A means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSP.
Trusted path	a means by which a user and a TSF can communicate with

Term	PP CM (all security levels)
	necessary confidence to support the TSP.
Unauthenticated User	An identified user not being authenticated and having rights as identified in the component FIA_UAU.1.
Unauthorized user	A user who may obtain access only to system provided public objects if any exist.
Unidentified User	An user not being identified and having rights as identified in the component FIA_UID.1
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE (includes both authorized and unauthorized entities).
Verification authentication data	Data provided by an entity in an authentication attempt to prove their identity to the TOE.

8.2 Acronyms

Acronym	Term
A.xxx	Assumption
CC	Common Criteria
n.a.	Not applicable
O.xxx	Security objective for the TOE
OE.xxx	Security objective for the TOE environment
OSP	Organisational security policy
SAR	Security assurance requirements
SFR	Security functional requirement
T.xxx	Threat
TOE	Target of Evaluation
TSF	TOE security functions

9 Literature

Common Criteria

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 2.3, August 2005, CCMB-2005-08-001

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 2.3, August 2005, CCMB-2005-08-002

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 2.3, August 2005, CCMB-2005-08-003

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 2.3, August 2005, CCMB-2005-08-004

Cryptography

[5] Geeignete Algorithmen zur Erfüllung der Anforderungen nach §17 Abs. 1 bis 3 SigG vom 22. Mai 2001 in Verbindung mit Anlage 1 Abschnitt I Nr. 2 SigV vom 22. November 2001, Bonn, 10.8.2004 (Zieldatum der Veröffentlichung ist Januar 2005)

[6] ISO/IEC 14888-3: Information technology – Security techniques – Digital signatures with appendix – Part 3: Certificate-based mechanisms, 1999

[7] NIST: FIPS PUB 186-2 Digital signature standard (DSS), 2000 January 27 with Change Notice 1, October 2001

10 Annex

10.1 Backup (informal)

This chapter describes additional security problem definition, security objectives and security functional requirements for back-up. The ST writer may use this information in case the TOE supports back-up.

10.1.1 Security Problem Definition

A.Data_Store *Storage and Handling of TOE data*

The TOE environment ensures the confidentiality, integrity and availability of their security relevant data for TOE initialisation, start-up and operation if stored or handled outside the TOE. The TOE environment ensures the availability of the backup data.

Examples of the TOE data are verification authentication data, cryptographic key material and documentation of TOE configuration data.

T.Malfunction *Malfunction of TOE*
Internal malfunction of TOE functions may result in the modification of keys and CSP, misuse of TOE services, disclosure or distortion of TOE or denial of service for authorised users. This includes the destruction of the TOE as well as hardware failures which prevent the TOE from performing its services. This includes also the destruction of the TOE by deliberate action or environmental failure. Technical failure may result in an insecure operational mode violating the integrity and availability of the TOE services.

T.Insecure_Init *Insecure Initialisation of the TOE*
Unauthorised personnel or authorised personnel without using adequate organisational controls may initialise the TOE with insecure system data, management data or user data. An attacker may manipulate the backup data to initialise the TOE insecurely by the restore procedure.

T.Compromise_Backup *Compromise of backup data*
An attacker may have access to the backup data to compromise confidential cryptographic keys, CSPs and TSF data and use this knowledge to compromise the confidentiality and integrity of user data protected under these secondary assets.

Securtiy objectives for the TOE

O.Protect_Exported_Data *Protection of Data Exported by the TOE*

The TOE shall apply integrity and confidentiality protection mechanisms to all assets requiring integrity or confidentiality protection when they are exported from the TOE or imported into the TOE for the purpose of backup and restore. Operations for backup and restore shall be performed under dual personal control..

Security objectives for the TOE environment

OE.Recovery Secure Recovery in Case of Major Failure

Recovery plans and procedures shall exist that allow a secure and timely recovery in the case of a major problem with the TOE (i.e. if TOE is blocked in its secure state after a failure, service discontinuity or detected physical tampering). These procedures shall ensure that the confidentiality and integrity of security relevant data for TOE initialisation, start-up and operation are maintained and that the recovery does not result in a situation that allows personnel to extend the TOE services they are allowed to use.

10.1.2 Extension of Class FDP with Family FDP_BKP

The TOE supports backup of cryptographic keys, CSP, other user data and TSF data to restore the operational mode of the same crypto module or for a new crypto module in the event of a system failure or other serious error. The export, import and protection of the backup data are combined in a specific way. The TOE ensures the confidentiality of the backup data and detects loss of the integrity of the backup data. The availability of the backup data will be ensured by the TOE environment.

This component is necessary to specify a unique requirement of certificate issuing and management components that is not addressed by the Common Criteria. The specific requirements address the protection of cryptographic keys, key components, CSP and TSF data for backup and recovery.

Backup and recovery (FDP_BKP)

Family behavior

This family defines export and import of the backup data. The TOE ensures the confidentiality of the backup data and detects loss of the integrity of the backup data. The availability of the backup data will be ensured by the TOE environment.

Component leveling:

FDP_BKP TOE Backup and recovery
1

FDP_BKP.1 Backup and recovery provides export, import and protection of the backup data.

Management: FDP_BKP.1

There are no management activities foreseen.

Audit: FDP_BKP.1

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

a) Use of the backup function,

b) Use of the recovery function,

c) Unsuccessful recovery because of detection of modification of the backup data.

FDP_BKP.1 Backup and recovery

Hierarchical to: No other components.

Dependencies: [FCS_CKM.1 Cryptographic key generation or
FCS_CKM.2 Cryptographic key distribution or
FDP_ITC.1 Import of user data without security attributes]
FCS_COP.1 Cryptographic operation

FDP_BKP.1.1 The TSF shall be capable of invoking the backup function on demand.

FDP_BKP.1.2 The data stored in the backup shall be sufficient to recreate the state of the TOE at the time the backup was created using only: (1) a copy of the same version of the TOE as was used to create the backup data; (2) a stored copy of the backup data; (3) the cryptographic key(s) needed to decrypt the backup data; (4) the cryptographic key(s) needed to verify the cryptographic checksum of the backup data.

FDP_BKP.1.3 The TSF shall include a recovery function that is able to restore the state of the TOE from a backup.

FDP_BKP.1.4 The cryptographic keys, other critical security parameters and other confidential backup data shall be exported in encrypted form only.

FDP_BKP.1.5 The backup data shall be checked for modification through the use of cryptographic checksums. Modified backup data shall not be used for recovery.

10.1.3 Security Functional Requirements for TOE supporting Back-up

FCS_COP.1/Backup_Enc Cryptographic operation – Encryption of Backup data

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
MT_MSA.2 Secure security attributes

FCS_COP.1.1/Backup_Enc The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of* ***Endorsed*** *standards*].

FCS_COP.1/Backup_Int Cryptographic operation – Backup Integrity protection

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
MT_MSA.2 Secure security attributes

FCS_COP.1.1/Backup_Int The TSF shall perform *calculation and verification of cryptographic checksums* in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of* ***Endorsed*** *standards*].

Application note 1: The standards for encryption, decryption, calculation and verification of cryptographic checksums shall be assigned from the list of endorsed algorithms only.

FDP_ACC.2/Backup Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

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

(1) Subjects: Crypto officer, Administrator;

(2) Objects: cryptographic keys, CSP, and backup data, back up key components,

(3) Operations: backup (FDP_BKP.1), restore (FDP_BKP.1), import of backup key components (FCS_CKM.2/Import).

FDP_ACC.2.2/Backup The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

FDP_ACF.1/Backup Security attribute based access control - Backup

Hierarchical to: No other components.

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

FDP_ACF.1.1/Backup The TSF shall enforce the Backup SFP to objects based on *Identity and Role*.

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

(1) *Crypto officer under dual person control of another user in the Crypto officer role or Administrator role is allowed (a) to backup cryptographic keys, CSP and backup data (FDP_BKP.1), (b) to restore cryptographic keys, CSP and backup data (FDP_BKP.1),*

(2) *Crypto officers are allowed to enter backup key components (FCS_CKM.2/Import).*

FDP_ACF.1.3/Backup The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes that explicitly authorise access of subjects to objects*].

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

(1) *no role is allowed without dual control of an Crypto officer (a) to backup cryptographic keys, CSP and backup data (FDP_BKP.1), (b) to restore cryptographic keys, CSP and backup data (FDP_BKP.1),*

(2) *any other role than Crypto officer is not allowed to import a backup key component (FCS_CKM.2).*

Application note 2: "Dual person control" requires at least two subjects bind to two different users authenticated for the required role authorized to perform the required actions. One of them shall be an authorized Crypto officer. FCS_CKM.2/Import enforces requirement for manually-entered key components similar to dual control: the Crypto officers are allowed to import only one of at least two key components but import of these key components may be performed in different points of time.

10.1.4 Backup and recovery (FDP_BKP.1)

FDP_BKP.1 Backup and recovery

Hierarchical to: No other components.

Dependencies: [FCS_CKM.1 Cryptographic key generation or
FCS_CKM.2 Cryptographic key distribution or
FDP_ITC.1 Import of user data without security attributes]
FCS_COP.1 Cryptographic operation

FDP_BKP.1.1 The TSF shall be capable of invoking the backup function on demand.

FDP_BKP.1.2 The data stored in the backup shall be sufficient to recreate the state of the TOE at the time the backup was created using only:

(1) a copy of the same version of the TOE as was used to create the backup data;

(2) a stored copy of the backup data;

(3) the cryptographic key(s) needed to decrypt the backup data;

(4) the cryptographic key(s) needed to verify the cryptographic checksum of the backup data.

FDP_BKP.1.3 The TSF shall include a recovery function that is able to restore the state of the TOE from a backup.

FDP_BKP.1.4 The cryptographic keys, other critical security parameters and other confidential backup data shall be exported in encrypted form only.

FDP_BKP.1.5 The backup data shall be checked for modification through the use of cryptographic checksums. Modified backup data shall not be used for recovery.

10.1.5 Rationale

The following thoughts help the ST writer to provide rationale for the security objectives and the security functional requirements.

The assumption **A.Data_Store "**Storage and Handling of TOE data" is covered by the security objective for the environment OE.Recovery "Secure Recovery in Case of Major Failure", second sentence.

The threat **T.Malfunction** "Malfunction of TOE" is covered by adequate reaction in case of malfunction based on security mechanisms of the TOE required by O.Protect_Exported_Data "Protection of Data Exported by the TOE" and implemented by organisational security measures required by security objective for the environment OE.Recovery "Secure Recovery in Case of Major Failure".

The threat **T.Insecure_Init** "Insecure Initialisation of the TOE" is covered by the security objectives for the TOE O.Protect_Exported_Data "Protection of Data Exported by the TOE" (cf. dual personal control) together with O.I&A "Identification and authentication of users", O.Roles "Roles known to TOE", and for the environment OE.Recovery "Secure Recovery in Case of Major Failure" and OE.Personal "Personal security"

The threat **T.Compromise_Backup** "Compromise of backup data " is prevented by the security objective O.Protect_Exported_Data "Protection of Data Exported by the TOE" requiring protection of the confidentiality and integrity of the backup data of .the Operations for backup and restore.

All SFR FCS_COP.1/Backup_Enc, FCS_COP.1/Backup_Int, FDP_ACC.2/Backup, FDP_ACF.1/Backup and FDP_BKP.1 Backup and recovery are mapped to the security object O.Protect_Exported_Data "Protection of Data Exported by the TOE". Further more FAU_GEN.1 (with the addition described above) contributes to O.Protect_Exported_Data.