CryptoServer

Security Target Lite for CryptoServer Se-Series Gen2 CP5

utimaco®

Imprint

Table of Contents

1 Introduction

1.1 Change History

Version	Date	Description
1.9.5	13th November 2018	First release of Security Target Lite for CryptoServer CP5 5.1.0.0, based on full Security Target with same version
2.0.0	23th November 2018	Release of Security Target Lite for CryptoServer CP5 5.1.0.0, based on full Security Target with same version
2.0.2	1st April 2020	Release of Security Target Lite for CryptoServer CP5 5.1.0.0, Maintenance re-certification, based on full Security Target with same version: Version number of SAM Developer Guide updated
2.0.4	18th November 2020	• Updated user guidance versions (SAM Developer Guide and others) and details of non-TOE smartcard deliverables. • Conformance claim is updated to version 1.0 of the protection profile (EN 419221-5:2018 Protection Profiles for TSP Cryptographic Modules - Part 5: Cryptographic Module for Trust Services) and all the references to protection profile are updated accordingly.
2.1.1	06th September 2023	• CC references are updated to CC Version 3.1 Revision 5, April 2017 for the CP5 5.1.0.0 recertification • Guidance doc references are updated • SOG-IS reference is updated to the current version 1.3 • Certification Authority reference is updated to TrustCB B.V.
2.1.2	21st November 2023	• Remaining LAN v4 picture is deleted
2.2.0	4th February 2025	Update for CP5.1.2.0 • Update Firmware Module Version numbers • Update DH key size to 3072 bits • Add new smartcard Add Note 27&29 from “SOGIS-Agreed-Cryptographic-Mechanisms-1.2” concerning RSA key lengths Adapt Table 1

1.2 Document Introduction

This Security Target (ST) was developed based on the Protection Profile (PP) “EN 419 221-5 Protection Profiles for TSP Cryptographic Modules - Part 5: Cryptographic Module for Trust Services”, version v1.0 [PP_CMTS].

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

1.2.1 Acknowledgement

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

1.2.2 Notations

The notation, formatting, and conventions used in this ST are consistent with those used in the Common Criteria, Version 3.1, Revision 5, April 2017 [CC1], [CC2], [CC3].
The Common Criteria allow several operations to be performed on security requirements: refinement, selection, assignment and iteration are defined in Section C.2 of [CC1].

- **Refinement:** The refinement operation is used to add details to a requirement, and thus further restricts a requirement.
- **Selection:** The selection operation occurs where a given component contains an element where a choice from several items has to be made by the PP/ST author. Whenever an element within a PP contains a selection, the PP author could leave the selection uncompleted, restrict the selection by removing some of the choices (but leaving two or more) or complete the selection by choosing one or more items. Whenever an element within an ST contains a selection, the ST author has to complete that selection. If a selection was already completed in the PP, the PP text is shown in *non-underlined italic letters*. If a selection is completed by the ST author the text is shown in *underlined italicized letters*.
- **Assignment:** The assignment operation is used to assign a specific value to an unspecified parameter (e.g. the length of a password). Whenever an element in a PP contains an assignment, the PP author could leave the assignment uncompleted, complete the assignment, narrow the assignment, to further limit the range of values that is allowed or transform the assignment to a selection, thereby narrowing the assignment. Whenever an element in an ST contains an assignment, the ST author has to complete that assignment, the PP text is also given within a footnote where the original text is given. If an assignment was already completed in the PP, the PP text is shown in *non-underlined italic letters*. If an assignment is completed by the ST author the text is shown in *underlined italicized letters*.
- **Iteration:** The iteration operation is used when a component is repeated with varying operations. Iterations within [PP_CMTS] are denoted by showing a slash “/” and an iteration indicator after the CC component identifier. Iterations within the ST are denoted by showing a double-slash “//” and an iteration indicator after the PP component and the CC component, respectively, identifier.

1.2.3 Abbreviations

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

T.<xxx>	Threats
P.<xxx>	Organisational security policies
A.<xxx>	Assumptions about the TOE security environment
OT.<xxx>	Security objectives for the TOE
OE.<xxx>	Security objectives for the operating environment

1.2.4 References

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

1.2.5 Terminology

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

2 Security Target Introduction

2.1 ST and TOE Reference

Title:	CryptoServer - Security Target Lite for CryptoServer Se-Series Gen2 CP5
ST Version:	2.2.0
ST Date:	4th February 2025
Author:	Utimaco IS GmbH
Developer:	Utimaco IS GmbH
Product:	CryptoServer Se-Series Gen2 CP5
TOE-name long:	CryptoServer Se-Series Gen2 CP5
TOE-name short:	CryptoServer CP5
TOE-versions:	CryptoServer CP5 Se12 5.1.2.0, CryptoServer CP5 Se52 5.1.2.0, CryptoServer CP5 Se500 5.1.2.0, CryptoServer CP5 Se1500 5.1.2.0
Product Type:	Cryptographic module
Certification Authority:	TrustCB B.V.
CC Version:	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017 [CC1], [CC2], [CC3]
Keywords:	Cryptographic module, Hardware security module (HSM), crypto server, electronic signature, digital signature, digital seal, sealing, eIDAS Regulation, Qualified Electronic Signature Creation Device, QSCD, SAM, Signature Activation Module, Trust Service Provider, TSP, key generation, key management, tamper protection, secure messaging, trusted channel, random number generation, ECDSA, RSA, AES, SHA, HMAC

2.2 Related Documents

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

2.3 Organisation

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

- **Chapter 2:** The TOE description provides general information about the TOE, its generic structure and boundaries.

- **Chapter 3:** The ST conformance claims section states conformance to Protection Profiles.
- **Chapter 4**: The security problem definition describes security aspects of the environment in which the TOE is intended to be used and the manner in which it is intended to be employed. The security problem definition includes threats relevant to secure TOE operation (section 4.3), organisational security policies (section 4.4), which must be complied by the TOE, and assumptions regarding the TOE's intended usage and environment of use (section 4.5).
- **Chapter 5:** The statement of security objectives defines the security objectives for the TOE (section 5.1) and for its environment (section 5.2). The rationale (section 8.1) presents evidence that the security objectives satisfy the threats and policies.
- **Chapter 6:** This chapter defines the extended components.
- **Chapter 7:** The security requirements are subdivided into TOE Security Functional Requirements (section 7.3) and Security Assurance Requirements (section7.4).
- **Chapter 8:** The rationale (section 8.2) explains how the set of requirements is complete relative to the security objectives.
- **Chapter 9:** The TOE summary specification provides a description of the TOE security functionality in narrative form.

The **annex** offers a glossary and acronyms as well as relevant references.

2.4 TOE Overview

The scope of this Security Target is to describe the functionality of the TOE "CryptoServer Se-Series Gen2 CP5" (short: CryptoServer CP5, or CryptoServer in this document) in terms of Common Criteria and to define security functional and assurance requirements for this system.

The CryptoServer CP5 is a hardware security module whose primary purpose is to provide secure cryptographic services such as signing and verification of data (ECDSA, RSA), encryption or decryption (for various cryptographic algorithms like AES and RSA), hashing, on-board random number generation and secure key generation, key storage and further key management functions in a tamper-protected environment. As such, it may serve as general purpose HSM. Apart from that, the CryptoServer CP5 supports local signing/sealing and server signing in accordance with EN 419 241-1 Security Requirements and EN 419 241-2 Protection Profile for QSCD for Server Signing [PP_QSCD]. Furthermore, it provides the functionality for creating protected backups of keys and for secure update of defined parts of the TOE software.

CryptoServer CP5 is a Qualified Signature/Seal Creation Device (QSCD) where the electronic signature/seal creation data is held in an entirely but not necessarily exclusively user-managed environment. Moreover, the CryptoServer CP5 is suitable for use by trust service providers (TSP) supporting electronic signature and electronic sealing operations, certificate issuance and revocation, time stamp operations, and authentication services, as identified in eIDAS Regulation (EU) No. 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market [Regulation]. In order to meet the security assurance requirements of Qualified Electronic Signature, and Electronic Seal, Creation Devices for use by trust service providers as specified in eIDAS Regulation [Regulation], the CryptoServer CP5 supports the concept of “Assigned keys” as defined in the Protection Profile (PP) [PP_CMTS]. Assigned keys ensure that the signatory has *sole control* over the use of his private, assigned key that is used to create digital signatures according to eIDAS. The PP [PP_CMTS] defines the security

requirements for cryptographic modules used by trust service providers as identified in eIDAS Regulation [Regulation].

The CryptoServer can optionally also be used as a general purpose HSM. Thus, in addition to Assigned keys, also General (non-Assigned) keys that have lower restrictions are supported. General keys can for instance be exported or imported (in an encrypted way), whereas key export or import is not allowed for Assigned keys.

In addition to its suitability as Qualified Signature Creation Device (QSCD) for local signing/sealing and server signing in accordance with eIDAS Regulation [Regulation], the CryptoServer can optionally also be used for implementing a QSCD for eIDAS-compliant *remote Server Signing* in the sense of CEN Protection Profile EN 419 241-2, “Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing”, [PP_QSCD]. In this case, the customer has to develop a Signature Activation Module (SAM) module in the sense of [PP_QSCD] and certify it against this CEN Protection Profile EN 419 241-2, [PP_QSCD]. The SAM may be either implemented as so-called external SAM which calls the CryptoServer’s external interface (PCIe, see below) for signature creation, and which must be located within a physically protected environment and communicate with the CryptoServer over a secure channel, or it may be implemented as so-called internal SAM in form of one or more firmware modules which must be loaded into the CryptoServer CP5 and run within its secure physical boundary. For signature creation, such internal SAM can use the CryptoServer’s internal interface, see also Figure 5. Both architectures allow the SAM to use the services of the TOE as a cryptographic module that is eIDAS-certified according to CEN Protection Profile prEN 419 221-5 [PP_CMTS]. SAM and CryptoServer CP5 together form a QSCD for remote Server Signing in the sense of CEN Protection Profile EN 419 241-2 [PP_QSCD].

An internal SAM can only be loaded into a CryptoServer CP5 if it is signed by Utimaco with a dedicated CryptoServer CP5 Module Signature Key. Utimaco will only sign firmware modules with this key if they belong to the TOE, or if the module is an internal SAM which is eIDAS-certified according to [PP_QSCD], and if, as part of this eIDAS certification, it has been validated that the SAM follows the CryptoServer CP5 User Guidance so that it doesn’t violate the TOE Security Functionality.

The CryptoServer CP5 is designed as a protected cryptographic module provided in form of a PCIe (PCI express) plug-in card (specific hardware and software product, see Figure 1) for high security applications.

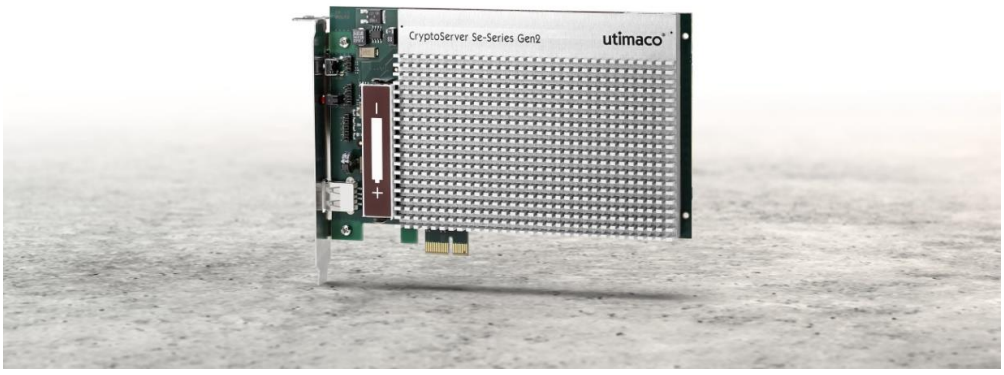


Figure 1: CryptoServer Se-Series Gen2 PCIe security module

All hardware components of the TOE, including the Central Processing Unit, all memory chips, Real Time Clock, and hardware noise generator for random number generation, are located on a printed circuit board (PCB). These hardware components are completely covered with potting material (epoxy resin) and a heat sink. This hard, opaque enclosure protects the sensitive CryptoServer hardware components from physical attacks. The resistance of the TOE hardware and sensory to physical and chemical attacks has been evaluated and successfully certified according to the requirements of FIPS 140-2 standard, level 3. The protected security module PCB in form of a PCIe plug-in card is called PCIe security module.

To enable communication of the cryptographic module with a host, the PCIe security module offers a PCIe interface and two USB interfaces.

Before delivery the PCIe security module can be optionally integrated into a Utimaco CryptoServer LAN appliance (see Figure 2). The CryptoServer LAN exists in two variants providing the same functionality but having a different height. Both are a 19-inch network appliance with display, control buttons and USB interfaces on the front panel. They contain an industry-quality PC motherboard, backplane with PCIe bus interface, flash disk (as mass storage), two redundant power supplies and backup battery. The PCIe security module is plugged into the PCIe bus interface of the backplane. The CryptoServer LAN may be connected to an Ethernet network via a Gigabit network interface on the backside.

Figure 2: CryptoServer LAN v5

2.5 TOE Description

This chapter contains the following sections:

- TOE configuration and TOE environment (section 2.5.1)
- TOE boundary (section 2.5.2)

2.5.1 TOE Configuration and TOE Environment

The Target of Evaluation (TOE) is the cryptographic module "CryptoServer Se-Series Gen2 CP5" (CryptoServer CP5, or CryptoServer), versions CryptoServer CP5 Se12 5.1.2.0, CryptoServer CP5 Se52 5.1.2.0, CryptoServer CP5 Se500 5.1.2.0 and CryptoServer CP5 Se1500 5.1.2.0. The different versions (models) reflect different performance capabilities regarding public key operations (RSA, ECDSA) and transactions per second, with CryptoServer CP5 Se1500 5.1.2.0 being the model with highest performance and CryptoServer CP5 Se12 5.1.2.0 being the model with lowest performance and lowest price.

The CryptoServer CP5 is a cryptographic module where at the time of delivery all hardware components of the cryptographic module, including the Central Processing Unit, all memory chips, Real Time Clock, and hardware noise generator for random number generation, are located on a printed circuit board (PCB). Versions CryptoServer CP5 Se500 5.1.2.0 and CryptoServer CP5 Se1500 5.1.2.0 additionally contain a crypto accelerator chip (in order to provide highest performance on RSA and ECDSA operations), which is not assembled in versions CryptoServer CP5 Se12 5.1.2.0 or CryptoServer CP5 Se52 5.1.2.0.

All hardware components are completely covered with potting material (epoxy resin) and heat sink. The this way protected PCB is given in form of a PCIe plug-in card and is called PCIe security module.

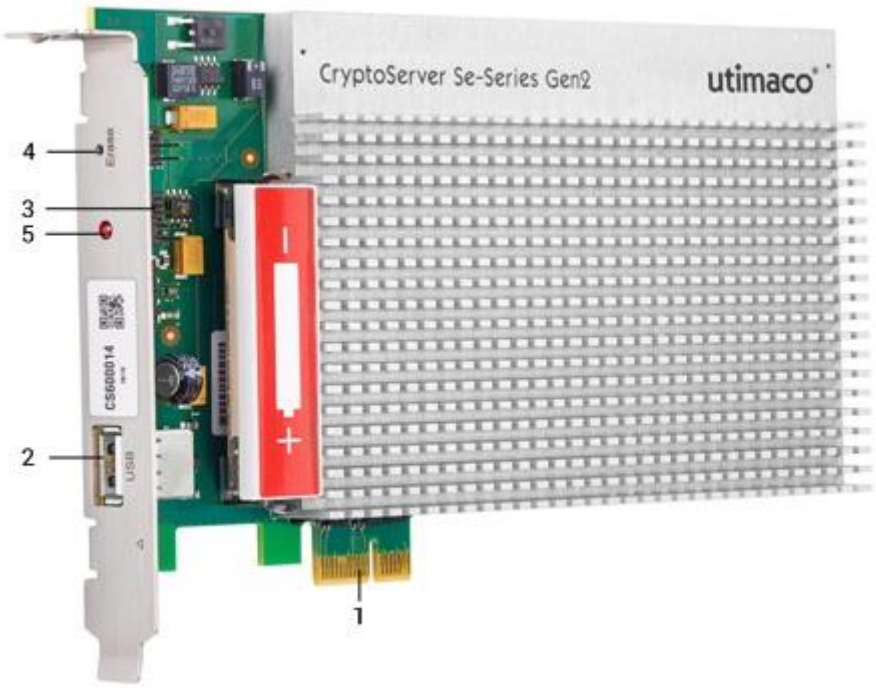


Figure 3: PCIe security module with interfaces

The PCIe security module provides the following interfaces (see Figure 3):

- (1) PCIe interface which is used for operational power input, data input, data output, control input and status output
- (2) USB 2.0 port (can be used for receiving status output)
- (3) USB 2.0 port (can be used for receiving status output)
- (4) Erase pushbutton for performing External Erase

- (5) LED flash light, flashes up red to indicate the activation of the Erase pushbutton

The module's cryptographic boundary consists of a physical boundary and a logical boundary. The physical boundary is defined as the outer perimeter of the heat sink on the top side and the epoxy surface on the bottom side of the module. Figure 4 below shows the physical boundary from the side and the top. The red dashed line indicates the TOE's physical boundary.

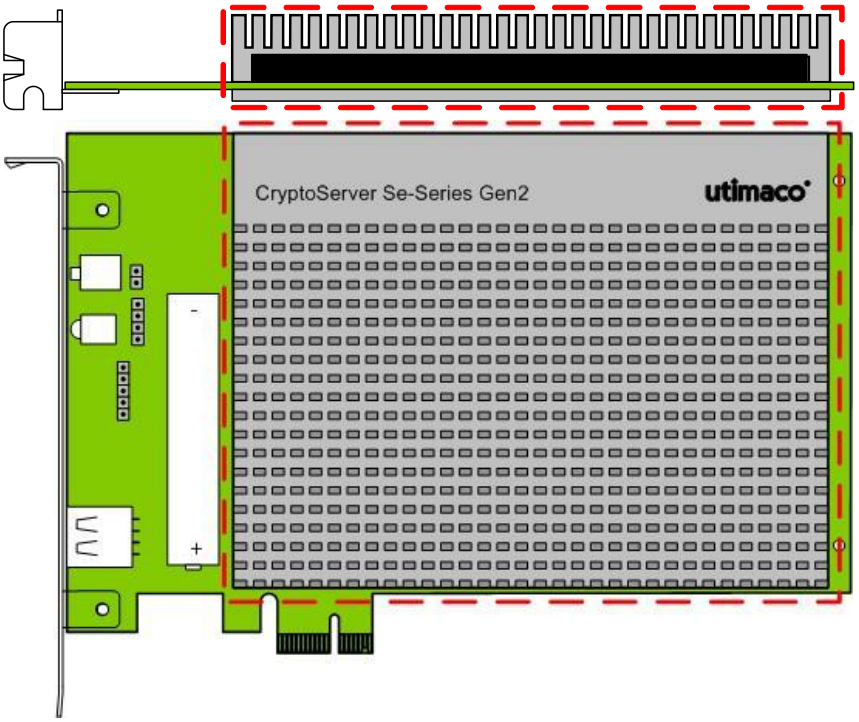


Figure 4: CryptoServer Se-Series Gen2 CP5 physical boundary

The communication between the customer's application software and the cryptographic module is supported by the PCIe interface, by a PCIe device driver and communication software. A PIN pad (smartcard reader with keypad) and smartcards are provided to support the administration of the cryptographic module, and for the management of key shares.

The software components integrated in the cryptographic module have a modular structure and comprise the following parts:

Part 1 – Boot loader

Part 2 – Security Module Operating System (SMOS)

Part 3 – Firmware modules

These software components are developed by Utimaco and will be loaded by Utimaco into the cryptographic module. Additionally, the operating system SMOS and all firmware modules are packed in a so-called CryptoServer CP5 firmware package and can also be loaded into the cryptographic module by the customer.

The logical boundary consists of the external TOE interface which can be accessed via the TOE's PCIe interface, and of an internal TOE interface which may optionally be used by an internal SAM (Signature Activation Module). Such internal SAM consists of one or more firmware modules which are loaded and running within the secure physical boundary of the CryptoServer CP5. An internal SAM provides its own external interface functions which are

available at the PCIe interface in addition to the CryptoServer CP5 external interface functions. It must use the TOE command handling as provided by the CryptoServer in order to communicate via the PCIe interface.

The following requirements hold for an internal SAM:

1. Such internal SAM must be eIDAS-certified according to [PP_QSCD].
2. As part of this certification it is validated that the SAM follows the TOE user guidance and does not violate the TOE Security Functionality.
3. An internal SAM must be signed by Utimaco with the dedicated CryptoServer CP5 Module Signature Key (this signature is only used for internal SAM modules fulfilling (1) and (2)),
4. and it must be loaded into the CryptoServer CP5. This is only possible if the SAM is signed with the CryptoServer CP5 Module Signature Key, and if the command for the download is authenticated by a user in Administrator role.

Figure 5 below shows the logical boundary of the TOE (red dashed line). The physical boundary is indicated by the blue line.

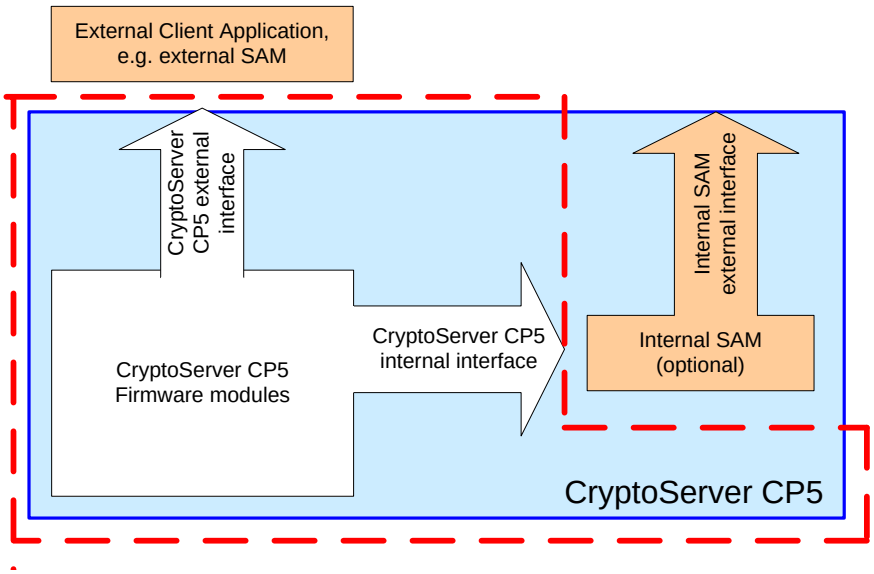


Figure 5: CryptoServer Se-Series Gen2 CP5 logical boundary

The CryptoServer CP5 supports the following cryptographic algorithms:

- AES for encryption, decryption, CMAC calculation and secure messaging
- ECDSA with key size >= 224 bit on dedicated elliptic curves for signature generation and signature verification
- RSA with key size >= 2048[1] bit and <= 8192 bit for signature generation and signature verification and encryption and decryption
- SHA-2, SHA-3 and HMAC for hashing

[1] The acceptability deadline for the legacy use of modulus of size above 1900 bits, but less than 3000 bits, is set to December 31, 2025. See [SOG-IS-Crypto] Note 27. For legacy reasons shorter key sizes are allowed for signature verification and decryption.

Furthermore the CryptoServer CP5 offers key establishment:

- AES key generation
- ECDSA key generation
- RSA key generation
- Diffie-Hellman Key Agreement
- Key Derivation

For random number generation and generation of all cryptographic keys, the CryptoServer CP5 relies on an implemented hardware random noise generator.

For operation purposes, the CryptoServer CP5 supports the following cryptographic services:

- Functions for Initialisation:
  - Generation and export of Master Backup Key
  - Import of Master Backup Key
- Functions for Key Management:
  - Key generation (AES keys, ECDSA key pairs, RSA key pairs)
  - Encrypted import and export of private and secret keys
  - Backup and restore of keys
  - Key deletion
- Cryptographic Functions:
  - Signature generation (ECDSA, RSA)
  - Signature verification (ECDSA, RSA)
  - Encryption (AES, RSA)
  - Decryption (AES, RSA)
  - Generation of random bytes

For the operation purpose, the CryptoServer CP5 supports the following administrative services:

- User administration (creation and deletion of users, change of reference authentication data (RAD))
- System time setting/display
- Export and deletion of audit data
- Unblock user
- Unblock key

To support the security of the above mentioned features of the TOE, the CryptoServer CP5 provides appropriate countermeasures for resistance especially against the following attacks:

- Cloning of the product
- Unauthorised disclosure of confidential data (during generation, storage and processing)
- Unauthorised manipulation of data (during generation, storage and processing)

- Unauthorised usage of private and secret keys
- Forgery of data to be processed
- Derivation of information on the private key from the related public part for generated key pairs
- Physical and chemical attacks

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

The CryptoServer CP5 product life-cycle is decomposed into the following phases:

- **Development phase:** The design and production of the TOE together constitute the development phase of the TOE. In the design phase, the components and the software of the TOE are designed and developed. In the production phase, the TOE is manufactured consisting of an assembly of supplied components and the software. The development phase ends with the delivery of the TOE parts (PCIe security module, software and guidance documents) together with some non-TOE deliverables to the customer.
- **Usage phase:** The initialisation and operational use of the TOE together constitute the usage phase of the TOE. In the initialisation phase the TOE is initialised by generating or importing the Master Backup Key, which is a support key that will be used for later backup and restore of e. g. signature keys. After initialisation (which includes the creation of user accounts), the TOE is enabled for use in key management functions of secret, public and private keys and cryptographic operations like e. g. signing operations. The operational usage phase begins and after authentication the user can use the TOE for key management and for cryptographic tasks. Furthermore, the TOE provides a software update function.

Considering the TOE and its life-cycle described above, the following types of environments can be distinguished:

- Development environment corresponding to the design phase
- Manufacturing environment corresponding to the production phase
- Initialisation environment corresponding to the initialisation phase
- Operational environment corresponding to the operational usage phase

The TOE developers ensure that the assignment of responsibilities during the design phase is done in a manner which maintains IT security. The **development environment** in which the TOE is developed is a well-structured environment with well-defined responsibilities. The specification, implementation and tests in the development departments are well organised. Suitable measures enforce the usage of the guidelines. The confidentiality and integrity of development results is protected. The used measures are always documented.

In the **manufacturing environment** responsibilities are assigned in manner which maintains IT security. The TOE is protected from physical attacks which might compromise security. The manufacturing environment is well documented. Measures are defined to protect security data like cryptographic keys against disclosure and manipulation. Security data generation algorithms are accessible to authorised and trusted persons only. Security data are generated, transported and inserted into the TOE, in such a way to preserve its appropriate confidentiality and integrity. Optionally, the PCIe security module can be integrated into an Utimaco CryptoServer LAN (19-inch network appliance).

Leaving the manufacturing environment, the TOE parts are delivered to the customer. At the customer in the **initialisation environment** the administrative preparations for the operational usage take place initially. The initialisation environment may be identical with the operational environment.

In the **operational environment** the main tasks are user management, key management of secret and private keys and creation of digital signatures or digital seals. The responsibilities are assigned in manner which maintains IT security. After authentication the user can use the TOE e.g. to manage cryptographic keys or to generate signatures. Furthermore, the TOE provides a secure software update mechanism.

The Common Criteria (CC) does not prescribe any specific life-cycle model. However, in order to define the application of the assurance classes, the CC assume the following implicit life-cycle model consisting of three phases: TOE development (including the development as well as the production of the TOE), TOE delivery and TOE operational use.

For the evaluation of the CryptoServer CP5 the development phase (consisting of design and production phase) corresponds to the TOE development in the sense of CC. The usage phase (consisting of initialisation and operational usage phase) corresponds to the TOE operational use in the sense of CC. The TOE delivery takes place between both phases.

The following paragraphs outline how some CC assurance activities apply to parts of the life-cycle (cf. chapter 7.4):

The ALC class which deals with security measures in the development environment of the TOE applies to the development environment (belonging to the design phase) and to the manufacturing environment (belonging to the production phase). In particular, the site where the TOE software is developed as well as the production site are subject to this CC class.

The measures for TOE delivery to the customer are subject to the aspect ALC_DEL of the ALC class.

The guidance documentation delivered by the TOE developer as part of the TOE delivery procedures are subject to the AGD class. The guidance documentation describes all measures necessary for secure usage of the TOE.

The operational usage phase of the TOE is explicitly in focus of this ST. All TOE hardware and executable software are covered by the evaluation. In particular, the ADV class applies to the specification and implementation of the security functionality of the TOE, its security architecture and design. Testing is subject to the ATE class providing assurance that the TOE behaves as described. Vulnerability assessment class AVA addresses the possibility of exploitable vulnerabilities introduced in the development or the operation of the TOE.

2.5.2 TOE Boundary

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

- Hardware, version 5.01.4.0 Se12/Se52 (module without crypto accelerator) or version 5.01.4.0 Se500/Se1500 (module with crypto accelerator)
- Software, version numbers see below
- Guidance documents for the administrator and the user of the CryptoServer CP5, delivered as electronic files.

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

TOE deliverable	Type/Form, Name		Exact reference
Hardware	HW	Hardware of the TOE, PCIe security module (with/without crypto accelerator)	5.01.4.0 Se500/Se1500 (module with crypto accelerator) 5.01.4.0 Se12/Se52 (module without crypto accelerator)
Software	SW	Boot Loader	5.01.4.0
		FPGA	5.01.0.8
		Sensory Controller	2.00.0.31
		and CryptoServer CP5 firmware package consisting of the following firmware modules:	
		ADM (.msc and .sys) (Module Administration)	3.0.25.5
		AES (.msc and .sys) (AES Cryptography)	1.4.1.7
		ASN1 (.msc and.sys) (Decoding and Encoding ASN.1) asn1.msc: asnq.sys:	 1.0.3.81.0.3.4
		CMDS (.msc and.sys) (Command Scheduler) cmds.msc: cmds.sys:	 3.6.0.13 3.6.0.11
		CXI (.msc) (Cryptographic Services eXternal Interface)	2.2.3.7
		CXIAL (.msc) (CXI Abstraction Layer)	1.0.0.1
		DB (.msc and .sys) (Database Management)	1.3.2.4
		ECA (.msc and .sys) (Elliptic Curve Arithmetic)	1.1.12.4
		ECDSA (.msc and .sys) (ECDSA Cryptography)	1.1.16.2
		EXAR (.msc and .sys) (Driver for Crypto Accelerator)	2.2.1.1
		HASH (.msc and .sys) (Hashing Algorithms)	1.0.12.1

TOE deliverable	Type/Form, Name	Exact reference
	HCE (.msc and .sys) (Generic Internal Interface for Crypto Accelerator)	2.2.2.3
	LNA (.msc and .sys) (Long Number Arithmetic)	1.2.4.4
	MBK (.msc) (Master Backup Key Management)	2.3.0.0
	POST (.msc and .sys) (Power-On Self-Tests)	1.0.0.2
	SMOS (.msc and.sys) (Security Module Operating System) smos.msc: smos.sys:	5.6.6.1 5.5.9.2
	UTIL (.msc and .sys) (Utilities for RTC and RNG)	3.0.5.3
	VDES (.msc and .sys) (DES Cryptography)	1.0.9.4
	VRSA (.msc and .sys) (RSA Cryptography)	1.3.6.5
General guidance documents	Doc *Operating Manual in two variants (delivery variant PCIe/LAN v5):*	
	CryptoServer Se-Series Gen2 CP5 PCIe Operating Manual	2017-0006-en, version 1.2.0
	CryptoServer Se-Series Gen2 CP5 LAN V5 Operating Manual	2018-0011-en, version 1.2.0
	User Manual:	
	CryptoServer Se-Series Gen2 CP5 Administration Manual	2017-0008, version 2.3.0
	Interface Specifications:	
	CryptoServer - Firmware Module CXI for CryptoServer CP5 – Interface Specification	2017-0010, version 1.0.4
	CryptoServer - Firmware Module ADM - Interface Specification - ADM Version ≥ 3.0.0.0	2009-0010, version 1.7.6
	CryptoServer - Firmware Module CMDS - Interface Specification	2023-0022, version 1.0.1
	CryptoServer – Firmware Module MBK – Interface Specification	2003-0006, version 1.10.1

TOE deliverable	Type/Form, Name	Exact reference
Internal SAM Developer Documentation *(only delivered to developers of an internal SAM)*	Doc CryptoServer Se-Series Gen2 CP5 - SAM Developer Guide	2018-0013, version 1.1.6
	Interface Specifications:	
	CryptoServer - Firmware Module AES - Interface Specification	2003-0008, version 1.6.1
	CryptoServer – Firmware Module ASN1 – Interface Specification	2002-0006, version 1.2.4
	CryptoServer – Firmware Module CXIAL – Interface Specification	2018-0002, version 1.0.1
	CryptoServer – Firmware Module DB – Interface Specification	2002-0009, version 1.1.9
	CryptoServer – Firmware Module ECA – Interface Specification	2006-0004, version 1.3.6
	CryptoServer – Firmware Module ECDSA – Interface Specification	2006-0005, version 1.4.8
	CryptoServer – Firmware Module HASH – Interface Specification	2002-0010, version 1.6.2
	CryptoServer – Firmware Module SMOS – Interface Specification - SMOS Version ≥ 2.5.0.0	2008-0001, version 2.7.1
	CryptoServer – Firmware Module UTIL – Interface Specification - UTIL Version ≥ 3.0.0.0	2009-0012, version 1.2.5
	CryptoServer – Firmware Module VRSA – Interface Specification	2002-0019, version 1.9.4

Table 1: TOE deliverables

2.6 Required Non-TOE Hardware/Software/Firmware

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

Additional deliverables	Type/Form	Exact reference
PIN pad (smartcard reader with keypad)	HW/SW Utimaco cyberJack one	FW-Version V1.0
10 smartcards (for administrative purposes)	HW/SW Java Card	JCOP J2E081 V2.4.2. R3, or JCOP J2A081 V2.4.1 R3 JCOP J3R180 V4

The TOE is delivered in two different variants:

- CryptoServer Se-Series Gen2 CP5 – PCIe (PCIe plug-in card)
- CryptoServer Se-Series Gen2 CP5 – LAN v5 (network-attached appliance v5)

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

Deliverable:	**CSLAN v5**	**cable**	**product CD**
Delivered variant:			
CryptoServer Se-Series Gen2 CP5 – PCIe	-	-	1
CryptoServer Se-Series Gen2 CP5 – LAN v5	1	2	1

Herein denotes:

- **CSLAN v5**: CryptoServer LAN (19-inch network appliance with two redundant power supplies, version v5) (non-TOE hardware)
- **Cable**: power supply cable (non-TOE hardware)
- **Product CD**: The product CD containing the following firmware, software and data:
  - The CryptoServer driver (for Windows and Linux) (non-TOE software)
  - The USB driver for the PIN pad “Utimaco cyberJack one” for Windows (non-TOE software)
  - Various cryptographic APIs (non-TOE software, to be used on host)
  - The German versions of the Operating Manuals for both TOE delivery variants
  - The documentation of the cryptographic APIs in PDF and HTML format (non-TOE documentation)
  - The installation files of various administration tools and key management tools (non-TOE software, to be used on host)
  - Further guidance documents, e. g. for all administration tools (non-TOE documentation)
  - The `ADMIN.key` keyfile with the authentication key for the default administrator ADMIN of the CryptoServer (non-TOE data)

The TOE is never delivered with any internal SAM loaded.

3 Conformance Claims

3.1 CC Conformance Claim

This ST claims conformance to

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

as follows

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

The

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

has to be taken into account.

3.2 PP Claim

This Security Target claims strict conformance to the Protection Profile *EN 419 221-5:2018 Protection Profiles for TSP Cryptographic Modules – Part 5: Cryptographic Module for Trust Services* [PP_CMTS].

3.3 Package Claim

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

3.4 Conformance Rationale

This Security Target claims strict conformance with the Protection Profile [PP_CMTS].

4 Security Problem Definition

This chapter contains the following sections:

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

4.1 Assets

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

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

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

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

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

4.2 Subjects

The types of subjects identified in this ST are:

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

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

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

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

4.3 Threats

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

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

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

T.KeyMod Unauthorised modification of a key
An attacker makes an unauthorised modification to a secret or public key (R.SecretKey or R.PubKey) while it is stored in, or under the control of, the TOE, including export and backups if supported. This includes replacement of a key as well as making changes to the value of a key, or changing its attributes such as required authorisation, usage constraints or identifier (changing the identifier to the identifier used for another key would allow unauthorised substitution of the original key with a key known to the attacker). The threat therefore includes the case where an attacker is able to break the binding between a key and its critical attributes[2].

T.KeyMisuse Misuse of a key
An attacker uses the TOE to make unauthorised use of a secret key (R.SecretKey) that is managed by the TOE (including the unauthorised use of a secret key for a cryptographic function that is not permitted for that key[3]), without necessarily obtaining access to the value of the key.

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

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

T.KeyOveruse **Overuse of a key**
An attacker uses a key (R.SecretKey) that has been authorised for a specific use (e.g. to make a single signature) in other cryptographic functions that have not been authorised.

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

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

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

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

4.4 Organisational Security Policies

P.Algorithms **Use of approved cryptographic algorithms**
The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognised authorities as appropriate for use by TSPs.

Application Note 1 (PP)
The relevant authorities and endorsements are determined by the context of the client applications that use the TOE. For digital signatures within the European Union this is as indicated in [Regulation] and an exemplary list of approved algorithms and parameters is given in [TS 119 312] (see also [PP_CMTS] section 1.3.1.3).

P.KeyControl **Support for control of keys**
The life cycle of the TOE and any secret keys that it manages (where such keys are associated with specific entities, such as the signature creation data associated with a signatory or the seal creation data associated with a seal creator[4]), shall be implemented in such a way that the secret keys can be reliably protected by the legitimate owner against use by theirs, and in such a way that the use of the secret keys by the TOE can be confined to a set of authorised cryptographic functions.

[4] A seal creator may be a *legal person* (see [Regulation]) rather than a *natural person*, and seal creation data may therefore be authorised for use by a number of natural persons, depending on the nature and requirements of the trust service provided.

Application Note 2 (PP)
This policy is intended to ensure that the TOE can be used for qualified electronic seals and qualified electronic signatures as in [Regulation], but recognises that not all keys are used for such purposes. Therefore, although the TOE must be able to support the necessary strong controls over keys in order to create such seals and signatures, not all keys need the same level and type of control.

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

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

Application Note 3 (PP)
The cryptographic module TOE is assumed to be part of a larger system that manages audit data. The TOE therefore logs audit records, and it is assumed that these are collected, maintained and reviewed in the larger system. Hence there is no separate auditor role within the cryptographic module TOE, but the role of System Auditor is assumed to exist in the larger system – cf. A.AuditSupport in [PP_CMTS] section 3.5.

4.5 Assumptions

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

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

A.DataContext Appropriate use of TOE functions
Any client application using the cryptographic functions of the TOE will ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and confidentiality). For example, when creating a digital signature over a DTBS the client application will ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and will correctly and securely manage the signature received from the TOE; and when certifying a public key, the client application will ensure that necessary checks are made to prove possession of the corresponding private key. The client

application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) performs a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS.
Client applications are also responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events.
Similar requirements apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys.
Appropriate procedures are defined for the initial creation of data and continuing operation of the TOE according to the specific risks applicable to the deployment environment and the ways in which the TOE is used.

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

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

Application Note 4 (PP)
As noted for P.Audit in [PP_CMTS] section 3.4, the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.

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

5 Security Objectives

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

5.1 Security Objectives for the TOE

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

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

OT.Algorithms Use of approved cryptographic algorithms
The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognised authorities as appropriate for use by TSPs. This ensures that the algorithms used do not enable publicly known data to be used to derive secret keys.

Application Note 5 (PP)
See note under P.Algorithms (section 4.4) on relevant references for digital signatures within the European Union.

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

OT.Auth Authorisation for use of TOE functions and data
The TOE carries out an authentication/authorisation check on all subjects before allowing them to use the TOE. The following types of entity are distinguished for the purposes of authorisation (i.e. each type has a distinct method of authorisation):

- administrators of the TOE
- users of TOE cryptographic functions (client applications using secure channels)
- users of secret keys.

In particular, the TOE always requires authorisations before using a secret key.

Application Note 6 (PP)
Local client applications within a suitable security environment (such as client applications that are connected to the TOE by a channel such as a PCIe bus within the same hardware

appliance) do not require authentication to communicate with the TOE, as noted in [PP_CMTS] section 1.3.1. However, use of a secret key always requires prior authorisation.

Application Note 1 (ST)

Local Client Applications may either be internal, running within the physical boundary of the TOE and using the internal TOE interface, or non-internal, using the external TOE PCIe interface.

- *An external SAM (Signature Activation Module) is a non-internal Local Client Application using the local (external) TOE PCIe interface which has to authenticate like any other non-internal Local Client Application.*
- *An internal SAM is an internal Local Client Application which consists of firmware modules running within the secure physical boundary of the CryptoServer CP5 and using the internal TOE interface in accordance with the TOE Guidance. Such internal SAM is integrity protected by the TOE and does not require authentication to communicate with the TOE, as noted in Application Note 6 (PP) above and Application Note 29 (PP) (taken from [PP_CMTS], Application Note 6 in section 1.3.1, and Application Note 29).*

TOE Guidance describes the internal TOE interface and specifies all rules how the internal TOE interface must be used by an internal SAM module so that the TOE security functionality is not compromised. The adherence to this guidance must be validated in the context of the eIDAS evaluation of the internal SAM according to [PP_QSCD]. Therefore, the TOE can assume that the internal SAM is trustworthy and does not compromise the TOE security functionality.

An internal SAM without this evaluation cannot be loaded into the TOE because it will not be signed with the CryptoServer CP5 Module Signature Key and hence the TOE will reject it.

The internal TOE interface contains a general CXIAL interface providing interface functions which allow the internal SAM to benefit from the security functionality as implemented in the TOE including key attribute management and key authorisation.

The internal TOE interface also contains a service, the SAEK signature interface (SAEK = SAM Authorised External Key) for signature calculation, which allows the internal SAM to implement its own key authorisation functionality: The SAEK signature interface allows an internal SAM application to request usage of a signature key for which the TOE will not check the key authorisation. As a consequence, an internal SAM calling the SAEK signature interface takes full responsibility on correct legitimation of this operation, including key authorisation as required by [PP_CMTS]: As requested by TOE Guidance, the internal SAM may only invoke the SAEK signature interface if it has completely validated key authorisation of the signature key before. Therefore, the TOE can implicitly derive prior successful key authorisation of the signature key from each invocation of the SAEK signature interface.

Additionally, the internal TOE interface contains a general cryptographic interface providing interface functions which allow the internal SAM to use it as a pure cryptographic library and to benefit from the cryptographic SFRs FCS_COP.x and FCS_CKM.x as implemented by the TOE. For cryptographic keys generated or used with functions from this general cryptographic interface none of the other SFRs like FDP_<xxx> are claimed, instead, the internal SAM is fully responsible for the secure usage and management of such keys.

OT.KeyUseConstraint Constraints on use of keys

Any key (secret or public) has an unambiguous definition of the purposes for which it can be used, in terms of the cryptographic functions or operations (e.g. encryption or signature) that it is permitted to be used for. The TOE rejects any attempt to use the key for a purpose that is not permitted. The TOE also has an unambiguous definition of the subjects that are

permitted to access the key (and the purposes for which this access can be used) and allows this to be set to the granularity of an individual subject – these access constraints apply to use of the key even where the key value is not accessible. This objective means that the TOE also prevents unauthorised use of any cryptographic functions that use a key.

OT.KeyUseScope Defined scope for use of a key after authorisation
The TOE is required to define and apply clearly stated limits on when authorisation and re-authorisation are required in order for a secret key to be used[5]. For example, the TOE may allow secret keys to be used for a specified time period or number of uses after initial authorisation, or for may allow the key to be used until authorisation is explicitly rescinded. As another example, the TOE may implement a policy that requires re-authorisation before every use of a secret key.

Application Note 7 (PP)
Such limits on the use of a key after initial authorisation are termed "re-authorisation conditions" in this PP. A wide range of policies and re-authorisation conditions are allowed, and different policies may be applied to different types of secret key, but the re-authorisation conditions for all types of secret key must be unambiguously defined in the Security Target. The decision to use supported re-authentication conditions is made on the basis of the application context. Making appropriate use of re-authorisation conditions supports client applications in meeting their requirements for OE.DataContext and OE.AppSupport.

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

Application Note 8 (PP)
Protection of secret keys (as a specific type of sensitive data) is also subject to additional protection specified in other TOE objectives. Any requirements for secure storage and control of access to other types of client application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport. For example, if a client application uses the TOE to perform cryptographic functions on data that represent a passphrase value and the passphrase value is to be stored on the TOE, then the client application would need to use an appropriate encryption function before storing the data on the TOE.

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

Application Note 9 (PP)
Any requirements for integrity protection of client application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport.

OT.ImportExport Secure import and export of keys

[5] Any attempt to use the key in cryptographic functions that are not permitted for that key is addressed by OT.KeyUseConstraint.

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

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

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

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

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

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

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

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

5.2 Security Objectives for the Operational Environment

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

OE.ExternalData Protection of data outside TOE control
Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities shall provide appropriate protection for that data to a level

required by the application context and the risks in the deployment environment. This includes protection of data that is exported from, or imported to, the TOE (such as audit data and encrypted keys).
In particular, any backups of the TOE and its data shall be maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data shall not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data shall require at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

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

- Protection against loss or theft of the TOE or any of its externally stored assets.
- Inspections to deter and detect tampering (including attempts to access side-channels, or to access connections between physically separate parts of the TOE, or parts of the hardware appliance).
- Protection against the possibility of attacks based on emanations from the TOE (e.g. electromagnetic emanations) according to risks assessed for the operating environment.
- Protection against unauthorised software and configuration changes on the TOE and the hardware appliance.
- Protection to an equivalent level of all instances of the TOE holding the same assets (e.g. where a key is present as a backup in more than one instance of the TOE).

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

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

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

Application Note 10 (PP)
As noted for P.Audit in [PP_CMTS] section 3.4, the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.

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

6 Extended Components Definition

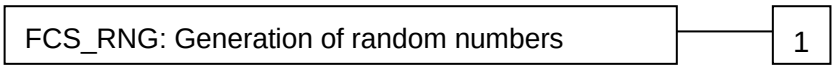
6.1 Generation of Random Numbers (FCS_RNG)

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

Family behaviour:

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

Component levelling:



Management:FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1	**Generation of random numbers**

Hierarchical to: No other components.

Dependencies: No dependencies.

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

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

Application Note 11 (PP)

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

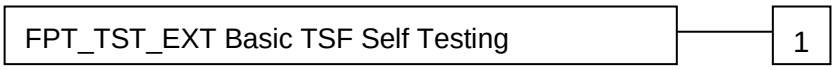
6.2 Basic TSF Self Testing (FPT_TST_EXT.1)

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

Family behaviour:

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

Component levelling:



Management:FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

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

- Indication that TSF self-test was completed.

FPT_TST_EXT.1	**Basic TSF Self Testing**

Hierarchical to: No other components.

Dependencies: No dependencies.

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

7 Security Requirements

This chapter gives the security functional requirements (SFR) and the security assurance requirements (SAR) for the TOE and the environment.
Security functional requirements components given in section 7.3 are drawn from Common Criteria part 2 [CC2]. Some security functional requirements represent extensions to [CC2], with a reasoning given in section 6. Operations for assignment, selection and refinement have been made.

The TOE security assurance requirements statements given in section 7.4 “Security Assurance Requirements” are drawn from the security assurance components from Common Criteria part 3 [CC3].

7.1 Typographical Conventions

The following conventions have been used by the Protection Profile [PP_CMTS] in the definitions of the SFRs and SARs:

- Refinements are denoted in one of two ways, depending on whether they add detail to an SFR or SAR (‘explanatory refinements’) or update the text of an SFR or SAR element (‘element refinements’). Explanatory refinements follow the SFR/SAR that they update and are marked by the word “**Refinement**” in **bold** followed by text describing the refinement. Element refinements are indicated by **bold** text within an SFR/SAR element, with the original text indicated in a footnote.
- Selections and assignments made in the PP are *italicized*, and the original text is indicated in a footnote. Selections and assignments that are left to be filled in by the Security Target author appear in square brackets with an indication that a selection or assignment is to be made, [selection:] or [assignment:], and the description of selection options or assignment description are *italicized*.

If an Application Note e. g. to an SFR was added by the Protection Profile, this is denoted by “**Application Note <nn> (PP)**”, with <nn> being the number of the Application Note as given in the Protection Profile. If an additional Application Note was added by the Security Target writer, this is denoted by “**Application Note <nn> (ST)**”.

7.2 SFR Architecture

7.2.1 SFR Relationships

The following diagrams Figure 6 and Figure 7 are taken from the PP (Figures 2 and 3 in [PP_CMTS]). They give a graphical presentation of the connections between the Security Functional Requirements (SFRs) from section 7.3 below and the underlying functional areas and operations that the TOE provides. The diagrams provide a context for SFRs that relates to their use in the TOE, whereas section 7.3 defines the SFRs grouped by the abstract class and family groupings in [CC2].

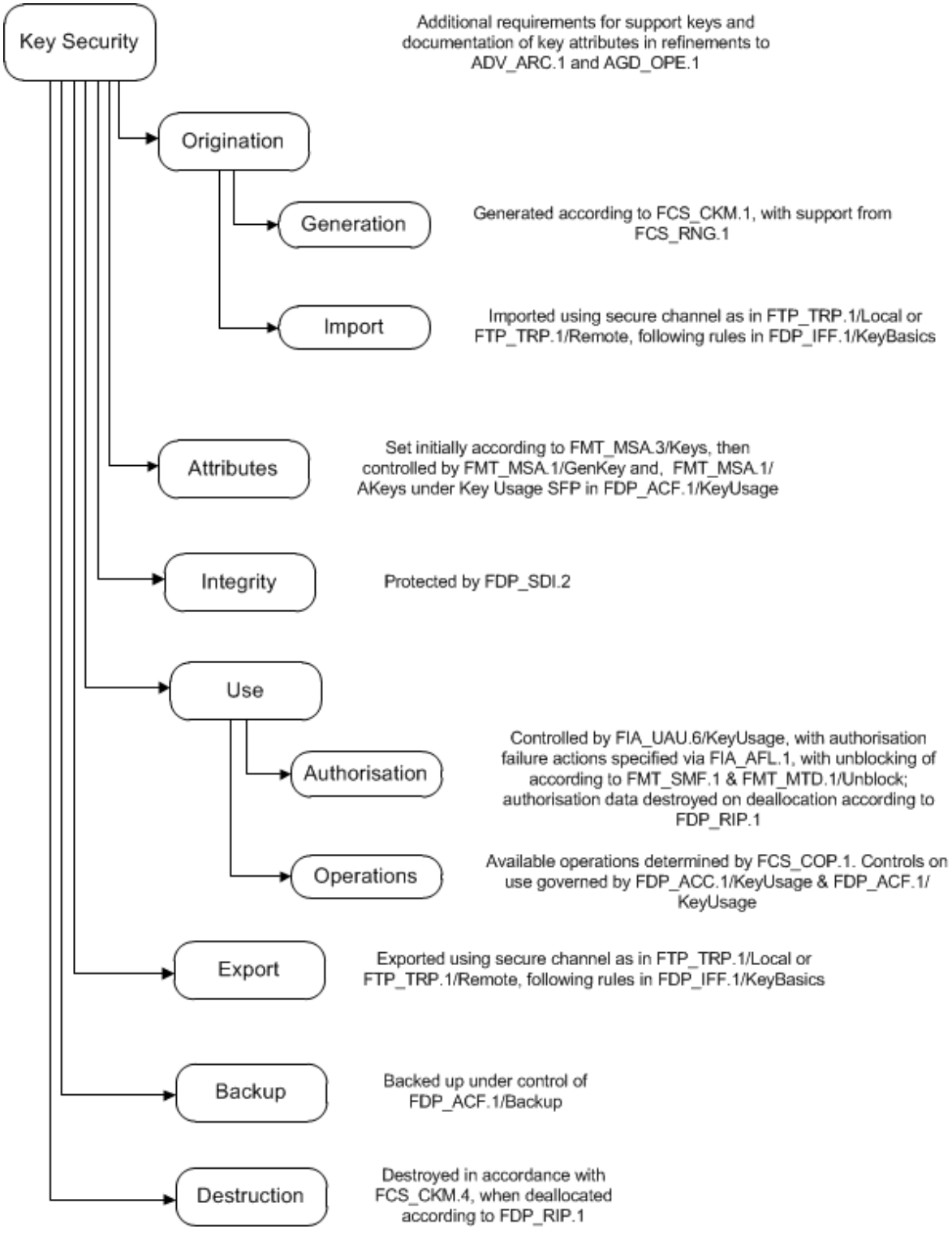


Figure 6: Architecture of Key Protection SFRs

Figure 5 illustrates the architecture of the SFRs from the PP that provide for the protection of cryptographic keys. In this ST most of the SFRs are iterated (see chapter 7.3), the security architecture therefore is enhanced in a natural way that all iterations of a specific SFR of the PP are responsible to implement the security requirements from the PP. For example, all

cryptographic keys are generated according to all iterations of SFR FCS_CKM.1 and with support from FCS_RNG.1, etc.

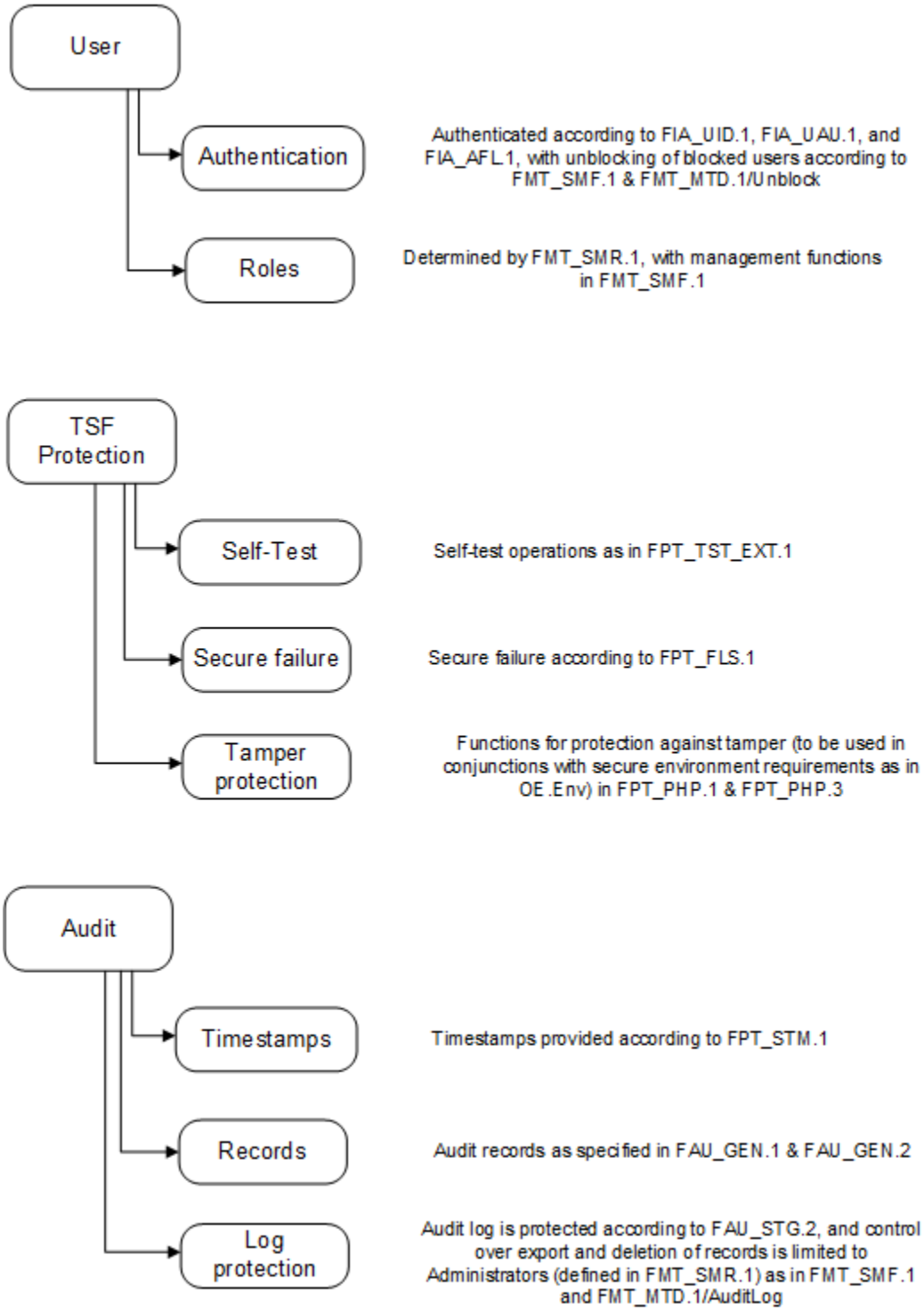


Figure 7: Architecture of User, TSF Protection & Audit SFRs

Figure 6 illustrates the architecture of the SFRs from the PP that provide for the requirements on the user concept, TSF protection and auditing. Also here, in this ST some of the SFRs are iterated (see chapter 7.3), the security architecture therefore is enhanced in a natural way

that dedicated iterations of a specific SFR of the PP are responsible to implement the security requirements from the PP. In particular, users are authenticated according to FIA_UID.1, FIA_UAU.1//UserAuth, and FIA_AFL.1//UserAuth, with unblocking of blocked users according to FMT_SMF.1 and FMT_MTD.1/Unblock//User.

7.2.2 SFRs and the Key Lifecycle

The generic lifecycle for a key is illustrated in Figure 8 that is taken from the PP (Figure 4 in [PP_CMTS]). It shows the methods by which a key may arrive in the TOE (import, generation or restore from backup), resulting in binding of a set of attributes to the key and storage of the key, and finally the ways in which a stored key may be processed (export, use in a cryptographic function, backup, or destruction). The SFRs related to each of these aspects are described below Figure 8.

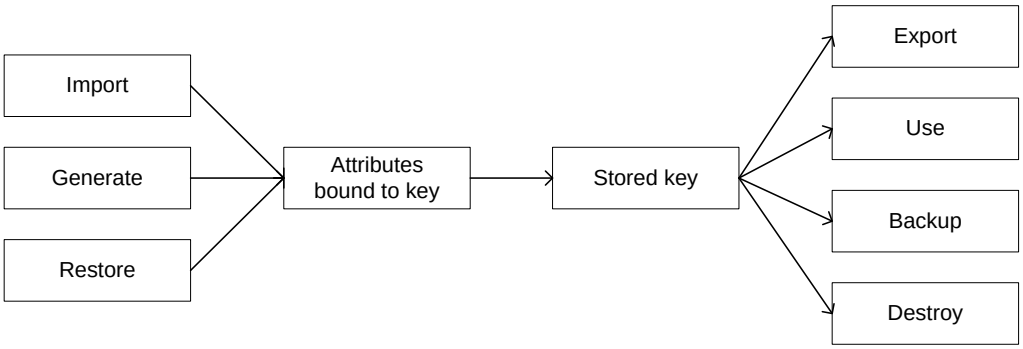


Figure 8: Generic Key Lifecycle and Related SFRs

Import:

- FDP_IFF.1/KeyBasics requires a secure channel (FTP_TRP.1/Local and FTP_TRP.1/External) and import in encrypted form or by using at least two components
- FAU_GEN.1 requires auditing of key import

Generate:

- FCS_CKM.1 (all iterations) requires approved algorithms
- FCS_RNG.1 defines requirements on random number generation
- FMT_MSA.3/Keys defines requirements on key attribute initialisation
- FAU_GEN.1 requires auditing of key generation (and of failure of RNG)

Restore:

- FDP_ACF.1/Backup requires that only an Administrator can restore from a backup, all backups must preserve confidentiality and integrity of keys (as appropriate to key type) and their attributes, and any restore must be under dual person control
- FAU_GEN.1 requires auditing of a restore (or of any integrity failure during a restore attempt)

Attributes bound to key:

- FMT_MSA.3/Keys defines requirements on key attribute initialisation
- FDP_ACF.1/KeyUsage, FMT_MSA.1/GenKeys (all iterations) and FMT_MSA.1/AKeys (all iterations) define requirements on key attribute modification

- FAU_GEN.1 requires auditing of changes to key attributes

Stored key:
- FDP_IFF.1/KeyBasics requires no plaintext access
- FDP_SDI.2 requires protection of the integrity of keys and their attributes
- FAU_GEN.1 requires auditing of integrity errors detected

Export:
- FDP_IFF.1/KeyBasics requires a secure channel (FTP_TRP.1), authorisation before key export, no export of Assigned Keys, export controlled by the Export flag attribute, and export in encrypted form
- FAU_GEN.1 requires auditing of key export

Use:
- FIA_AFL.1//KeyAuth requires blocking of access to a key on reaching an authorisation failure threshold (FDP_IFF.1/KeyBasics and FMT_MTD.1/Unblock//Key define requirements on unblocking)
- FDP_ACF.1/KeyUsage requires authorisation before use of a key and that the key can only be used as identified in its Key Usage attribute
- FIA_UAU.6/KeyAuth requires authorisation before initial use of a key and describes any additional requirements for re-authorisation conditions such as expiry of a time period or number of uses of a key (or when the key authorisation period has been explicitly ended)
- FDP_RIP.1 requires protection of authorisation data on de-allocation
- FDP_IFF.1/KeyBasics requires no access to intermediate values in any operation using a secret key
- FCS_COP.1 requires the use of approved algorithms
- FAU_GEN.1 requires auditing of authorisation failures (and blocking or unblocking)

Backup:
- FDP_ACF.1/Backup requires that only an Administrator can make a backup; all backups must preserve confidentiality and integrity of keys (as appropriate to their key type) and their attributes
- FAU_GEN.1 requires auditing of a backup

Destroy:
- FDP_RIP.1 requires keys to be protected on de-allocation
- FCS_CKM.4 requires key zeroisation on de-allocation
- FAU_GEN.1 requires auditing of key destruction

7.3 Security Functional Requirements

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

No.	SFR	Dependency
	FCS	**Cryptographic Support**
1.	FCS_CKM.1//AES	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
2.	FCS_CKM.1//RSA	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
3.	FCS_CKM.1//ECDSA	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
4.	FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
5.	FCS_COP.1//AES_Encryption_CBC	[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
6.	FCS_COP.1//AES_Encryption_OFB	[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
7.	FCS_COP.1//AES_Decryption_CBC	[FDP_ITC.1 I port of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
8.	FCS_COP.1//AES_Decryption_OFB	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
9.	FCS_COP.1//AES_CMAC	[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
10.	FCS_COP.1//AES_ECB	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

No.	SFR	Dependency
11.	FCS_COP.1//AES_GCM	[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
12.	FCS_COP.1//RSA_Sign	[FDP_ITC.1 I port of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
13.	FCS_COP.1//RSA_Verify	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
14.	FCS_COP.1//RSA_Encryption	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
15.	FCS_COP.1//RSA_Decryption	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
16.	FCS_COP.1//ECDSA_Sign	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
17.	FCS_COP.1//ECDSA_Verify	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
18.	FCS_COP.1//HMAC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
19.	FCS_COP.1//Hash	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
20.	FCS_COP.1//Diffie-Hellman	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

No.	SFR	Dependency
21.	FCS_COP.1//KeyDerivation	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
22.	FCS_RNG.1	No dependencies.
	FIA	**Identification and Authentication**
23.	FIA_UID.1	No dependencies.
24.	FIA_UAU.1//UserAuth	FIA_UID.1 Timing of identification
25.	FIA_UAU.1//KeyAuth	FIA_UID.1 Timing of identification
26.	FIA_AFL.1//UserAuth	FIA_UAU.1//UserAuth Timing of authentication
27.	FIA_AFL.1//KeyAuth	FIA_UAU.1//KeyAuth Timing of authentication
28.	FIA_UAU.6/KeyAuth	No dependencies
	FDP	**User data protection**
29.	FDP_IFC.1/KeyBasics	FDP_IFF.1 Simple security attributes
30.	FDP_IFF.1/KeyBasics	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
31.	FDP_ACC.1/KeyUsage	FDP_ACF.1 Security attribute based access control
32.	FDP_ACF.1/KeyUsage	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
33.	FDP_ACC.1/Backup	FDP_ACF.1 Security attribute based access control
34.	FDP_ACF.1/Backup	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
35.	FDP_SDI.2	No dependencies
36.	FDP_RIP.1	No dependencies
	FTP	**Trusted path/channels**
37.	FTP_TRP.1/Local	No dependencies
38.	FTP_TRP.1/External	No dependencies
	FPT	**Protection of the TSF**
39.	FPT_STM.1	No dependencies

No.	SFR	Dependency
40.	FPT_TST_EXT.1	No dependencies
41.	FPT_PHP.1	No dependencies
42.	FPT_PHP.3	No dependencies
43.	FPT_FLS.1	No dependencies
	FMT	**Security management**
44.	FMT_SMR.1	FIA_UID.1 Timing of identification.
45.	FMT_SMF.1	No dependencies
46.	FMT_MTD.1/Unblock//User	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
47.	FMT_MTD.1/Unblock//Key	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
48.	FMT_MTD.1/AuditLog	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
49.	FMT_MTD.1//SWUpdate	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
50.	FMT_MSA.1/GenKeys//AFlag	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
51.	FMT_MSA.1/GenKeys//ExportF	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
52.	FMT_MSA.1/GenKeys//AuthD	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
53.	FMT_MSA.1/GenKeys//None	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
54.	FMT_MSA.1/AKeys//AuthD	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
55.	FMT_MSA.1/AKeys//None	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
56.	FMT_MSA.3/Keys	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles

No.	SFR	Dependency
	FAU	**Security audit data generation**
57.	FAU_GEN.1	FPT_STM.1 Reliable time stamps
58.	FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification
59.	FAU_STG.2	FAU_GEN.1 Audit data generation

Table 2: Security Functional Requirements

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

7.3.1 Cryptographic Support (FCS)

FCS_CKM.1//AES Cryptographic key generation

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

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

FCS_CKM.1//RSA Cryptographic key generation

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

FCS_CKM.1.1//RSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *RSA key pair generation with pre-defined or given public exponent*[9] and specified cryptographic key sizes *of minimum 2048*[1] *and maximum 8192 bits modulus length*[10] that meet the

[6] [assignment: cryptographic key generation algorithm]

[7] [assignment: cryptographic key sizes]

[8] [assignment: list of standards]

[9] [assignment: cryptographic key generation algorithm]

[10] [assignment: cryptographic key sizes]

following: *generation of RSA key pairs according to [SOG-IS-Crypto] section 4.1*[11].

FCS_CKM.1//ECDSA Cryptographic key generation

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

FCS_CKM.1.1//ECDSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *ECDSA key pair generation with given elliptic curve domain parameters*[12] and specified cryptographic key sizes *of minimum 224 bits*[13] that meet the following: *ECDSA key pair generation for ECC domain parameters Curve P-224, Curve P-256, Curve P-384 or Curve P-521 as specified in [FIPS 186-4] appendix 6, or brainpoolP224r1, brainpoolP256r1, brainpoolP320r1, brainpoolP384r1, brainpoolP512r1, brainpoolP224t1, brainpoolP256t1, brainpoolP320t1, brainpoolP384t1 or brainpoolP512t1 as specified in [ECCBP] chapter 10, or curve FRP256v1 as specified in [ANSSI] and with random number generation according to FCS_RNG.1*[14].

Application Note 12 (PP)

The Security Target must include all key generation operations that are intended to support TSP operations using one or more iterations of FCS_CKM.1.

The relevant authorities and endorsements for completion of the SFRs are determined by the context of the client applications that use the TOE. For digital signatures within the European Union this is as indicated in [Regulation] and an exemplary list of algorithms and parameters is given in [TS 119 312] or *[SOG-IS-Crypto] (see also [PP_CMTS] section 1.3.1.3).*

Note that key generation needs to be linked to the setting of security attributes of a key (including the link to a subject who owns the key, via the setting of authorisation data) as in FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys,

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]

[11] [assignment: list of standards]

[12] [assignment: cryptographic key generation algorithm]

[13] [assignment: cryptographic key sizes]

[14] [assignment: list of standards]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *zeroisation*[15] that meets the following: *overwriting the key by zeroising in case of secret or private keys*[16].

Application Note 13 (PP)

The Security Target must specify the method(s) of secure destruction of all secret keys and all support keys[17]*, and must ensure that all are covered by a secure destruction method. If necessary, then more than one iteration of FCS_CKM.4 may be included to describe different standards for secure deletion. The 'list of standards' in the final assignment may be met in the Security Target by simply providing a description of the action taken to zeroise the keys rather than referencing an external standard.*

Application Note 2 (ST)

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

Encrypted secret and private keys are destroyed by deleting the logical address, and by zeroizing the encryption key in case of a physical attack: For permanent storage inside the TOE, the TOE enforces all secret and private keys to be stored encrypted with the TOE's internal Master Key. The commands for key deletion delete the encrypted secret and private keys by deletion of the logical addresses, respectively. After that it is no longer possible to address the memory areas of the encrypted keys via the TOE interface.

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

This ensures secure storage and destruction also for encrypted secret and private keys.

FCS_COP.1//AES_Encryption_CBC Cryptographic operation

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

FCS_COP.1.1//AES_Encryption_CBC The TSF shall perform *encryption*[18] in accordance with a specified cryptographic algorithm *AES block cipher in CBC mode with ISO 7816-4 or PKCS#5 padding*[19] and cryptographic key sizes *of*

[15] [PP_CMTS] [assignment: cryptographic key destruction method]

[16] [assignment: list of standards]

[17] See the description of 'support keys' in the refinement of ADV_ARC.1 in section 7.4.1.

[18] [assignment: list of cryptographic operations]

[19] [assignment: cryptographic algorithm]

16, 24 or 32 bytes length[20] that meet the following: *[NIST SP 800-38A] chapter 6.2, [FIPS 197] chapter 5 (AES block cipher in CBC mode)*[21].

FCS_COP.1//AES_Encryption_OFB Cryptographic operation

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

FCS_COP.1.1//AES_Encryption_OFB The TSF shall perform *encryption*[22] in accordance with a specified cryptographic algorithm *AES block cipher in OFB mode with ISO 7816-4 or PKCS#5 padding*[23] and cryptographic key sizes *of 16, 24 or 32 bytes length*[24] that meet the following: *[NIST SP 800-38A] chapter 6.4, [FIPS 197] chapter 5 (AES block cipher in OFB mode)*[25].

FCS_COP.1//AES_Decryption_CBC Cryptographic operation

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

FCS_COP.1.1//AES_Decryption_CBC The TSF shall perform *decryption*[26] in accordance with a specified cryptographic algorithm *AES block cipher in CBC mode with ISO 7816-4 or PKCS#5 padding*[27] and cryptographic key sizes *of 16, 24 or 32 bytes length*[28] that meet the following: *[NIST SP 800-38A] chapter 6.2, [FIPS 197] chapter 5 (AES block cipher in CBC mode)*[29].

FCS_COP.1//AES_Decryption_OFB 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

[20] [assignment: cryptographic key sizes]
[21] [assignment: list of standards]
[22] [assignment: list of cryptographic operations]
[23] [assignment: cryptographic algorithm]
[24] [assignment: cryptographic key sizes]
[25] [assignment: list of standards]
[26] [assignment: list of cryptographic operations]
[27] [assignment: cryptographic algorithm]
[28] [assignment: cryptographic key sizes]
[29] [assignment: list of standards]

Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1//AES_Decryption_OFB The TSF shall perform *decryption*[30] in accordance with a specified cryptographic algorithm *AES block cipher in OFB mode with ISO 7816-4 or PKCS#5 padding*[31] and cryptographic key sizes *of 16, 24 or 32 bytes length*[32] that meet the following: *[NIST SP 800-38A] chapter 6.4, [FIPS 197] chapter 5 (AES block cipher in OFB mode)*[33].

FCS_COP.1//AES_CMAC

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

FCS_COP.1.1//AES_CMAC The TSF shall perform *data integrity protection*[34] in accordance with a specified cryptographic algorithm *AES CMAC*[35] and *cryptographic key sizes of 16, 24 or 32 bytes length*[36] that meet the following: *[NIST SP 800-38B]*[37].

FCS_COP.1//AES_ECB Cryptographic operation

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

FCS_COP.1.1//AES_ECB The TSF shall perform *encryption and decryption*[38] in accordance with a specified cryptographic algorithm *AES block cipher in ECB mode*[39] and cryptographic key sizes *of 16, 24 or 32 bytes length*[40] that meet the following: *[NIST SP 800-38A] chapter 6.1, [FIPS 197] chapter 5 (AES block cipher in ECB mode)*[41].

[30] [assignment: list of cryptographic operations]
[31] [assignment: cryptographic algorithm]
[32] [assignment: cryptographic key sizes]
[33] [assignment: list of standards]
[34] [assignment: list of cryptographic operations]
[35] [assignment: cryptographic algorithm]
[36] [assignment: cryptographic key sizes]
[37] [assignment: list of standards]
[38] [assignment: list of cryptographic operations]
[39] [assignment: cryptographic algorithm]
[40] [assignment: cryptographic key sizes]
[41] [assignment: list of standards]

Application Note 3 (ST)

Encryption and decryption in accordance with FCS_COP.1//AES_ECB can only be invoked by an internal SAM. It is not provided as a cryptographic service at the external TOE interface.

FCS_COP.1//AES_GCM Cryptographic operation

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

FCS_COP.1.1//AES_GCM The TSF shall perform *authenticated encryption and decryption*[42] in accordance with a specified cryptographic algorithm *AES block cipher in GCM mode*[43] and cryptographic key sizes *of 16, 24 or 32 bytes length*[44] that meet the following: *[NIST SP 800-38A] chapter 7, [FIPS 197] chapter 5 (AES block cipher in GCM mode)*[45].

FCS_COP.1//RSA_Sign Cryptographic operation

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

FCS_COP.1.1//RSA_Sign The TSF shall perform *the generation of a digital signature*[46] in accordance with a specified cryptographic algorithm *RSA signature scheme with appendix according to [PKCS#1], RSASSA-PSS or RSASSA-PKCS-v1_5,*[47] and cryptographic key sizes *of minimum 2048*[1] *and maximum 8192 bits modulus length*[48] that meet the following: *[PKCS#1], chapters 8.1.1 or 8.2.1*[49].

FCS_COP.1//RSA_Verify Cryptographic operation

Hierarchical to: No other components.

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

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

FCS_COP.1.1//RSA_Verify The TSF shall perform *the verification of a digital signature*[50] in accordance with a specified cryptographic algorithm *RSA signature scheme with appendix according to [PKCS#1], RSASSA-PSS or RSASSA-PKCS1-V1_5,*[51] and cryptographic key sizes *of minimum 2048*[1] [52] *and maximum 8192 bits modulus length*[53] that meet the following: *[PKCS#1], chapters 8.1.2 or 8.2.2*[54].

FCS_COP.1//RSA_Encryption Cryptographic operation

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

FCS_COP.1.1//RSA_Encryption The TSF shall perform *encryption*[55] in accordance with a specified cryptographic algorithm *RSA encryption scheme according to [PKCS#1], RSAES-OAEP or RSAES-PKCS-v1_5,*[56] and cryptographic key sizes *of minimum 2048*[1] *and maximum 8192 bits modulus length*[57] that meet the following: *[PKCS#1], chapters 7.1.1 or 7.2.1*[58].

FCS_COP.1//RSA_Decryption Cryptographic operation

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

FCS_COP.1.1//RSA_Decryption The TSF shall perform *decryption*[59] in accordance with a specified cryptographic algorithm *RSA encryption scheme according to [PKCS#1], RSAES-*

[50] [assignment: list of cryptographic operations]
[51] [assignment: cryptographic algorithm]
[52] For legacy usage shorter key sizes are allowed for signature verification
[53] [assignment: cryptographic key sizes]
[54] [assignment: list of standards]
[55] [assignment: list of cryptographic operations]
[56] [assignment: cryptographic algorithm]
[57] [assignment: cryptographic key sizes]
[58] [assignment: list of standards]
[59] [assignment: list of cryptographic operations]

OAEP or RSAES-PKCS-v1_5,[60] and cryptographic key sizes *of minimum 2048*[1,61] *and maximum 8192 bits modulus length*[62] that meet the following: *[PKCS#1], chapters 7.1.2 or 7.2.2*[63].

FCS_COP.1//ECDSA_Sign Cryptographic operation

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

FCS_COP.1.1//ECDSA_Sign The TSF shall perform *the generation of a digital signature*[64] in accordance with a specified cryptographic algorithm *ECDSA*[65] and cryptographic key sizes *of minimum 224 bits*[66] that meet the following: *signature generation according to [ANSI-X9.62] with signature keys based on ECC domain parameters Curve P-224, Curve P-256, Curve P-384 or Curve P-521 as specified in [FIPS 186-4] appendix D, or brainpoolP224r1, brainpoolP256r1, brainpoolP320r1, brainpoolP384r1, brainpoolP512r1, brainpoolP224t1, brainpoolP256t1, brainpoolP320t1, brainpoolP384t1 or brainpoolP512t1 as specified in [ECCBP] chapter 10, or curve FRP256v1 as specified in [ANSSI]*[67].

FCS_COP.1//ECDSA_Verify cryptographic operation

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

FCS_COP.1.1//ECDSA_Verify The TSF shall perform *the verification of a digital signature*[68] accordance with a specified cryptographic algorithm *ECDSA*[69] and cryptographic key sizes *of minimum 224 bits*[70] that meet the following: *signature generation according to [ANSI-X9.62] with signature keys based on ECC domain parameters curve P-224, curve P-256, curve P-384 or curve P-*

[60] [assignment: cryptographic algorithm]
[61] For legacy usage shorter key sizes are allowed for signature verification
[62] [assignment: cryptographic key sizes]
[63] [assignment: list of standards]
[64] [assignment: list of cryptographic operations]
[65] [assignment: cryptographic algorithm]
[66] [assignment: cryptographic key sizes]
[67] [assignment: list of standards]
[68] [assignment: list of cryptographic operations]
[69] [assignment: cryptographic algorithm]
[70] [assignment: cryptographic key sizes]

521 as specified in [FIPS 186-4] appendix D, or curve brainpoolP224r1, brainpoolP256r1, brainpoolP320r1, brainpoolP384r1, brainpoolP512r1, brainpoolP224t1, brainpoolP256t1, brainpoolP320t1, brainpoolP384t1 or brainpoolP512t1 as specified in [ECCBP] chapter 10, or curve FRP256v1 as specified in [ANSSI][71].

FCS_COP.1//HMAC Cryptographic operation

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

FCS_COP.1.1//HMAC The TSF shall perform *HMAC calculation*[72] in accordance with a specified cryptographic algorithm *HMAC*[73] and cryptographic key sizes *between 4 and 1024 bytes*[74] that meet the following: *[FIPS 198] and [RFC 2104], with hash value calculation according to FCS_COP.1//Hash*[75].

Application Note 4 (ST)

HMAC calculation in accordance with FCS_COP.1.1//HMAC and cryptographic key size smaller than 13 bytes can only be used in the context of command authentication. It is not provided as a cryptographic service.

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

FCS_COP.1//Hash Cryptographic operation

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

FCS_COP.1.1//Hash The TSF shall perform *hash value calculation*[76] in accordance with a specified cryptographic algorithm *SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384 or SHA3-512*[77] and cryptographic key sizes

[71] [assignment: list of standards]

[72] [assignment: list of cryptographic operations]

[73] [assignment: cryptographic algorithm]2

[74] [assignment: cryptographic key sizes]

[75] [assignment: list of standards]

[76] [assignment: list of cryptographic operations]

[77] [assignment: cryptographic algorithm]

none[78] that meet the following: *[FIPS 180-4] chapter 6 for SHA-2, and [FIPS 202] for SHA-3*[79].

FCS_COP.1//Diffie-Hellman Cryptographic operation

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

FCS_COP.1.1//Diffie-Hellman The TSF shall perform *Diffie-Hellmann Key agreement*[80] in accordance with a specified cryptographic algorithm *Diffie-Hellman protocol*[81] and cryptographic key sizes *3072*[82] that meet the following: *[PKCS#3], chapter 6.8*[83].

Application Note 5 (ST)

Diffie-Hellman Key agreement in accordance with FCS_COP.1.1//Diffie-Hellman can only be used in the context of establishing a Secure Messaging session (trusted channel according to FTP_TRP.1/Local or FTP_TRP.1/External). It is not provided as a cryptographic service.

FCS_COP.1//KeyDerivation Cryptographic operation

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

FCS_COP.1.1//KeyDerivation The TSF shall perform *Key Derivation*[84] in accordance with a specified cryptographic algorithm *KDF in Feedback Mode with HMAC*[85] and cryptographic key sizes *4-1024*[86] that meet the following: *[NIST SP 800-108], chapter 5.2, with HMAC calculation according to FCS_COP.1//HMAC*[87].

[78] [assignment: cryptographic key sizes]
[79] [assignment: list of standards]
[80] [assignment: list of cryptographic operations]
[81] [assignment: cryptographic algorithm]
[82] [assignment: cryptographic key sizes]
[83] [assignment: list of standards]
[84] [assignment: list of cryptographic operations]
[85] [assignment: cryptographic algorithm]
[86] [assignment: cryptographic key sizes]
[87] [assignment: list of standards]

Application Note 6 (ST)

Key Derivation in accordance with FCS_COP.1.1//KeyDerivation can only be used in the context of establishing a Secure Messaging session (trusted channel according to FTP_TRP.1/Local or FTP_TRP.1/External) and for the backup of cryptographic keys (FDP_ACC.1/Backup, FDP_ACF.1/Backup). It is not provided as a cryptographic service.

Application Note 14 (PP)

The Security Target must include all cryptographic functions that are intended to support TSP operations using one or more iterations of FCS_COP.1. This includes cryptographic operations for digital signatures and seals, implementing trusted paths (FTP_TRP.1) and secure channels (FTP_TRP.1), key encryption (e.g. FDP_IFF.1/KeyBasics), and any backups (FDP_ACF.1/Backup) that the TOE creates. If the TOE supports software or firmware updates then the iterations must include the cryptographic operations used to support the validation of digital signatures on the updates as described in the refinement to ADV_ARC.1 in section 7.4.1.

The relevant authorities and endorsements for completion of each of these iterations are determined by the context of the client applications that use the TOE. For digital signatures and seals within the European Union this is as indicated in [Regulation] and an exemplary list of algorithms and parameters is given in [TS 119 312] or [SOG-IS-Crypto] *(see also [PP_CMTS], section 1.3.1.3).*

FCS_RNG.1 Random number generation

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

FCS_RNG.1.1 The TSF shall provide a *hybrid deterministic*[88] random number generator that implements: *RNG class DRG.4 of [AIS 20/31] chapter 4.9*[89]

(DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2 as random source[90]*.*

(DRG.4.2) The RNG provides forward secrecy.

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

(DRG.4.4) The RNG provides enhanced forward secrecy on condition[91] *that 1000 requests for pseudo random bits have been made after last entropy input during instantiation or reseeding*[92]

(DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2[93]

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

[89] [assignment: list of security capabilities]

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

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

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

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

FCS_RNG.1.2 The TSF shall provide *octets of bits*[94] that meet:

(DRG.4.6) The RNG generates output for which $7 \cdot 10^7$[95] *strings of bit length 128 are mutually different with probability 0.9998.*[96]

(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A[97, 98]*.*

Application Note 15 (PP)

For more information on the selections and assignments see the SFR definition in section 6.1. The Security Target describes the uses made of the RNG and its relationship to other SFRs such as FCS_CKM.1, and to any random number generation function/service made available to users or clients applications.

7.3.2 Identification and Authentication (FIA)

FIA_UID.1 Timing of identification

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

FIA_UID.1.1 The TSF shall allow

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

(2) usage of commands where no user authentication is needed, including requests for the status of the TOE[99]*,*

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

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

Application Note 16 (PP)

The 'list of additional TSF-mediated actions' may be left empty (equivalent to an assignment of 'None') if applicable.

Application Note 7 (ST)

An internal SAM is identified, authenticated and authorised when loaded to the CryptoServer by verification of its signature with the CryptoServer CP5 Module Signature Key. It does not need further identification or authentication when communicating with the TOE, as noted in [PP_CMTS] section 1.3.1, Application Note 6. (See also Application Note 6 (PP) and Application Note 29 (PP) in this ST.)

---

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

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

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

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

98 [AIS 20/31]: [assignment: *a defined quality metric*]

99 [assignment: *list of additional TSF mediated actions*]

FIA_UAU.1//UserAuth Timing of authentication

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

FIA_UAU.1.1//UserAuth The TSF shall allow

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

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

(3) *usage of commands where no user authentication is needed, including requests for the status of the TOE*[100]*.*

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

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

Application Note 8 (ST)

An internal SAM is identified, authenticated and authorised when loaded to the CryptoServer by verification of its signature with the CryptoServer CP5 Module Signature Key. It does not need further identification or authentication when communicating with the TOE, as noted in [PP_CMTS] section 1.3.1, Application Note 6. (See also Application Note 6 (PP) and Application Note 29 (PP) in this ST.)

FIA_UAU.1//KeyAuth Timing of authentication

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

FIA_UAU.1.1//KeyAuth The TSF shall allow

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

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

(3) *Authentication of the user by means of TSF required by FIA_UAU.1//UserAuth,*

(4) *usage of commands where user authentication is needed but where no authorisation for access to a secret key is needed*[101]

on behalf of the user to be performed before the user is ~~authenticated~~*authorised for access to a secret key*. [102]

FIA_UAU.1.2//KeyAuth The TSF shall require each user to be successfully ~~authenticated~~ *authorised for access to a secret key* before allowing any other TSF-mediated actions *that require access to the secret key* on behalf of that user.[103]

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

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

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

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

Application Note 17 (PP)

The Security Target must separately identify any different types of identification and authentication, e.g. for Administrators, local users, application users, using separate iterations of the FIA_UID.1 and FIA_UAU.1 SFRs where the methods differ. The Security Target must also separately identify the difference between authentication of users and authorisation for use of keys as required for FIA_UAU.6/KeyAuth. Separate iterations of FIA SFRs may be necessary to capture these separate cases.

The 'list of additional TSF-mediated actions' in FIA_UAU.1.1 may be left empty (equivalent to an assignment of 'None') if applicable.

Application Note 9 (ST)

For FIA_UAU.1 different types of identification and authentication via the external interface, e.g. for Administrators, local users, application users are not needed because they all use the same authentication mechanism.

An internal SAM is identified, authenticated and authorised when loaded to the CryptoServer by verification of its signature with the CryptoServer CP5 Module Signature Key. It does not need further identification or authentication when communicating with the TOE, as noted in [PP_CMTS] section 1.3.1, Application Note 6. (See also Application Note 6 (PP) and Application Note 29 (PP) in this ST.)

The internal SAM is allowed to request usage of a secret key for signature generation either by calling the generic internal TOE interface or by calling the internal SAEK signature interface offered by the TOE. In the first case, usage of the key will be rejected unless the SAM presents explicit key authorisation data (KRAD) to the TOE for verification. In the second case, the SAM has the responsibility of performing and validating key authorisation (see Application Note 1 (ST)) as mandated in the TOE Guidance. The TOE will implicitly derive the result of key authorisation from the SAM.

FIA_AFL.1//UserAuth Authentication failure handling

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

FIA_AFL.1.1//UserAuth The TSF shall detect when *five*[104] unsuccessful authentication **~~or authorisation~~** attempts occur related to *consecutive failed authentication ~~or authorisation~~ attempts*.[105]

FIA_AFL.1.2//UserAuth When the defined number of unsuccessful authentication **~~or~~ ~~authorisation~~** attempts has been *met*[106], the TSF shall *block access to the user account*[107] *until unblocked by a User Administrator*[108] [109].

---

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

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

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

[107] *[assignment: description of the relevant functionality]*

[108] *[selection: unblocked by [assignment: identification of the authorised subject or role], a time period [assignment: time period] has elapsed]*

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

FIA_AFL.1//KeyAuth Authentication failure handling

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

FIA_AFL.1.1//KeyAuth The TSF shall detect when *five*[104] unsuccessful ~~authentication or~~ **authorisation** attempts occur related to *consecutive failed ~~authentication or~~ authorisation attempts*[105].

FIA_AFL.1.2//KeyAuth When the defined number of unsuccessful ~~authentication or~~ **authorisation** attempts has been *met*[106], the TSF shall *block access to the key*[107] *until unblocked by a Key Manager*[108] [109].

Application Note 18 (PP)

The Security Target must separately identify the different types of authentication or authorisation to which failure responses apply, and this should include all of the different types of authentication identified for FIA_UAU.1 and failed authorisation attempts related to attempts to use keys as in FIA_UAU.6/KeyAuth. Where different authentication/authorisation failure responses apply then the SFR should be iterated.

The unblocking of functionality blocked as described in each iteration of FIA_AFL.1.2 must be described in a corresponding iteration of FMT_MTD.1 (cf. section 7.3.6).

Application Note 10 (ST)

If an internal SAM invokes the SAEK signature interface, prior successful key authorisation is implicitly derived by the TOE (see Application Note 1 (ST) for further explanation).

FIA_UAU.6/KeyAuth Re-authenticating

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

FIA_UAU.6.1/KeyAuth The TSF shall **authorise and re-authorise**[110] the user **for access to a secret key** under the conditions

(1) Authorisation in order to be granted initial access to the key; and
(2) Re-authorisation of the key under the following conditions:
- *after the number of uses of the secret key (as specified in the secret key's attributes) for which the secret key was last authorised has already been made;*
- *after explicit rescinding of previous authorisation for access to the secret key*[111].

Application Note 19 (PP)

Use of a key requires an initial authorisation by presentation of the correct authorisation data. Subsequent uses may require re-authorisation on every use (in this case 'Authorisation on

[110] [PP_CMTS] re-authenticate

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

every subsequent access to the key' is selected in FIA_UAU.6.1/KeyAuth (2)), or else the TOE may allow some uses of the key without further authorisation until one of the specified re-authorisation conditions occurs.

The TOE may also allow different re-authorisation conditions for different types of secret key. The types of secret keys may be identified (in the first assignment in (2)) as individual keys, or in terms of a generic definition (e.g. 'all non-Assigned keys'). Where different re-authorisation conditions apply to different types of key then the second assignment in (2) may be used to specify the other types of key and the conditions that apply to them in a similar manner.

The explicit rescinding of an authorisation period in (2) ensures that client applications or users can decide to revoke a previous authorisation in (2) that may still be in force. If the TOE intends to allow unlimited uses of a secret key after initial authorisation, until authorisation is rescinded by a client application or user, then the selection 'after explicit rescinding of previous authorisation for access to the secret key' is chosen in the Security Target without any accompanying selections for time periods or number of uses. The Security Target describes the method or methods used for such rescinding (such as particular API commands).

It is the responsibility of the client application to make appropriate use of these any re-authentication conditions according to the application context (cf. OE.DataContext and OE.AppSupport).

Each 'use' of a key is expected to relate to one cryptographic function carried out with the key. If there are circumstances where a different interpretation may be placed on the 'use' of a key then this must be identified and explained in the Security Target and the Operational Guidance. The intention here is to make clear any situations that are relevant to a key owner who can be held responsible for use of the key (such as any case where a single authorisation for use of a key could allow the creation of more than one signature using the authorised key). Note that in order to make qualified electronic signatures under [Regulation] then the user/application must be able to precisely control the signatures that can be made under each authorisation.
Actions taken by the TOE in the case of successive authorisation failures must be specified using an iteration of FIA_AFL.1.

7.3.3 User Data Protection (FDP)

FDP_IFC.1/KeyBasics Subset information flow control

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

FDP_IFC.1.1/KeyBasics The TSF shall enforce the *Key Basics SFP*[112] on
(1) subjects: all
(2) information: keys
(3) operations: all[113].

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

[113] [PP_CMTS] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]*

FDP_IFF.1/KeyBasics Simple security attributes

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

FDP_IFF.1.1/KeyBasics The TSF shall enforce the *Key Basics SFP*[114] based on the following types of subject and information security attributes:

(1) whether a key is a secret or a public key

(2) whether a secret key is an Assigned Key

(3) whether channels selected to export keys are secure

(4) the value of the Export Flag of a key[115]*.*

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

(1) Export of secret keys shall only be allowed provided that the secret key is not an Assigned Key, that the secret key is encrypted, and that a secure channel (providing authentication and integrity protection) is used for the export

(2) Public keys shall always be exported with integrity protection of their key value and attributes

(3) Keys shall only be imported over a secure channel (providing authentication and integrity protection)

(4) A secret key can only be imported if it is a non-Assigned key

(5) Secret keys shall only be imported in encrypted form or using split-knowledge procedures requiring at least two key components to reconstruct the key, with key components supplied by at least two separately authenticated users

(6) Unblocking access to a key shall not allow any subject other than those authorised to access the key at the time when it was blocked[116]*.*

Application Note 20 (PP)
A secure channel for export of keys in FDP_IFF.1.2/KeyBasics (1) or for import of keys in FDP_IFF.1.2/KeyBasics (3) is one that meets the requirements of FTP_TRP.1/Local or FTP_TRP.1/External.
The encrypted form required for keys imported or exported over a secure channel requires encryption of the key itself, in addition to any encryption provided by the secure channel. Unblocking a key as in FDP_IFF.1.2/KeyBasics (6) is intended only to restore the ability of subjects to authorise for access to a key by presenting the correct authorisation data. As noted for FMT_MTD.1/Unblock//Key, the subject who unblocks the key must not be able also to use the key as a result of the unblocking (unless of course they are able to supply the correct authorisation data). This is a part of ensuring that sole control of secret keys can be achieved.

---

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

[115] [PP_CMTS] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

[116] [PP_CMTS] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

FDP_IFF.1.3/KeyBasics The TSF shall enforce the **following additional information flow control rules**: *none*[117].

FDP_IFF.1.4/KeyBasics The TSF shall explicitly authorise an information flow based on the following rules: *none*[118].

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

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

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

(3) No subject shall be allowed to export an Assigned Key.

(4) No subject shall be allowed to export a secret key without submitting the correct authorisation data for the key

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

(6) A key with an Export Flag value marking it as non-exportable shall not be exported[119]

Application Note 21 (PP)
The requirements of FDP_IFF.1/KeyBasics apply regardless of how the key is stored by the TOE, including when the key is externally stored (cf. [PP_CMTS]*, section 1.3.1.2).*

Direct access to a key value in FDP_IFF.1.5/KeyBasics (1) is access that makes the value available for reading or modification – this includes operations that would subsequently allow reading or modification of the key (e.g. making a copy of the key with different attributes, or with a different object type that would then allow direct read access). Note that this PP assumes that key values are never modified after they have been generated.
Export of a key as in FDP_IFF.1.5/KeyBasics (1), (2), (4) and (6) is not the same as backup (governed by FDP_ACF.1/Backup) or external storage of keys under continuing TOE control (governed by other parts of the Key Basics SFP in FDP_IFF.1/KeyBasics, and the Key Usage SFP in FDP_ACF.1/KeyUsage). Thus an Export Flag of 'non-exportable' does not prevent backup or external storage of the keys under continuing TOE control.
The Security Target and/or Operational Guidance shall specify how any attributes not supplied with an imported key are set when the key is imported (or alternatively how such keys are rejected). Similarly, the Security Target and/or Operational Guidance shall describe how the key's attributes are represented when exported, so that their meaning can be understood by the receiver.
If the TOE does not provide facilities to import or export keys then the relevant part of the SFR is trivially satisfied, and this should be stated in the Security Target.

FDP_ACC.1/KeyUsage Subset access control

[117] [PP_CMTS] [assignment: *additional information flow control SFP rules*]

[118] [PP_CMTS] [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

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

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

FDP_ACC.1.1/KeyUsage The TSF shall enforce the Key Usage SFP[120] to objects based on the following

(1) Subjects: all;

(2) Object: Keys

(3) Operations: all[121]

FDP_ACF.1/KeyUsage Security attribute based access control

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

FDP_ACF.1.1/KeyUsage The TSF shall enforce the Key Usage SFP[122] to objects based on the following:

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

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

(3) the cryptographic function that is attempting to use the secret key[123]*.*

Application Note 22 (PP)

Whether a subject is currently authorised for access to a secret key is determined by whether the subject has submitted the correct authorisation data for the key, and whether this authorisation is yet subject to one or more of the re-authorisation conditions in FIA_UAU.6/KeyAuth.

Whether a subject is currently authorised to change the attributes of a secret key is determined by the iterations of FMT_MSA.1 in section 7.3.6.

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

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

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

(3) Only cryptographic functions permitted by the secret key's Key Usage attribute shall be carried out using the secret key[124]*.*

---

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

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

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

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

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

Application Note 23 (PP)

FDP_ACF.1.2/KeyUsage (1) refers to controls over changing attributes that are specified in more detail in the iterations of FMT_MSA.1.

FDP_ACF.1.2/KeyUsage (2) requires that a key can only be used when the relevant subject has been authorised either by presenting the correct authorisation data for the key as part of the request for the operation or else the authorisation has previously been presented by the subject and the current use of the key does not yet require re-authorisation according to FIA_UAU.6/KeyAuth (meaning that the current usage is therefore within the usage constraints for time and number of uses since the last authorisation of use of the key). The reference to use of the plaintext value of the key does not imply that a subject has access to that value, only that it can be used to carry out operations within the TOE – reference to operations of this sort are thus distinguished from operations that may use an encrypted form of a secret key (e.g. for external storage of keys) and that are not necessarily restricted in this way.

FDP_ACF.1.3/KeyUsage The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*[125].

FDP_ACF.1.4/KeyUsage The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[126].

Application Note 24 (PP)

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

FDP_ACC.1/Backup Subset access control

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

FDP_ACC.1.1/Backup The TSF shall enforce the Backup SFP[127] on

(1) subjects: all

(2) objects: keys

(3) operations: backup, restore[128]*.*

FDP_ACF.1/Backup Security attribute based access control

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

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

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

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

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

FDP_ACF.1.1/Backup The TSF shall enforce the Backup SFP[129] to objects based on the following:

(1) whether the subject is an administrator[130].

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

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

(2) Any restore of the TSF shall only be possible under at least dual person control, with each person being an administrator

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

(4) Any backup and restore operations shall preserve the integrity of the key attributes, and the binding of each set of attributes to its key[131].

Application Note 25 (PP)

Preserving the binding of a set of attributes to its key (in FDP_ACF.1.2/Backup (4)) means that it is not possible for the attributes to be changed during a backup operation, or by modification of the backup data while it is away from the TSF.

Backups may contain keys whose export flag attribute marks them as 'non-exportable'. The ST author specifies the cryptographic operations used to protect confidentiality and integrity of any supported backups using one or more iterations of FCS_COP.1.

Application Note 11 (ST)

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

- *FCS_COP.1//AES_Encryption_CBC*
- *FCS_COP.1//AES_Decryption_CBC*
- *FCS_COP.1//AES_CMAC*
- *FCS_COP.1//KeyDerivation*

FDP_ACF.1.3/Backup The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*[132].

FDP_ACF.1.4/Backup The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[133]

Application Note 26 (PP)

---

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

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

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

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

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

If the TOE does not provide backup and restore operations, then the Security Target shall include FDP_ACC.1/Backup and FDP_ACF.1/Backup but shall state in an Application Note for each of these SFRs that the relevant security requirements are trivially met because no backup facility is provided.

FDP_SDI.2 Stored data integrity monitoring and action

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

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for integrity errors[134] on all **keys (including security attributes)[135]**, based on the following attributes: *integrity protection data*[136].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall
(1) prohibit the use of the altered data
(2) notify the error to the user[137].

Application Note 27 (PP)

No specific requirement is placed here on the nature of the integrity protection data, but the Security Target shall describe this protection measure, and shall identify the iteration of FCS_COP.1 that covers any cryptographic algorithm used.

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

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

FDP_RIP.1 Subset residual information protection

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

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *de-allocation of the resource from*[138] the following objects:
(1) authorisation data
(2) secret keys[139].

---

[134] [PP_CMTS] [assignment: integrity errors

[135] [PP_CMTS] objects

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

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

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

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

Application Note 28 (PP)
Authorisation data is not to be stored persistently in the TOE; the refinements to ADV_ARC.1 in section 7.4.1 require the approach to minimising the time that this data is held before de-allocation according to FDP_RIP.1.

7.3.4 Trusted Path/Channels (FTP)

FTP_TRP.1/Local Trusted Path

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

FTP_TRP.1.1/Local The TSF shall provide a communication path between itself and *local*[140] ***client applications***[141] that are logically distinct from other communication paths and provides assured ***authentication***[142] of its end points and protection of the communicated data from *modification and disclosure*[143].

FTP_TRP.1.2/Local The TSF shall permit *local client applications*[144] to initiate communication via the trusted path.

FTP_TRP.1.3/Local The TSF shall require the use of the trusted path for *protecting the confidentiality and integrity of sensitive data exchanged between the local client application and the TOE over a channel that passes through an insecure environment* [145].

Application Note 29 (PP)
FTP_TRP.1/Local must be completed in a Security Target to identify the local client applications and to reflect the way that the TOE communicates with them, and to justify the security of this communication path. Where the TOE and local client applications are located within the physical boundary of the same hardware appliance (e.g. local applications running on a server and communicating with a PCI card on the server's internal PCI bus) then the trusted path may be mapped in the Security Target to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment).

If the TOE does not provide an interface for local client applications, then this SFR is not applicable and is trivially satisfied. This should be stated in the Security Target.

The TOE may provide other additional channels that provide only authentication and integrity protection (not confidentiality), in which case other iterations of FTP_TRP.1 may be added in the ST, allowing the selection of only modification protection in FTP_TRP.1.1 for these additional iterations.

---

[140] [PP_CMTS] [selection: *remote, local*]

[141] [PP_CMTS] users

[142] [PP_CMTS] identification

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

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

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

The Security Target shall identify in an application note the iterations of FCS_COP.1 that provide any cryptographic functions that contribute to the implementation of the trusted path, and the SFRs that provide the authentication of the end points.

Application Note 12 (ST)
Although non-internal local client applications and remote external client applications may run in different environments they have to use the identically same trusted communication mechanism to communicate with the TOE. The following iterations of FCS_COP.1 are used to create the trusted path and to provide the authentication of its end points:

- *FCS_COP.1//Diffie-Hellman*
- *FCS_COP.1//KeyDerivation*
- *FCS_COP.1//RSA_Sign*
- *FCS_COP.1//RSA_Verify*
- *FCS_COP.1//AES_Encryption_CBC*
- *FCS_COP.1//AES_Decryption_CBC*
- *FCS_COP.1//AES_CMAC*
- *FCS_COP.1//HMAC*
- *FCS_COP.1//Hash*

An internal SAM, being an internal local client application, is authenticated by the TOE when loaded within its physical boundary by verifying its module signature applied with the CryptoServer CP5 Module Signature Key. In line with Application Notes 6 and 29 from the PP, the physical protection provided by the TOE is considered a sufficiently trusted path and no further cryptographic protection is required for the communication between the TOE and the internal SAM (see also Application Note 1 (ST)).

FTP_TRP.1/External Trusted Path

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

FTP_TRP.1.1/External The TSF shall provide a communication path between itself and *remote*[146] **external *client applications***[147] that are logically distinct from other communication paths and provides assured ***authentication***[148] of its end points and protection of the communicated data from *modification and disclosure*[149].

FTP_TRP.1.2/External The TSF shall permit *remote external client applications*[150] to initiate communication via the trusted path.

FTP_TRP.1.3/External The TSF shall require the use of the trusted path for *protecting the confidentiality and integrity of sensitive data exchanged between the*

[146] [PP_CMTS] [selection: *remote, local*]

[147] [PP_CMTS] users

[148] [PP_CMTS] identification

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

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

external client application and the TOE over a channel that passes through an insecure environment[151].

Application Note 30 (PP)

FTP_TRP.1/External must be completed in a Security Target to identify the external client applications and to reflect the way that the TOE communicates with them, and to justify the security of this communication path. The word "remote" in FTP_TRP.1.1/External and FTP_TRP.1.2/External refers to client applications that are described as "external" in the rest of this PP.

If the TOE does not provide an interface for external client applications, then this SFR is not applicable and is trivially satisfied. This should be stated in the Security Target.

The TOE may provide other additional channels that provide only authentication and integrity protection (not confidentiality), in which case other iterations of FTP_TRP.1 may be added in the ST, allowing the selection of only modification protection in FTP_TRP.1.1 for these additional iterations.

The Security Target shall identify in an application note the iterations of FCS_COP.1 that provide any cryptographic functions that contribute to the implementation of the trusted path, and the SFRs that provide the authentication of the end points.

Application Note 13 (ST)

Although non-internal local client applications and remote external client applications may run in different environments they have to use the identically same trusted communication mechanism to communicate with the TOE. The following iterations of FCS_COP.1 are used to create the trusted path and to provide the authentication of its end points:

- *FCS_COP.1//Diffie-Hellman*
- *FCS_COP.1//KeyDerivation*
- *FCS_COP.1//RSA_Sign*
- *FCS_COP.1//RSA_Verify*
- *FCS_COP.1//AES_Encryption_CBC*
- *FCS_COP.1//AES_Decryption_CBC*
- *FCS_COP.1//AES_CMAC*
- *FCS_COP.1//HMAC*
- *FCS_COP.1//Hash*

7.3.5 Protection of the TSF (FPT)

FPT_STM.1 Reliable time stamps

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

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

Application Note 31 (PP)

The TOE must provide timestamps suitable for supporting the time in an audit record for FAU_GEN.1. If the TOE provides additional time stamping services for client applications, or other record of the time of an operation for client applications, then these should be covered

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

in one or more separate iterations of the SFR, with an Application Note added to define any specific requirement for reliability of the time information for that service.

FPT_TST_EXT.1 Basic TSF Self Testing

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

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests during initial start-up (or power-on *or reset*) and *at the conditions firmware download, PTRNG request, DRBG request and key pair generation*[152] to demonstrate the correct operation of the TSF:

- *At initial start-up (or power-on or reset):*
  - *Software/firmware integrity test*
  - *Cryptographic algorithm tests*
  - *Critical Functions Tests including memory tests and Random number generator tests*
- *At firmware download:*
  - *Firmware download test (via RSA signature verification)*
- *At each PTRNG request:*
  - *PTRNG online test according to [AIS 20/31] for RNG class PTG.2*
- *At each DRBG request:*
  - *Conditional DRBG test according to [FIPS 140-2] §4.9.2*
- *At key pair generation:*
  - *Pair-wise consistency test according to [FIPS 140-2] §4.9.2*[153].

Application Note 32 (PP)

Completion of the selection in FPT_TST_EXT.1.1 may be by 'None' (in which case the 'and' preceding the selection should be deleted and no selection text included). Completion of the list of additional tests in the final assignment may include tests performed at initial start-up (or power-on) and/or tests run under the conditions specified in the earlier selection and assignment. The term 'start-up (or power-on) means that the tests should be executed at least any time that the TOE is powered-on.

The tests of the cryptographic functions shall include all cryptographic functions covered by FCS_COP.1. The Operational Guidance shall include a description of the errors that may arise from self-test and the actions that should be taken in response to each.

FPT_PHP.1 Passive detection of physical attack

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

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

---

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

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

FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

Application Note 33 (PP)

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

Because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 for this TOE is equivalent to the physical security mechanisms for tamper detection and response required by section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3. (Cf. refinement of AVA_VAN.5 in section 7.4.1.)

FPT_PHP.3 Resistance to physical attack

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

FPT_PHP.3.1 The TSF shall resist *physical manipulation*[154] to the *entire TOE components implementing the TSF*[155] by responding automatically such that the SFRs are always enforced.

Application Note 34 (PP)

This SFR is linked to the requirements for passive detection of physical attacks in FPT_PHP.1, and should identify the relevant responses of the TOE involved in meeting the key zeroisation requirements of ISO/IEC 19790:2012 Security Level 3. As in the case of FPT_PHP.1, because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.3 for this TOE is equivalent to the level of assessment for this aspect of tamper detection and response required for section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements by each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3. (Cf. refinement of AVA_VAN.5 in section 7.4.1.)

FPT_FLS.1 Failure with preservation of secure state

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

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

1. *Self-test according to FPT_TST_EXT.1 fails*
2. *Environmental conditions are outside normal operating range (including temperature and power)*

[154] [assignment: physical tampering scenarios]

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

3. *Failures of critical TOE hardware components (including the RNG) occur*
4. *Corruption of TOE software occurs*
5. *Failures caused by sensitive TOE software components*[156].

Application Note 35 (PP)

The Operational Guidance shall include a description of the specific failures that are detected (e.g. the thresholds for environmental conditions, and the nature of the monitoring of specific critical TOE hardware components), how these failures are notified, and the actions that should be taken in response to each.

7.3.6 Security Management (FMT)

For the purposes of specifying a minimum set of security attributes of keys, and the constraints on initialisation and modification of these attributes in FMT_MSA.1 and FMT_MSA.3, two separate types of keys are defined: Assigned Keys (defined and recognised by having their 'Assigned Flag' attribute set to 'assigned'), and general keys (keys that have their 'Assigned Flag' attribute set to 'non-assigned').
According to the Protection Profile [PP_CMTS], Assigned Keys represent a type of key that can be more easily mapped to requirements for sole control as required by [Regulation], because changes to some of their attributes are more tightly controlled (see FMT_MSA.1/AKeys, and the description of attributes below) and, since they are intended for use within the TOE, because they cannot be imported or exported[157]. In particular, an Administrator cannot avoid the need to provide the current authorisation data in order to use such a key, nor can an Administrator change the authorisation data (which would then allow use of the key by the Administrator). This enables a key to be generated and then to be made an Assigned Key at the point where it is assigned to an individual signatory or, in the case of a key used for the creation of electronic seals, to a group of key users[158].
In the FMT_MSA SFRs specified for keys by the PP [PP_CMTS], the permitted values of assignments have been restricted to identify a minimum set of attributes that must be mapped to their implementation in the TOE, and to specify a minimum set of constraints on their initialisation and subsequent modification. Additional notes regarding these attributes are as follows:

- key identifier: this must be sufficient to uniquely identify the key within the system of which the TOE is a part
- key type: this identifies at a minimum whether the key is a secret key of a symmetric cryptographic algorithm or the secret (commonly called private) key of an asymmetric cryptographic algorithm
- authorisation data: value of data that allows the key to be used for cryptographic operations according to the rules in other SFRs such as FDP_IFF.1/KeyBasics,

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

[157] Assigned Keys may be stored externally in a form that protects the confidentiality and integrity of the key and the binding of the key to its attributes (in particular the requirements of the SFRs FDP_IFF.1/KeyBasics and FDP_SDI.1 apply to externally stored keys), as discussed in [PP_CMTS] section 1.3.1.

[158] Secure operating procedures will be needed in order to ensure that the process from generation to assignment is suitable for maintaining any requirements for non-repudiation that may apply to the application context for use of the key (cf. OE.DataContext and the refinement to AGD_OPE.1 in section 7.4.1).

FDP_ACF.1/KeyUsage, and FDP_ACF.1/Backup. Authorisation data is required only for secret keys

- re-authorisation conditions: the constraints on uses of the key that can be made before reauthorisation is required according to FIA_UAU.6/KeyAuth, and which determines whether a subject is currently authorised to use a key as in FDP_ACF.1/KeyUsage. The types of secret keys to which re-authorisation conditions apply, and the details of the re-authorisation conditions for the specific TOE are described in FIA_UAU.6/KeyAuth in section 7.3.2
- key usage: the cryptographic functions that are allowed to use the key as detailed in FDP_ACF.1/KeyUsage
- export flag: indicates whether the key is allowed to be exported (cf. FDP_IFF.1/KeyBasics); allowed values are referred to in the PP as 'true' (meaning that export is allowed) and 'false' (meaning that export is not allowed) but may be mapped to other suitable binary values in the TOE implementation
- assigned flag: indicates whether the key has currently been assigned. Once a key has been assigned by an Administrator then its authorisation data can only be changed on successful validation of the current authorisation data – it cannot be changed or reset by an Administrator – and the key usage attribute cannot be changed; allowed values are referred to in the PP as 'assigned' and 'non-assigned' but may be mapped to other suitable binary values in the TOE implementation.

FMT_SMR.1 Security roles

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1	The TSF shall maintain the roles *Administrator, Non-internal Local Client Application, Internal SAM, External Client Application*[159]*, Key User, User Administrator, Key Manager, Security Officer*[160] [161].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

Application Note 36 (PP)

The Local Client Application role represents an identifiable subject that communicates locally with the TOE, i.e. within the same hardware appliance. The External Client Application role represents an identifiable subject that communicates remotely with the TOE over a secure channel. A TOE can support one or both types of Client Applications.

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

Application Note 14 (ST)

The TOE implements the following roles for the different users:

- *Administrator Roles*

[159] *[selection: Local Client Application, External Client Application]*

[160] [*assignment: list of additional authorised identified roles*]

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

  - *User Administrator (user management tasks like creation of users, deletion of users)*
  - *Administrator (general administration of the CryptoServer like system time setting, load, update and deletion of firmware)*
  - *Key Manager (key management tasks necessary for the usage of the CryptoServer like unblocking of blocked keys, key generation, key export and import, key backup and key restore, key deletion)*
  - *SO (Security Officer) (creating, modifying or deleting key group specific configuration objects and initiating key groups where he belongs to)*
- *Key User (uses the CryptoServer for cryptographic operations like signature creation)*
- *External Client Application (uses the CryptoServer for creating a secure channel; thus each authenticated user can in addition assume the role External Client Application)*
- *Local Client Application; two types of local client applications exist:*
  - *An Internal SAM is an internal Local Client Application that runs within the physical boundary of the TOE and uses the internal TOE interface. It can be assumed to be sufficiently identified and authenticated (see Application Note 1 (ST)).*
  - *A Non-internal Local Client Application connects to the TOE via the PCIe interface, it uses the CryptoServer for creating a secure channel; thus each authenticated user can in addition assume the role Local Client Application.*

FMT_SMF.1 Security management functions

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

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

1. *Unblock of access due to authentication or authorisation failures*
2. *Modifying attributes of keys*
3. *Export and deletion of the audit data, which can take place only under the control of the Administrator role*
4. *backup and restore functions*[162]
5. *key import function*[163]
6. *key export function*[164]
7. *software update function (FMT_MTD.1//SWUpdate)*[165].

Application Note 37 (PP)

The unblocking of authentication or authorisation failures in FMT_SMF.1.1 (1) is related to the authentication and authorisation failures described in FIA_AFL.1 (and its iterations). The attributes of keys in FMT_SMF.1.1 (2) correspond to the attributes in FMT_MSA.1/GenKeys

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

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

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

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

and FMT_MSA.1/AKeys. Export of audit data in FMT_SMF.1.1 (3) relates to the ability to export audit data from the TOE for preservation and storage elsewhere. The selections in FMT_SMF.1.1 (4), (5) and (6) identify whether or not the TOE provides the relevant functions (and must therefore correspond to the relevant statements in the ST for FDP_IFF.1.2/KeyBasics, FDP_ACC.1/Backup and FDP_ACF.1/Backup.

FMT_MTD.1/Unblock//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/Unblock//User The TSF shall restrict the ability to *unblock*[166] the *any user account blocked due to consecutive authentication failures*[167] to *User Administrators*[168].

FMT_MTD.1/Unblock//Key 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/Unblock//Key The TSF shall restrict the ability to *unblock*[169] the *any key blocked due to consecutive authorisation failures*[170] to *Key Managers*[171].

Application Note 38 (PP)

The list of TSF data assigned must correspond to the relevant data blocked by authentication or authorisation failures according to the associated iteration(s) of FIA_AFL.1. For the purposes of unblocking, the TSF data in the assignment includes any key that can be affected by blocking due to failure of authorisation (as in FIA_UAU.6), as well as user accounts (as in FIA_UAU.1) blocked by authentication/authorisation failures.

There is a distinction between administrators authorised to unblock a key and users authorised to use the key. When unblocking a secret key, the unblocking process must not allow a subject to use the key other than a subject who is authorised by presentation of the current authorisation data. For example, an administrator who is able to unblock the key cannot then use the key as a result of the unblocking (so the unblocking process does not itself allow the key to be used, nor does it enable the authorisation data to be changed without proving knowledge of the previous authorisation data). This is a part of ensuring that sole control of secret keys can be achieved.

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

[167] [assignment: list of TSF data]

[168] [assignment: the authorised identified administrative roles]

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

[170] [assignment: list of TSF data]

[171] [assignment: the authorised identified administrative roles]

FMT_MTD.1/AuditLog Management of TSF data

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

FMT_MTD.1.1/AuditLog The TSF shall restrict the ability to *control export and deletion of*[172] the *audit log records*[173] to the *Administrator, User Administrator, Key Manager and Security Officer role (for the ability to control export) and to the Administrator and User Administrator role (for the ability to control deletion of the audit log records)*[174].

Application Note 39 (PP)

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

FMT_MTD.1//SWUpdate Management of TSF data

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

FMT_MTD.1.1//SWUpdate The TSF shall restrict the ability to *update*[172] the *TSF executable code stored in the TOE in form of software or firmware*[173] to *the Administrator role*[174].

FMT_MSA.1/GenKeys//AFlag 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/GenKeys//AFlag The TSF shall enforce the *Key Usage SFP*[175] to restrict the ability to *modify*[176] the security attributes *Assigned Flag of any General*

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

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

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

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

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

(non-Assigned) Key[177] to *Key Managers, and only to change from non-Assigned to Assigned*[178].

FMT_MSA.1/GenKeys//ExportF 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/GenKeys//ExportF The TSF shall enforce the *Key Usage SFP*[179] to restrict the ability to *modify*[180] the security attributes *Export Flag of any General (non-Assigned) Key*[181] to *Key Managers, and only to change from 'true' (meaning that export is allowed) to 'false' (meaning that export is not allowed)*[182].

FMT_MSA.1/GenKeys//AuthD 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/GenKeys//AuthD The TSF shall enforce the *Key Usage SFP*[183] to restrict the ability to *modify*[184] the security attributes *Authorisation Data of any General (non-Assigned) Key*[185] to *any Key User, but only when modification operation of Authorisation Data includes presentation of current Authorisation Data, or to Key Managers*[186].

FMT_MSA.1/GenKeys//None Management of security attributes

Hierarchical to: No other components.
Dependencies: [FDP_ACC.1 Subset access control, or

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

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

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

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

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

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

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

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

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

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

FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/GenKeys//None The TSF shall enforce the *Key Usage SFP*[187] to restrict the ability to *modify*[188] the security attributes *Key ID, Key Type, Re-Authorisation conditions, Key Usage, Integrity Protection Data of any General (non-Assigned) Key*[189] to *none (moreover the Re-Authorisation conditions are implicit and constant for all keys, and Integrity Protection Data are maintained automatically by TSF)*[190].

FMT_MSA.1/AKeys//AuthD 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/AKeys//AuthD The TSF shall enforce the *Key Usage SFP*[191] to restrict the ability to modify[192] the security attributes *Authorisation Data of any Assigned Key*[193] to *any Key User or Key Manager but only when modification operation of Authorisation Data includes presentation of current Authorisation Data*[194].

FMT_MSA.1/AKeys//None 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/AKeys//None The TSF shall enforce the *Key Usage SFP*[195] to restrict the ability to modify[196] the security attributes *Key ID, Key Type, Re-Authorisation conditions, Key Usage, Export Flag, Assigned Flag, Integrity Protection Data*

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

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

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

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

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

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

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

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

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

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

of any Assigned Key[197] to *none (moreover the Re-Authorisation conditions are implicit and constant for all keys, and Integrity Protection Data are maintained automatically by TSF)*[198].

Application Note 40 (PP)

The Key Attributes Modification Table is referenced from FMT_MSA.1/GenKeys, and FMT_MSA.1/AKeys (and all its iterations). The required constraints on security attribute modification specified in ~~this PP~~ [PP_CMTS] are shown in ~~Table 1~~Table 3: Key Attribute Modification Table; the Security Target completes the other parts not specified here (along with any other information for other security attributes relevant to a particular TOE). The specific attributes used by a particular TOE may vary, but the Security Target must make clear how control is achieved over the ability to modify attributes of keys in terms of the specific attributes and controls imposed by the TOE. Where applicable to the operational environment for a particular TOE, these controls should be described with reference to the ways that they are used to provide qualified electronic signatures and qualified electronic seals that meet the requirements of [Regulation] (cf. the refinement to AGD_OPE.1 in section 7.4.1).

Where a TOE does not support one of the individual types of key then the Security Target states this, and the requirements for that type of key are considered to be trivially satisfied. Authorisation Data and Re-authorisation conditions are required for secret keys only. Re-authorisation conditions include the conditions specified for FIA_UAU.6.1/KeyAuth (matching the assignments and selections made for that SFR in the Security Target).

Key Attribute (MSA.1)	Assigned Key	General Key
Key ID	Cannot be modified	Cannot be modified
Key Type	Cannot be modified	Cannot be modified
Authorisation data	Modified only when modification operation includes successful validation of current (pre-modification) authorisation data	Modified only when modification operation includes successful validation of current (pre-modification) authorisation data, or by an Administrator
Re-authorisation conditions	Cannot be modified	-
Key Usage	Cannot be modified	-
Export Flag	Cannot be modified	-
Assigned Flag	Cannot be modified	Can be modified only by Administrator, and only to change from non-Assigned to Assigned

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

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

Key Attribute (MSA.1)	Assigned Key	General Key
Integrity Protection Data	Cannot be modified by users (maintained automatically by TSF)	Cannot be modified by users (maintained automatically by TSF)

Table 3: Key Attribute Modification Table[199]

FMT_MSA.3/Keys Static attribute initialisation

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

FMT_MSA.3.1/Keys The TSF shall enforce the *Key Usage SFP*[200] to provide *permissive*[201] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Keys The TSF shall allow the *Key Manager or Internal SAM*[202] to specify alternative initial values to override the default values when an object or information is created.

Key Attribute (MSA.1)	Assigned Key	General Key
Key ID	Initialised by generation process	Initialised by generation process
Key Type	Initialised by generation process	Initialised by generation process
Authorisation data	Initialised by creator during generation	Initialised by creator during generation
Re-authorisation conditions	Initialised by Administrator during generation	-
Key Usage	Initialised by creator during generation	-
Export Flag	False (i.e. no export allowed)	-
Assigned Flag	Initialised by generation process	Non-Assigned
Integrity Protection Data	Initialised automatically by TSF	Initialised automatically by TSF

Table 4: Key Attribute Initialisation Table[199]

[199] It is acceptable for a Security Target to specify more restrictive modification conditions than listed in this table, but not to specify less restrictive modification conditions. Where no specific condition is specified (denoted by '---') then the Security Target is not constrained by this PP, but clearly the requirements of the system of which the cryptographic module is a part may have more detailed requirements for a specific deployment (i.e. operational environment).

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

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

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

Application Note 41 (PP)

The Key Attributes Initialisation Table is referenced from FMT_MSA.3/Keys and matches the attributes covered by the separate iterations of FMT_MSA.1 above. The required constraints on security attribute initialisation specified in [PP_CMTS] are shown in ~~Table 2~~Table 4: Key Attribute Initialisation Table; the Security Target completes the other parts not specified here (along with any other information for other security attributes relevant to a particular TOE). The specific attributes used by a particular TOE may vary, but the Security Target must make clear how control is achieved over the ability to modify attributes of keys in terms of the specific attributes and controls imposed by the TOE. Where applicable to the operational environment for a particular TOE, these controls should be described with reference to the ways that they are used to provide qualified electronic signatures and qualified electronic seals that meet the requirements of [Regulation] (cf. the refinement to AGD_OPE.1 in section 7.4.1).

Where a TOE does not support one of the individual types of key then the Security Target states this, and the requirements for that type of key are considered to be trivially satisfied.

Authorisation Data and Re-authorisation conditions are required for secret keys only, and only as described in the assignments and selections made in the Security Target for FIA_UAU.6/KeyAuth.

Attributes assigned by the TOE to any imported keys must be described in the Security Target and in operational user guidance (see the refinements to AGD_OPE.1 in section 7.4.1), noting that a secret key can only be imported if it is a non-Assigned key (cf. FDP_IFF.1/KeyBasics).

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

7.3.7 Security Audit Data Generation (FAU)

FAU_GEN.1 Audit data generation

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

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the *not specified*[203] level of audit; and[204]
c) Startup of the TOE;
d) Shutdown of the TOE
e) Cryptographic key generation (FCS_CKM.1 (all iterations));
f) Cryptographic key destruction (FCS_CKM.4);
g) Failure of the random number generator (FCS_RND.1);
h) Authentication and authorisation failure handling (FIA_AFL.1 (all iterations)): all unsuccessful authentication or authorisation attempts, the reaching of the threshold for the unsuccessful authentication or authorisation attempts and the blocking actions taken;

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

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

i) *All attempts to import or export keys (FDP_IFF.1/KeyBasics);*

j) *All modifications to attributes of keys (FDP_ACF.1/KeyUsage, FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys (all iterations));*

k) *Backup and restore (FDP_ACF.1/Backup): use of any backup function, use of any restore function, unsuccessful restore because of detection of modification of the backup data;*

l) *Integrity errors detected for keys (FDP_SDI.2);*

m) *Failures to establish secure channels (FTP_TRP.1/Local, FTP_TRP.1/External);*

n) *Self-test completion (FPT_TST_EXT.1);*

o) *Failures detected by the TOE (FPT_FLS.1);*

p) *All administrative actions (FMT_SMF.1, FMT_MSA.1 (all iterations), FMT_MSA.3/Keys,);*

q) *Unblocking of access (FMT_MTD.1/Unblock//User and FMT_MTD.1/Unblock//Key);*

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

s) *none*[205].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST:

- *none*[206].

Application Note 42 (PP)

The Security Target is not required to identify separate levels of audit in FAU_GEN.1.1. However, the Operational Guidance is required to describe any configuration or other actions that apply to audit functions, and to make clear, in cases where logging of particular audit events is optional, how to ensure that any individual audit event is logged. Default logging actions of the TOE must also be described in Operational Guidance.

The Administrative Actions logged need not be limited to those related to FMT SFRs: other administrative actions affecting the operation of SFRs should also be included (and listed as part of the assignment in FAU_GEN.1.1).

FAU_GEN.2 User identity association

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

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

[206] [assignment: other audit relevant information]

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

FAU_STG.2 Guarantees of audit data availability

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

FAU_STG.2.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
FAU_STG.2.2 The TSF shall be able to *prevent*[207] unauthorised modifications to the stored audit records in the audit trail.
FAU_STG.2.3 The TSF shall ensure that *all*[208] stored audit records will be maintained when the following conditions occur: *audit storage exhaustion*[209].

Application Note 43 (PP)

The Operational Guidance is required to describe any use that the TOE makes of an external audit server, the situation regarding records held locally on the TOE and those held externally on an audit server (e.g. the TOE might accumulate records locally before transferring them to an external audit server), and the way in which audit records are maintained when local audit storage is exhausted (including description of the actions taken by the TOE when audit storage exhaustion is detected). The Operational Guidance shall describe the protection applicable to all records created by the TOE (in order to provide prevention or detection of unauthorised modifications as in FAU_STG.2.2), and shall identify any obligations for the environment in maintaining audit trail protection. The expectation is that this will comprise cryptographic methods of prevention or detection of unauthorised modification (including deletion) of audit records.

Control over export and deletion of the audit log records is limited to the Administrator role as specified in FMT_MTD.1/AuditLog.

7.4 Security Assurance Requirements

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

Assurance Class	Assurance Components
Security Target (ASE)	ST introduction (ASE_INT.1)
	Conformance claims (ASE_CCL.1)
	Security problem definition (ASE_SPD.1)
	Security objectives (ASE_OBJ.2)

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

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

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

Assurance Class	Assurance Components
	Extended components definition (ASE_ECD.1)
	Derived security requirements (ASE_REQ.2)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Security architecture description (ADV_ARC.1)
	Complete functional specification (ADV_FSP.4)
	Basic modular design (ADV_TDS.3)
	Implementation representation of the TSF (ADV_IMP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Production support, acceptance procedures and automation (ALC_CMC.4)
	Problem tracking CM coverage (ALC_CMS.4)
	Delivery procedures (ALC_DEL.1)
	Identification of security measures (ALC_DVS.1)
	Developer defined life-cycle model (ALC_LCD.1)
	Well-defined development tools (ALC_TAT.1)
Tests (ATE)	Functional testing (ATE_FUN.1)
	Analysis of coverage (ATE_COV.2)
	Testing: basic design (ATE_DPT.1)
	Independent testing – sample (ATE_IND.2)
Vulnerability assessment (AVA)	**Advanced methodical vulnerability analysis (AVA_VAN.5)**

Table 5: Security Assurance Requirements

7.4.1 Refinement of Security Assurance Requirements

The following refinements are made to selected assurance requirements in Table 5:

ADV_ARC.1 Security architecture description

Refinement:

The following specific topics must be addressed as part of ADV_ARC.1 for this Protection Profile. It is acceptable for references to deliverables supplied for other assurance families, such as ADV_FSP, to be used to meet these requirements, provided that the relationship of the relevant interface specifications to the concepts in the Protection Profile is clear. Note that in some cases, the requirement for description of these particular aspects under ADV_ARC is intended to make clear any differences between the full capabilities of the product and the scope of the Security Target.

(1) In general cryptographic modules will make use of 'support keys' as part of their implementation of protection mechanisms, where these keys are generally not held on behalf of specific users[210] or client applications, but are used by the TOE to carry out its

[210] Some support keys may be seen as being held on behalf of administrators, but the main intention of distinguishing support keys and user keys is for the ADV_ARC.1 deliverables to describe all the different types of key available, their properties, and their relationship to the SFRs in this Protection Profile.

normal operations and as part of the implementation mechanism other SFRs and to protect the TSF itself. These support keys may be used for a variety of purposes (including aspects such as authentication, authorisation, secure channels, security of external storage, or internal data protection), For the purposes of this PP, support keys used by the TOE are treated as TSF data, and require a specific security rationale to be included as part of the ADV_ARC.1 deliverables. This rationale must include a description of the key architecture, identifying all support keys used by the TOE (at least in its evaluated configuration), their method of generation and storage, their purpose in TOE operation, and the ways in which they are protected so as to support the requirements of FDP_IFF.1/KeyBasics and FDP_ACF.1/KeyUsage (noting that the mechanisms used for support keys may differ from those used for user keys). Examples would be keys used for wrapping user keys in order to allow secure storage of the user keys, keys used to implement secure channels, and keys used to protect backups. The description must demonstrate that sufficient entropy has been used in the generation of each support key, and the source of that entropy. The rationale must demonstrate that these support keys cannot be exported/imported in a way that threatens the secure operation of the TOE. The evaluator shall include the description of the support keys in their analysis of the protection of user data (e.g. to confirm that it does not introduce vulnerabilities in the implementation of the SFRs).

(2) If updates to the TOE software or firmware are supported then the ADV_ARC.1 deliverables must describe how the TOE is protected against unauthorised updates, by using digital signatures. This shall be confirmed by evaluator testing (if updates are supported) to confirm that updates with invalid signatures are rejected without being executed. The digital signature algorithms used to protect updates shall be included in the scope of FCS_COP.1 signature SFR(s).

(3) The ADV_ARC.1 deliverables must in particular describe

a. Any use that the TOE makes of an audit server
b. The locations used for any externally stored keys and the structure and format of the externally stored keys including the cryptographic structures that protect the keys in their externally stored form, and that bind them to their attributes (support keys are separately addressed by the description required in item 1 above)
c. All key import and/or export functions and the secure channels that they use
d. The secure channels supported by the TOE and the authentication mechanisms that they use (cf. FTP_TRP.1/Local & FTP_TRP.1/External)
e. All local and external interfaces used for communications with users, client applications, audit data, and stored TOE data (cf. *[PP_CMTS]* Figure 1)
f. The specific key attributes supported, their method of representation (e.g. the relevant data structures and permitted values) and the method by which they are bound to the corresponding key value (cf. FMT_MSA.1 *(all iterations)*) This also includes identifying the types of keys (if any) that support re-authorisation conditions described in FIA_UAU.6/KeyAuth
g. The user types and roles supported, the interfaces by which they interact with the TOE (e.g. a local administrator console or an externally available API), the authentication methods used (cf. FIA_UAU.1 and Application Note 17 *(PP)*), and any privileges available to the user type/role
h. All of the cryptographic functions provided (cf. *[PP_CMTS]* section 1.3.1.1) and whether any non-endorsed cryptographic algorithms and/or cryptographic functions are available (cf. FCS_COP.1 *(all iterations)* and *[PP_CMTS]* section 1.3.1.3)

i. The authorisation methods used for keys (cf. FIA_UAU.6/KeyAuth & FDP_ACC.1/KeyUsage)
j. Description of the way in which the TOE ensures that it only holds authorisation data for the minimum time possible before de-allocating it according to FDP_RIP.1
k. If the TOE provides backup operations then the ADV_ARC deliverables shall describe the use of support keys by the backup and restore processes (cf. FDP_ACF.1/Backup), and in particular shall describe the ways in which confidentiality and integrity of the backup are provided, and the way in which the TOE rejects an attempt to carry out a restore process using backup data that has been modified
l. Any mechanisms that the TOE uses to support dual person control (cf. FDP_ACF.1/Backup).

AGD_OPE.1 Operational user guidance

Refinement:

The following specific topics must be addressed as part of the Operational Guidance for the TOE:

1. The specific ways in which the TOE needs to be configured and used in order to provide qualified electronic signatures and qualified electronic seals that meet the requirements of [Regulation]. This includes ways in which the TOE can ensure that the signatory can, with high level of confidence, have sole control over the use of the secret key that acts as his/her signature creation data. Thus, for example, it may be necessary for client applications to use TOE interfaces according to certain guidance in order to correctly implement the requirements on attributes of keys as described in this PP. It may be necessary for the TOE to define ways in which secret keys to be used for signing purposes can be created in a way that does not allow subsequent modification of some or all of their attributes, e.g. by an administrator, before they are assigned to the signatory (cf. FMT_MSA.1/AKeys (all iterations)). The intention of this aspect of the operational user guidance documentation is to identify the configuration and secure use required for a particular TOE, and how it is necessary to connect this with other aspects such as procedural controls and client applications in the operational environment.

   The evaluators shall test the identified ways of using the TOE for qualified electronic signatures and qualified electronic seals to demonstrate that the description in the Operational Guidance is suitably complete, and that the keys produced by following the Operational Guidance do indeed meet the requirements of ~~requirements of~~ [Regulation], Annex II & Annex III~~]~~, for qualified electronic signatures and qualified electronic seals.
2. The use of trusted channels (cf. FTP_TRP.1/Local & FTP_TRP.1/External).
3. The available key attributes, their possible values, and the meaning of each of these values (cf. FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys *(all iterations))*, including their use to constrain the period and number of uses that are enabled by authorisation of a key (cf. FIA_UAU.6/KeyAuth and Application Note 19 *(PP)*).
4. Identification of any non-endorsed cryptographic algorithms and/or cryptographic functions that are available (cf. FCS_COP.1 *(all iterations)* and *[PP_CMTS]* section 1.3.1.3).
5. Identification of any other cryptographic algorithms and operations that are not included in the scope of the Security Target.

6. Possible errors from the self-test process and the actions that should be taken in response to each (cf. FPT_TST_EXT.1 & Application Note 32 *(PP)*).
7. Specific failures detected by the TOE (cf. FPT_FLS.1 & Application Note 35 *(PP)*).
8. Audit functions and their configuration (including specification of the available audit records), along with any other actions that are associated with audit functions (e.g. archiving or viewing audit records, or use of an external audit server) (cf. FAU_GEN.1 & Application Note 42 *(PP)*, FAU_STG.2 & Application Note 41 *(PP)*, FMT_MTD.1/AuditLog & Application Note 39 *(PP)*).
9. Any configuration and operation requirements for dual-control operations (cf. FDP_ACF.1/Backup).
10. If backup is provided by the TOE (cf. FDP_ACF.1/Backup), then the Operational Guidance shall describe the backup and restore functions, and the administrator roles that are required to carry them out.
11. If key import is provided by the TOE, then the Operational Guidance shall describe how attributes are defined for any imported keys (cf. FMT_MSA.3/Keys). The evaluators shall test the import process to demonstrate that the description in the Operational Guidance is suitably complete, and that the keys imported have attributes appropriately defined. Similarly, if key export is provided by the TOE then the Operational Guidance shall describe whether attributes are exported with keys (and if so, then how the attributes are represented and associated with the exported key), and the evaluators shall test the export process to demonstrate that the description in the Operational Guidance is suitably complete, and that the handling of attributes is as described.
12. The Operational Guidance must contain explicit guidance for the developer of an internal SAM how to invoke the internal TOE interface without compromising the TOE security functionality. It must be validated in the course of the eIDAS evaluation of the SAM that the internal SAM follows all these rules.

ATE_IND.2 Independent testing – sample

Refinement:

The following specific topics must be addressed as part of the independent testing of the TOE:

1. The evaluator shall execute the electronic signature and electronic seal operations provided by the TOE and shall confirm that the signatures and seals returned by the TOE correspond to the correct DTBS.
2. If software and/or firmware updates are supported by the TOE, then the evaluator shall carry out tests to ensure that only updates with valid digital signatures can be installed on the TOE.

AVA_VAN.5 Advanced methodical vulnerability analysis

Refinement:

Regarding the protection of the TOE against physical attacks: because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 and FPT_PHP.3 for this TOE is equivalent to the level of assessment for this aspect of tamper detection and response in section 7.7.2 Physical security general

requirements and section 7.7.3 Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3.

8 Rationales

8.1 Security Objectives Rationale

8.1.1 Security Objectives Rationale

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

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

Table 6: Security Problem Definition mapping to Security Objectives

8.1.2 Security Objectives Sufficiency

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

8.1.2.1 Threats

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

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

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

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

T.KeyOveruse is concerned with the possibility that more uses may be made of an authorised key than were intended, and this is addressed by the requirements of OT.KeyUseScope which requires that the TOE allows a user to define specific values for the number of uses, or the time period of use, of a key that an authorisation allows.

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

T.DataMod is concerned with the possibility of unauthorised modification of data transmitted between a client application and the TOE, and this is addressed by OT.DataMod which requires that the TOE provides secure channels that can be used to protect the integrity of

data that they carry. As with T.DataDisclose, the appropriate use of such channels is a requirement for the environment as expressed in OE.DataContext, as is the use of appropriate procedures in OE.AppSupport.

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

8.1.2.2 Organisational Security Policies

P.Algorithms requires the use of key generation and other cryptographic functions that are endorsed by appropriate authorities, and this is addressed by OT.Algorithms.
P.KeyControl requires that the TOE can provide controls and support a key lifecycle to ensure that secret keys can be reliably protected against use by those other than the owner of the key, and that the keys can be confined to use for certain cryptographic functions. This is addressed by a combination of TOE objectives as follows:

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

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

8.1.2.3 Assumptions

Each of the Assumptions in section 3.5 is directly matched by a security objective for the operational environment in section 4.2. The wording of each objective for the operational environment includes the wording of each assumption, and no further rationale is therefore given here.

8.2 Security Requirements Rationale

8.2.1 Security Requirements Coverage

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

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FCS_CKM.1//AES		X												
FCS_CKM.1//RSA		X												
FCS_CKM.1//ECDSA		X												
FCS_CKM.4	X													
FCS_COP.1//AES_Encryption_CBC		X												
FCS_COP.1//AES_Encryption_OFB		X												
FCS_COP.1//AES_Decryption_CBC		X												
FCS_COP.1//AES_Decryption_OFB		X												
FCS_COP.1//AES_CMAC		X												
FCS_COP.1//AES_ECB		X												
FCS_COP.1//AES_GCM		X												
FCS_COP.1//RSA_Sign		X												
FCS_COP.1//RSA_Verify		X												
FCS_COP.1//RSA_Encryption		X												
FCS_COP.1//RSA_Decryption		X												
FCS_COP.1//ECDSA_Sign		X												
FCS_COP.1//ECDSA_Verify		X												
FCS_COP.1//HMAC		X												
FCS_COP.1//Hash		X												
FCS_COP.1//Diffie-Hellman		X												
FCS_COP.1//KeyDerivation		X												
FCS_RNG.1											X			

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FIA_UID.1				X										
FIA_UAU.1//UserAuth				X										
FIA_UAU.1//KeyAuth				X										
FIA_AFL.1//UserAuth				X										
FIA_AFL.1//KeyAuth				X										
FIA_UAU.6/KeyAuth				X		X								
FDP_IFC.1/KeyBasics	X				X				X					
FDP_IFF.1/KeyBasics	X		X		X				X					
FDP_ACC.1/Key_Usage					X	X								
FDP_ACF.1/KeyUsage					X	X								
FDP_ACC.1/Backup										X				
FDP_ACF.1/Backup										X				
FDP_SDI.2			X											
FDP_RIP.1	X				X									
FTP_TRP.1/LOCAL			X	X			X	X	X					
FTP_TRP.1/External			X	X			X	X	X					
FPT_STM.1														X
FPT_TST_EXT.1													X	
FPT_PHP.1												X		
FPT_PHP.3												X		
FPT_FLS.1													X	
FMT_SMR.1				X										X

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FMT_SMF.1				X										X
FMT_MTD.1/Unblock//User				X										
FMT_MTD.1/Unblock//Key				X										
FMT_MTD.1/AuditLog														X
FMT_MTD.1//SWUpdate				X										
FMT_MSA.1/GenKeys//AFlag					X									
FMT_MSA.1/GenKeys//ExportF					X									
FMT_MSA.1/GenKeys//AuthD					X									
FMT_MSA.1/GenKeys//None					X									
FMT_MSA.1/AKeys//AuthD					X									
FMT_MSA.1/AKeys//None					X									
FMT_MSA.3/Keys					X									
FAU_GEN.1														X
FAU_GEN.2														X
FAU_STG.2														X

Table 7: TOE Security Objectives mapping to SFRs

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

OT.Algorithms is addressed by the need to use endorsed standards for FCS_COP.1 (cf. Application Note 14 (PP)) and the use of an appropriate random number generator in FCS_CKM.1. Note that the refinements to assurance components in section 7.4.1 also specify requirements that ensure clear documentation of endorsed and non-endorsed algorithms and functions provided by the TOE.

OT.KeyIntegrity is addressed primarily by FDP_SDI.2 which requires integrity protection of keys and their attributes by the TOE. FDP_IFF.1/KeyBasics requires that any importing or exporting of keys requires the use of secure channels and integrity protection (cf. the requirement for an integrity protected channel as part of FTP_TRP.1/Local and FTP_TRP.1/External, which is linked to the Key Basics SFP by Application Note 20 (PP) under FDP_IFF.1/KeyBasics

OT.Auth is addressed by FIA_UID.1, FIA_UAU.1//UserAuth, and FIA_AFL.1//UserAuth for user authentication (with FMT_MTD.1/Unblock//User, FMT_MTD.1/Unblock//Key, FMT_MTD.1/AuditLog, FMT_MTD.1//SWUpdate and its dependencies on FMT_SMR.1 and FMT_SMF.1 ensuring that appropriate roles and unblocking for authorisation and authentication failures are also provided). Authorisation for external client applications is provided by the requirements for authentication of endpoints in FTP_TRP.1/Local and FTP_TRP.1/External. Authorisation for access to a secret key is additionally addressed by FIA_UAU.1//KeyAuth, FIA_AFL.1//KeyAuth and FIA_UAU.6/KeyAuth.

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

OT.KeyUseScope is addressed by the Key Usage SFP in FDP_ACC.1/KeyUsage and FDP_ACF.1/KeyUsage and by the constraints on time period or number of uses since the last authorisation for use of a secret key required by FIA_UAU.6/KeyAuth.

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

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

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

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

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

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

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

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

8.2.2 SFR Dependencies

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

No.	SFR	Dependency	Dependency satisfied by
	FCS	**Cryptographic Support**	
1.	FCS_CKM.1//AES	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1//AES_* FCS_CKM.4
2.	FCS_CKM.1//RSA	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1//RSA_* FCS_CKM.4
3.	FCS_CKM.1//ECDSA	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1//ECDSA_* FCS_CKM.4
4.	FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1//AES FCS_CKM.1//RSA FCS_CKM.1//ECDSA
5.	FCS_COP.1//AES_Encryption_CBC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//AES FCS_CKM.4
6.	FCS_COP.1//AES_Encryption_OFB	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with	FCS_CKM.1//AES FCS_CKM.4

No.	SFR	Dependency	Dependency satisfied by
		security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	
7.	FCS_COP.1//AES_Decryption_CBC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//AES FCS_CKM.4
8.	FCS_COP.1//AES_Decryption_OFB	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//AES FCS_CKM.4
9.	FCS_COP.1//AES_CMAC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//AES FCS_CKM.4
10.	FCS_COP.1//AES_ECB	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//AES FCS_CKM.4
11.	FCS_COP.1//AES_GCM	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//AES FCS_CKM.4
12.	FCS_COP.1//RSA_Sign	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key	FCS_CKM.1//RSA FCS_CKM.4

No.	SFR	Dependency	Dependency satisfied by
		generation] FCS_CKM.4 Cryptographic key destruction	
13.	FCS_COP.1//RSA_Verify	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//RSA FCS_CKM.4
14.	FCS_COP.1//RSA_Encryption	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//RSA FCS_CKM.4
15.	FCS_COP.1//RSA_Decryption	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//RSA FCS_CKM.4
16.	FCS_COP.1//ECDSA_Sign	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//ECDSA FCS_CKM.4
17.	FCS_COP.1//ECDSA_Verify	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1//ECDSA FCS_CKM.4
18.	FCS_COP.1//HMAC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1 Cryptographic key generation; not relevant because a hash function does not use any cryptographic key. No

No.	SFR	Dependency	Dependency satisfied by
		FCS_CKM.4 Cryptographic key destruction	key generation can be expected here. FCS_CKM.4 Cryptographic key destruction
19.	FCS_COP.1//Hash	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 Cryptographic key generation: not relevant because a hash function does not use any cryptographic key. No key generation can be expected here. FCS_CKM.4 Cryptographic key destruction: not relevant because a hash function does not use any cryptographic key. No key destruction can be expected here.
20.	FCS_COP.1//Diffie-Hellman	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 Cryptographic key generation FCS_CKM.4 Cryptographic key destruction
21.	FCS_COP.1//KeyDerivation	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 Cryptographic key generation FCS_CKM.4 Cryptographic key destruction
22.	FCS_RNG.1	No dependencies.	n.a.
	FIA	**Identification and authentication**	
23.	FIA_UID.1	No dependencies.	n.a.
24.	FIA_UAU.1//UserAuth	FIA_UID.1 Timing of identification	FIA_UID.1
25.	FIA_UAU.1//KeyAuth	FIA_UID.1 Timing of identification	FIA_UID.1

No.	SFR	Dependency	Dependency satisfied by
26.	FIA_AFL.1//UserAuth	FIA_UAU.1 Timing of authentication	FIA_UAU.1//UserAuth
27.	FIA_AFL.1//KeyAuth	FIA_UAU.1 Timing of authentication	FIA_UAU.1//KeyAuth
28.	FIA_UAU.6/KeyAuth	No dependencies	n.a.
	FDP	**User data protection**	
29.	FDP_IFC.1/KeyBasics	FDP_IFF.1 Simple security attributes	FDP_IFF.1/ KeyBasic
30.	FDP_IFF.1/KeyBasics	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	FDP_IFC.1/ KeyBasics FMT_MSA.3/Keys
31.	FDP_ACC.1/Key_Usage	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/KeyUsage
32.	FDP_ACF.1/KeyUsage	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/KeyUsage FMT_MSA.3/Keys
33.	FDP_ACC.1/Backup/	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Backup
34.	FDP_ACF.1/Backup	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/Backup The dependency on FMT_MSA.3 is not relevant in this case since the attribute used in FDP_ACF.1/Backup is determined by the ability of the user to authenticate as an administrator according to FIA_UAU.1//UserAuth.
35.	FDP_SDI.2	No dependencies	n.a.
36.	FDP_RIP.1	No dependencies	n.a.
	TRP	**Trusted path/channels**	
37.	FTP_TRP.1/Local	No dependencies	n.a.
38.	FTP_TRP.1/External	No dependencies	n.a.

No.	SFR	Dependency	Dependency satisfied by
	FPT	**Protection of the TSF**	
39.	FPT_STM.1	No dependencies	n.a.
40.	FPT_TST_EXT.1	No dependencies	n.a.
41.	FPT_PHP.1	No dependencies	n.a.
42.	FPT_PHP.3	No dependencies	n.a.
43.	FPT_FLS.1	No dependencies	n.a.
	FMT	**Security management**	
44.	FMT_SMR.1	FIA_UID.1 Timing of identification.	FIA_UID.1
45.	FMT_SMF.1	No dependencies	n.a.
46.	FMT_MTD.1/Unblock//User	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
47.	FMT_MTD.1/Unblock//Key	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
48.	FMT_MTD.1/AuditLog	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
49.	FMT_MTD.1//SWUpdate	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
50.	FMT_MSA.1/GenKeys (all iterations)	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1/Key_Usage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
51.	FMT_MSA.1/AKeys (all iterations)	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1/Key_Usage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
52.	FMT_MSA.3/Keys	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1/GenKeys (all iterations) FMT_MSA.1/AKeys (all iterations) FMT_SMR.1

No.	SFR	Dependency	Dependency satisfied by
	FAU	**Security audit data generation**	
53.	FAU_GEN.1	FPT_STM.1 Reliable time stamps	FPT_STM.1
54.	FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification	FAU_GEN.1 FIA_UID.1
55.	FAU_STG.2	FAU_GEN.1 Audit data generation	FAU_GEN.1

Table 8: SFR Dependencies Rationale

Key attributes during import or export: the TOE may allow import or export of keys according to the rules in FDP_IFF.1/KeyBasics. For keys that may be imported or exported, the TOE does not place any specific requirements on whether attributes are imported and exported with keys. However, the refinement to AGD_OPE.1 in section *7.4.1* requires that the behaviour of the TOE in this situation is described in documentation, and that the evaluators confirm the behaviour that is documented. Application Note 41 (PP) (for FMT_MSA.1) also requires that the initialisation of any attributes on import is described in the Security Target.

8.2.3 Rationale for SARs

The assurance level for the chosen protection profile is **EAL4 augmented with AVA_VAN.5**. EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialised processes and practices. It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions.
The TOE described in the protection profile is just such a product. Augmentation results from the selection of AVA_VAN.5. All the dependencies of AVA_VAN.5 are satisfied by other assurance components in the EAL4 assurance package.

8.2.4 AVA_VAN.5 Advanced Methodical Vulnerability Analysis

The TOE generates, uses and manages the highly sensitive data in the form of secret keys, at least some of which may be used as signature creation data. The protection of these keys and associated security of their attributes and use in cryptographic operations can only be ensured by the TOE itself. While the TOE environment is intended to protect against physical attacks, a high level of protection against logical attacks (especially those that might be carried out remotely) is also necessary, and is therefore addressed by augmenting vulnerability analysis to deal with High attack potential.

9 TOE Summary Specification

This chapter describes how the TOE will realise the SFRs which are defined in chapter 7.3. For that purpose, the TOE Security Functionality (TSF) will be described by means of a set of security functions (SF.XXX) implemented by the TOE. This detailed description and analysis of the TSF demonstrates how the defined security functions of the TOE work together and support each other. Furthermore, it shows that no inconsistencies exist. Each SFR is implemented by at least one security function. For all SFRs an explanation is given, why and how the defined security functions of the TOE meet the respective SFRs. The given mapping of the SFRs and the security functions of the TOE at the end of this chapter should be considered as an overview and a guidance.

9.1 SF.USER_AUTH: User Authentication

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

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

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

- different administrator roles
    - *User Administrator* (user management tasks like creation of users, deletion of users)
    - *Administrator* (general administration of the CryptoServer like system time setting, load, update and deletion of firmware)
    - *Key Manager* (key management tasks necessary for the usage of the CryptoServer, like unblocking of blocked keys, key generation, key export and import, key backup and key restore, key deletion)
    - *SO (Security Officer)* (creating, modifying or deleting key group specific configuration objects and initiating a key group)
- Key User (who uses the CryptoServer for cryptographic operations like signature creation)
- External Client Application (that uses the CryptoServer for creating a secure channel; hence, each authenticated user can in addition assume the role External Client Application)
- Local Client Application:
    - Non-internal Local Client Application that connects to the CryptoServer via the local (external) PCIe interface and which uses the CryptoServer for creating a

secure channel; hence, each authenticated user can in addition assume the role Non-internal Local Client Application
- Internal SAM: Internal Local Client Application that invokes the internal TOE interface. It is authenticated by signature verification when initially loaded to the CryptoServer and integrity protected by the physical boundary of the TOE and therefore does not need to establish a cryptographically protected secure channel.

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

RSA Signature authentication mechanism: The authentication is performed with an RSA signature (RSA signature scheme RSASSA-PKCS1-v1_5 according to the standard [PKCS#1], chapter 8.2.1, with key lengths of minimum 2048[1] and maximum 8192 bit modulus lengths).

HMAC Password authentication mechanism: For this mechanism a password is used. First the host running the application software demands a 16-byte random value (challenge) from the TOE. Then the host calculates the HMAC value over this challenge and the command data block using the user's authentication password as the HMAC key.

Furthermore, for Internal SAM the following authentication mechanism is provided:

Module Signature authentication mechanism: The authentication is performed with the help of an RSA signature (PKCS#1 signature according to the standard [PKCS#1],) which has to be calculated over the firmware module with the dedicated CryptoServer CP5 Module Signature Key owned by the manufacturer.

After five unsuccessful user authentication attempts the corresponding user is blocked. Any additional attempt of this user to authenticate towards the TOE will fail. Thus SF.USER_AUTH supports FIA_AFL.1//UserAuth (Authentication failure handling). A blocked user can only be unblocked by a User Administrator, hence fulfilling FMT_MTD.1/Unblock//User.

For exchanging sensitive data, a Secure Messaging session (trusted channel) has to be set up between the TOE and the (local non-internal or remote external) client application. Such a Secure Messaging session is mandatory for each command which requires user authentication. Here, although they may run in different environments, for local non-internal client applications and remote external client applications the identically same trusted communication mechanism is enforced by SF.USER_AUTH, fulfilling the SFRs FTP_TRP.1/Local (Trusted Path) as well as FTP_TRP.1/External (Trusted Path).

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

9.2 SF.KEY_AUTH: Key Authorisation

The TOE's concept of mandatory key authorisation ensures that the signatory has sole control over the use of his private keys aimed to create digital signatures at a TSP according to eIDAS. Key authorisation before key usage is required for all secret and private keys.

The key authorisation is not possible without former user authentication. Only if a defined authentication status (e.g. authentication for the Key User role or the Key Manager role) has been obtained, the key authorisation can be realised. Thus, this security function is related to SF.USER_AUTH.

In addition to user authentication, key authorisation to access a secret or private key has to be performed before a key can be used by a cryptographic function or before a key can be exported, implementing in particular FDP_ACF.1.2/KeyUsage (2), FIA_UAU.1//KeyAuth and FDP_ACC.1/KeyUsage. A key can be authorised for a defined number of usage access operations, or for infinite use (until rescinding).

After five unsuccessful key authorisation attempts the corresponding key is blocked. Any additional attempt for key authorisation for this specific key will fail. Thus, SF.KEY_AUTH supports FIA_AFL.1//KeyAuth (Authentication failure handling). A blocked key can only be unblocked by a Key Manager, hence fulfilling FMT.MTD.1/Unblock//Key.

The authorisation for access to a secret or private key stays valid until either the key authorisation is explicitly rescinded with a dedicated command to end the previous key authorisation, or until the defined number of access trials has been reached, implementing FIA_UAU.6/KeyAuth (Re-authenticating). Key authorisation is also lost in case of reset or power-cycle of the TOE.

SF.KEY_AUTH furthermore implements all rules defined in FDP_ACF.1/KeyUsage (Security attribute based Access Control) for the users that shall be able to change key security attributes.

The SAEK signature interface allows an Internal SAM application to request usage of a signature key for which the TOE will not check the key authorisation. As a consequence, the internal SAM calling the SAEK signature interface takes full responsibility on correct legitimation of this operation, including key authorisation as required by [PP_CMTS]: As mandated by TOE Guidance, an Internal SAM will only invoke the SAEK signature interface if it has completely validated key authorisation of the signature key before. Therefore, the TOE can implicitly derive prior successful key authorisation of the signature key from each invocation of the SAEK signature interface.

9.3 SF.ADMIN: Administration

Security-relevant administration of the TOE cannot be done without user authentication: Only if a defined authentication status has been obtained then administration tasks can be executed. In addition to that, for some administration functions, related to key management, the key usage has to be authorised before. The administration security function SF.ADMIN is therefore related to SF.USER_AUTH and SF.KEY_AUTH.

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

- Backup of keys and users in accordance with the SFRs FDP_ACC.1/Backup and FDP_ACF.1/Backup (and iterations)

- Unblock of user accounts due to authentication failures in accordance with the SFR FMT_MTD.1/Unblock//User
- Unblock of cryptographic keys due to key authorisation failures in accordance with the SFR FMT_MTD.1/Unblock//Key
- Export of General (non-Assigned) keys in accordance with FDP_IFC.1 and FDP_IFF.1 (and iterations)
- Modifications of key attributes by authorised subjects in accordance with FDP_ACF.1 (and iterations)
- System time setting to support FPT_STM.1.
- The export and deletion of the audit log is performed in accordance with FMT_MTD.1/AuditLog.
- Software Update in accordance with the SFR FMT_MTD.1//SWUpdate.

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

9.4 SF.KEY_MAN: Key Management

Key management cannot be done without user authentication: Only if a defined authentication status has been obtained then key management tasks can be executed. In addition to that, for some key management functions the key usage has to be authorised before. The key management security function SF.KEY_MAN is therefore closely related to SF.USER_AUTH and SF.KEY_AUTH.

SF.KEY_MAN provides the following services by means of SF.CRYPTO fulfilling FDP_ACC.1.1, FMT_SMR.1 and parts of FMT_SMF.1:

- Generation and export of the Master Backup Key in accordance with the SFR FCS_CKM.1//AES and FDP_IFF.1/KeyBasics (authenticated by an Administrator)
- Import of the Master Backup Key (authenticated by an Administrator, and under dual person control)
- Backup and restore of keys (as required by FMT_SMF.1.1 (4), authenticated by a Key Manager or Internal SAM, secured with the Master Backup Key in order to fulfill FDP_IFF.1.5/KeyBasics (1); restore only possible under dual person control as required by FDP_ACF.1.2/Backup)
- Generation of keys (authenticated by a Key Manager or Internal SAM):
    - AES Keys in accordance with the SFR FCS_CKM.1//AES
    - ECDSA Keys in accordance with the SFR FCS_CKM.1//ECDSA
    - RSA Keys in accordance with the SFR FCS_CKM.1//RSA
- Deletion of keys (authenticated by a Key Manager or Internal SAM) in accordance with the SFR FCS_CKM.4
- Modification of key attributes as required by FMT_SMF.1.1 (2)
- Import and export of keys as required by FMT_SMF.1.1 (4) (authenticated by a Key Manager):
    - Import of keys in accordance with the rules in FDP_IFF.1/KeyBasics
    - Export of keys in accordance with the rules in FDP_IFF.1/KeyBasics

FDP_ACF.1/KeyUsage (Security attribute based Access Control) enforces the Key Usage SFP to authenticated users who are currently authorised to change attributes of secret key.

Management of security attributes of General keys and Assigned keys is performed in accordance with FMT_MSA.1/GenKeys (Management of security attributes of General keys, all iterations), FMT_MSA.1/AKeys (Management of security attributes of Assigned keys, all iterations) and FMT.MSA.3/Keys (Static attribute initialisation)

9.5 SF.CRYPTO: Cryptographic Support

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

SF.CRYPTO supports the following cryptographic operations:

- AES algorithm in CBC mode with a key length of 16, 24 or 32 bytes used for encryption or decryption in accordance with the SFRs FCS_COP.1//AES_Encryption_CBC and FCS_COP.1//AES_Decryption_CBC
- AES algorithm in OFB mode with a key length of 16, 24 or 32 bytes used for encryption or decryption in accordance with the SFR FCS_COP.1//AES_Encryption_OFB and FCS_COP.1//AES_Decryption_OFB
- AES algorithm in ECB mode with a key length of 16, 24 or 32 bytes used for encryption or decryption in accordance with the SFR FCS_COP.1//AES_ECB (for internal use only, to support an internal SAM)
- AES algorithm in GCM mode with a key length of 16, 24 or 32 bytes used for authenticated encryption or decryption in accordance with the SFR FCS_COP.1//AES_GCM
- AES algorithm with a key length of 16, 24 or 32 bytes used for CMAC generation and verification in accordance with the SFR FCS_COP.1//AES_CMAC
- ECDSA algorithm according to the standard [ANSI-X9.62] with key lengths of minimum 224 bit modulus lengths used for ECDSA signature generation or verification in accordance with the SFRs FCS_COP.1//ECDSA_Sign and FCS_COP.1//ECDSA_Verify
- RSA algorithm according to the standard [PKCS#1] with key lengths of minimum 2048[1] and maximum 8192 bit modulus lengths used for RSA encryption or decryption in accordance with the SFR FCS_COP.1//RSA_Encryption and FCS_COP.1//RSA_Decryption
- RSA algorithm according to the standard [PKCS#1] with key lengths of minimum 2048[1] and maximum of 8192 bit modulus lengths used for RSA signature generation and verification in accordance with the SFRs FCS_COP.1//RSA_Sign and FCS_COP.1//RSA_Verify
- HMAC calculation in accordance with the SFR FCS_COP.1//HMAC (HMAC key size shorter than 13 bytes for internal use only to support user authentication, key size 13 bytes and more also as cryptographic service)
- Hash algorithms SHA-224, SHA-256, SHA-384, SHA-512, SHA3-224, SHA3-256, SHA3-384 and SHA3-512 in accordance with the SFR FCS_COP.1//Hash
- Diffie-Hellmann key agreement in accordance with the SFR FCS_COP.1//Diffie-Hellman (for internal use only to support the implementation of the trusted channel)
- Key Derivation in accordance with the SFR FCS_COP.1//KeyDerivation (for internal use only to support the implementation of the trusted channel and the secure backup of keys)

- Random number generation by a hybrid RNG in accordance with the SFR FCS_RNG.1.

9.6 SF.REL: Reliability

SF.REL monitors the following events:

- Self-test error,
- Stored data integrity failure,
- Failure of user authentication or of key authorisation attempts,
- Results of services of SF.ADMIN, SF.KEY_MAN, SF.SWUPDATE,

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

SF.REL provides a service to query the audit records, this service has to be authenticated by a user in Administrator, User Administrator, Key Manager or Security Officer role, in accordance with FMT_MTD.1/AuditLog (Management of TSF data). The TOE does not provide any possibility to modify the audit records except for entire clearance, whereby the service for the clearance of the audit data has to be authenticated by a user in Administrator or User Administrator role, in accordance with the SFRs FAU_STG.2 (Guarantees of audit data availability) and FMT_MTD.1/AuditLog (Management of TSF data).

SF.REL preserves a secure operation state of the TOE when the following types of failures and attacks occur:

- Power supply too high/too low
- Temperature too high/too low
- Integrity check of cryptographic keys and stored firmware modules
- Self-test fails

The TOE provides an alarm mechanism which detects physical environmental failure attacks and reacts by destroying all sensitive data. For this mechanism a sensory is implemented which watches temperature and voltage.

Furthermore, the TOE with its tamper-evident enclosure (the heat sink and the potting material) implements the following physical security mechanisms against direct physical attacks:

- The cryptographic module’s hardware components are covered by hard, opaque potting material or the heat sink, which show evidence of tampering on the enclosure when a physical attack is attempted. This provides the capability to determine physical tampering according to FPT_PHP.1 (Passive detection of physical attack).
- The potting material is hard and opaque enough to prevent direct observation and easy penetration to the depth of the underlying hardware components. It is highly probable that anyone attempting to penetrate to the depth of the circuitry will break off large pieces of potting material and tear important hardware components off the module, causing serious damage to the TOE.

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

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

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

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

During the boot process after power-on or reset the TOE's boot loader and operating system SMOS perform further self-tests, like a memory RAM test. SMOS loads and initialises all remaining firmware modules and performs further self-tests in accordance with FPT_TST_EXT.1 (Basic TSF self-testing).

a)

It is only possible to execute any cryptographic or other security-relevant service after these power-on self-tests have been completed successfully. If one of these power-on self-tests fails, the TOE enters the secure Error State.

The TOE performs the following self-tests at specific conditions in accordance with FPT_TST_EXT.1 (Basic TSF self-testing):

a) Online Test of the digitised noise data of the PTRNG
b) Continuous DRNG tests (whenever random bytes are requested)
c) ECDSA Key Pair-wise Consistency Test (sign/verify) for any newly generated or imported ECDSA key pair according to FIPS 140-2 §4.9.2
d) RSA Key Pair-wise Consistency Tests (encrypt/decrypt and sign/verify) for any newly generated RSA key pair according to FIPS 140-2 §4.9.2
e) Firmware Load Test (via RSA signature verification) for every firmware module when being loaded

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

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

SF.REL monitors stored data and prohibits usage of altered data and notifies the user if integrity errors are detected in accordance to FDP_SDI.2.

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

9.7 SF.SWUPDATE: Software Update

SF.SWUPDATE allows to perform a secure software update on the TOE by providing the “Load File” service.

This service has to be authenticated by a user with the Administrator role, in accordance with FMT_MTD.1//SWUpdate.

The “Load File” service allows the download of firmware modules only in a dedicated format which contains also a signature calculated over the executable code (RSA signature according to [PKCS#1], with a key length of 4096 bit according to FCS_COP.1//RSA_Verify). The signature has to be calculated with a dedicated Module Signature Key owned by the manufacturer. If the signature cannot be verified, the download is prohibited and the “Load File” service will return an error code instead. If the set of loaded firmware modules is incomplete or in any way not compliant to the software that is released for this project, the TOE will be set to a secure Error State.

In this Error State no cryptographic operations are available, only status requests can be performed.

9.8 Coverage of SFRs by Security Functions

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

SFR	**SF.USER_AUTH**	**SF.KEY_AUTH**	**SF.ADMIN**	**SF.KEY_MAN**	**SF.CRYPTO**	**SF.REL**	**SF.SWUPDATE**
FCS_CKM.1//AES (Cryptographic key generation)				**X**	**X**		

SFR	SF.USER_AUTH	SF.KEY_AUTH	SF.ADMIN	SF.KEY_MAN	SF.CRYPTO	SF.REL	SF.SWUPDATE
FCS_CKM.1//RSA (Cryptographic key generation)				x	x		
FCS_CKM.1//ECDSA (Cryptographic key generation)				x	x		
FCS_CKM.4 (Cryptographic key destruction)				x		x	
FCS_COP.1//AES_Encryption_CBC (Cryptographic operation)					x		
FCS_COP.1//AES_Encryption_OFB (Cryptographic operation)					x		
FCS_COP.1//AES_Decryption_CBC (Cryptographic operation)					x		
FCS_COP.1//AES_Decryption_OFB (Cryptographic operation)					x		
FCS_COP.1//AES_CMAC (Cryptographic operation)					x		
FCS_COP.1//AES_ECB (Cryptographic operation)					x		
FCS_COP.1//AES_GCM (Cryptographic operation)					x		
FCS_COP.1//RSA_Sign (Cryptographic operation)					x		
FCS_COP.1//RSA_Verify (Cryptographic operation)					x		x
FCS_COP.1//RSA_Encryption (Cryptographic operation)					x		
FCS_COP.1//RSA_Decryption (Cryptographic operation)					x		
FCS_COP.1//ECDSA_Sign (Cryptographic operation)					x		

SFR	SF.USER_AUTH	SF.KEY_AUTH	SF.ADMIN	SF.KEY_MAN	SF.CRYPTO	SF.REL	SF.SWUPDATE
FCS_COP.1//ECDSA_Verify (Cryptographic operation)					x		
FCS_COP.1//HMAC (Cryptographic operation)					x		
FCS_COP.1//Hash (Cryptographic operation)					x		
FCS_COP.1//Diffie-Hellman (Cryptographic operation)					x		
FCS_COP.1//KeyDerivation (Cryptographic operation)					x		
FCS_RNG.1 (Generation of random numbers)					x		
FIA_UID.1 (Timing of identification)	x						
FIA_UAU.1//UserAuth (Timing of Authentication)	x						
FIA_UAU.1//KeyAuth (Timing of Authentication)		x					
FIA_AFL.1//UserAuth (User Authentication failure handling)	x						
FIA_AFL.1//KeyAuth (Key Authorisation failure handling)		x					
FIA_UAU.6/KeyAuth (Re-authentication)		x					
FDP_IFC.1/KeyBasics (Subset Information Control)			x				
FDP_IFF.1/KeyBasics (Simple security attributes)			x	x			
FDP_ACC.1/Key_Usage (Subset access control)		x	x	x			

SFR	SF.USER_AUTH	SF.KEY_AUTH	SF.ADMIN	SF.KEY_MAN	SF.CRYPTO	SF.REL	SF.SWUPDATE
FDP_ACF.1/KeyUsage (Security attrib. based access control)		x	x	x			
FDP_ACC.1/Backup (Security attrib. based access control)			x				
FDP_ACF.1/Backup (Security attrib. based access control)			x	x			
FDP_SDI.2 (Stored data integrity monitoring/action)						x	
FDP_RIP.1 (Subset residual information protection)						x	
FTP_TRP.1/Local (Trusted path)	x						
FTP_TRP.1/External (Trusted path)	x						
FPT_STM.1 (Reliable time stamps)			x			x	
FPT_TST_EXT.1 (Basic TSF self testing)						x	
FPT_PHP.1 (Passive detection of physical attack)						x	
FPT_PHP.3 (Resistance to physical attack)						x	
FPT_FLS.1 (Failure with preservation of secure state)						x	
FMT_SMR.1 (Security roles)	x		x	x			
FMT_SMF.1 (Security management functions)			x	x			
FMT_MTD.1/Unblock//User (Management of TSF Data)	x		x				

SFR	SF.USER_AUTH	SF.KEY_AUTH	SF.ADMIN	SF.KEY_MAN	SF.CRYPTO	SF.REL	SF.SWUPDATE
FMT_MTD.1/Unblock//Key (Management of TSF Data)		X	X				
FMT_MTD.1/AuditLog (Management of TSF Data)			X			X	
FMT_MTD.1//SWUpdate (Management of TSF Data)			X				X
FMT_MSA.1/GenKeys//AFlag (Management of security attributes)				X			
FMT_MSA.1/GenKeys//ExportF (Management of security attributes)				X			
FMT_MSA.1/GenKeys//AuthD (Management of security attributes)				X			
FMT_MSA.1/GenKeys//None (Management of security attributes)				X			
FMT_MSA.1/AKeys//AuthD (Management of security attributes)				X			
FMT_MSA.1/AKeys//None (Management of security attributes)				X			
FMT_MSA.3/Keys (Static attribute initialisation)				X			
FAU_GEN.1 (Audit data generation)						X	
FAU_GEN.2 (Identity association)						X	
FAU_STG.2 (Guarantees of audit data availability)						X	

Table 9: Mapping SFRs to Security Functions

10 Annex

This Annex contains the following sections:

Glossary and Acronyms

References

10.1 Glossary and Acronyms

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

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

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

Term	Description
Encrypted key	A cryptographic key that has been encrypted using an Endorsed security function with a key encrypting key, a PIN, or a password in order to disguise the value of the underlying plaintext key.
Encryption algorithm	Algorithm of processing a plaintext into a cipher text using an encryption key in a way that decoding of the cipher text into the plain text without knowledge of the corresponding decryption key is computationally infeasible.
Endorsed	For this security target, endorsed by the certification body for the evaluation of products of an intended type and resistance against attacks with attack potential addressed by the vulnerability analysis component in the security target[211].
Endorsed security function	For this security target, a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either a) specified in an Endorsed standard, b) adopted in an Endorsed standard and specified either in an appendix of the Endorsed standard or in a document referenced by the Endorsed standard, or c) specified in the list of Endorsed security functions.
Error detection code (EDC)	A code computed from data and comprised of redundant bits of information designed to detect, but not correct, unintentional changes in the data.
Error mode	Mode of operation when the cryptographic module has encountered an error condition as defined in FPT_FLS.1.
Error state	State related to the Error mode
Firmware	The programs and data components of a cryptographic module that are stored in hardware (e.g., ROM, PROM, EPROM, EEPROM or FLASH) and cannot be dynamically written or modified during execution.
Hardware	The physical equipment used to process programs and data.
Hash-based message authentication code (HMAC)	A message authentication code that utilises a keyed hash.

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

Term	Description
Information processing	The organisation, manipulation and distribution of information.
Initialisation vector (IV)	A vector used in defining the starting point of an encryption process within a cryptographic algorithm.
Input data	Information that is entered into a cryptographic module for the purposes of transformation or computation using an Endorsed security function.
Integrity	The property that sensitive data has not been modified or deleted in an unauthorised and undetected manner.
Internal secrets	Confidential data inside the cryptographic boundary not intended for export (e.g. secret or private plaintext keys, authentication reference data).
Internal TOE interface	Internal interface which is provided by the TOE and which can only be accessed by firmware modules running within the physical boundary of the TOE. It is intended to be used by an internal SAM.
Internal SAM	Signature Activation Module in the sense of [PP_QSCD] which is implemented as internal firmware module and which is eIDAS evaluated according to [PP_QSCD] and follows the TOE guidance.
Key establishment	The process by which cryptographic keys are securely distributed among cryptographic modules using manual transport methods (e.g., key loaders), automated methods (e.g., key transport and/or key agreement protocols), or a combination of automated and manual methods (consists of key transport plus key agreement).
Key management	The activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.
Key transport	Secure transport of cryptographic keys from one cryptographic module to another module.
Key usage type	Type of cryptographic algorithm a key can be used for (e.g. AES encryption, RSA signature-creation)
Key User	An individual (subject) that accesses a cryptographic module in order to obtain cryptographic services with a cryptographic key.

Term	Description
Logical external interface	A logical entry or exit point of a cryptographic module that provides access to the module for logical information flows representing physical signals (see also the term “port” for the physical aspects of a logical external interface). In the CC terminology it covers all logical external interfaces of the TOE (direct or indirect interfaces to the TSF or interfaces to the non-TSF portion of the TOE).
Maintenance mode	Mode of operation for maintaining and servicing a cryptographic module, including physical and logical maintenance testing.
Maintenance state	State related to the Maintenance mode .
Message authentication with appendix	A digital signature scheme which requires the message as input to the verification algorithm. The signature is attached to the message.
Microcode	The elementary processor instructions that correspond to an executable program instruction.
Operating conditions	Any environmental condition being accidental or induced outside of the normal range intended for the TOE may affect the correct operation or compromise of confidential information. These conditions include but are not limit to voltage of power supply, temperature, emanation which TOE environmental conditions.
Output data	Data containing information that is produced from a cryptographic module.
Password	A string of characters (letters, numbers, and other symbols) used to authenticate an identity or to verify access authorisation.
Permanent stored keys	Keys remains stored in the TOE after power off or reset.
Physical protection	The safeguarding of a cryptographic module, including its cryptographic keys and other critical security parameter, using physical means.
Plaintext key	An unencrypted cryptographic key.
Port	A physical input or output interface of a cryptographic module that provides access to the module for physical signals, represented by logical information flows. Physically separated ports do not share the same physical pin or wire. In the CC terminology a port is a physical external interface of the TOE (direct or indirect interface to the TSF or interface to the non-TSF portion of the TOE).
Power interface/port	Interface respective port providing all external electrical power supply.

Term	Description
Power On/Off mode	Mode of operation that indicates whether the cryptographic module is supplied by a power source. These modes may distinguish between different power sources (e.g., primary, secondary, backup power source or none) being applied to a cryptographic module.
Power On/Off state	State related to the Power On/Off mode (cf. ADV_ARC.1).
Private key	A cryptographic key, used with a public key cryptographic algorithm, that is uniquely associated with an entity and is not made public.
Protection Profile	An implementation-independent set of security requirements for a category of Targets of Evaluation (TOEs) that meet specific consumer needs.
Public key	A cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that may be made public.
Public key (asymmetric) cryptographic algorithm	A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible.
Public key certificate	A set of data that uniquely identifies an entity, contains the entity's public key, and is digitally signed by a trusted party, thereby binding the public key to the entity.
Random Number Generator	Random Number Generators (RNGs) used for cryptographic applications produce a sequence of zero and one bits that may be combined into sub-sequences or blocks of random numbers. There are three basic classes physical true RNG, non-physical true RNG, and deterministic RNG. A physical true RNG produces output that dependents on some physical random source inside the TOE boundary only. A non-deterministic true RNG gets its entropy from sources from outside the TOE boundary (e.g. by system data like RAM data or system time of a PC, output of API functions etc., or human interaction like key strokes, mouse movement etc.). A deterministic RNG consists of an algorithm that produces a sequence of bits from an initial random value (seed).
Reference authentication data	Data known for the claimed identity and used by the TOE to verify the verification authentication data provided by an entity in an authentication attempt to prove their identity.
Reset	Action to clear any pending errors or events and to bring a system to normal condition or initial state (e.g. after power-on).

Term	Description
SAEK signature interface	Part of the internal TOE interface using SAM Authorised External Keys. This interface for signature calculation is intended to be used by an internal SAM which itself validates key authorisation. See Application Note 1 (ST).
SAM	Signature Activation Module in the sense of [PP_QSCD]
Secret key	A cryptographic key, used with a secret key cryptographic algorithm that is uniquely associated with one or more entities and should not be made public.
Secret key (symmetric) cryptographic algorithm	A cryptographic algorithm the keys of which for both encryption and decryption respective MAC calculation and MAC verification are the same or can easily be derived from each other and therefore must be kept secret.
Seed key	A secret value used to initialise a cryptographic function or operation.
Self-test mode	Mode of operation in which the cryptographic module performs initial start-up self-test, self-test at power-on, self-test at the request of the authorised user and may perform other self-tests identified in FPT_TST_EXT.1
Self-test state	State related to the Self-test mode (cf. ADV_ARC.1).
Shutdown	Shutdown of the TOE initiated by the user (may not include reset after detection of error or power-off due to loss of power supply)
Signature-creation key	Private key for the creation of digital signatures
Signature-verification key	Public key for the verification of digital signatures
Software	The programs and data components, usually stored on erasable media (e.g., disk), that can be dynamically written and modified during execution.
Split knowledge	A process by which a cryptographic key is split into multiple key components, individually sharing no knowledge of the original key, that can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.
Status information	Information that is output from a cryptographic module for the purposes of indicating certain operational characteristics or modes of the module.

Term	Description
Status output interface/port	Interface respective port intended for all input commands, signals, and control data (including calls and manual controls such as switches, buttons, and keyboards) used to control the operation of the cryptographic module).
System software	The special software within the cryptographic boundary (e.g., operating system, compilers or utility programs) designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system, and associated programs, and data.
Tamper detection	The automatic determination by a cryptographic module that an attempt has been made to compromise the physical security of the module.
Target of Evaluation (TOE)	An information technology product or system and associated administrator and user guidance documentation that is the subject of an evaluation.
Timing analysis	Analysis of timing behaviour of a device, equipment, or system to gain information about its internal secrets or processes
TOE Security Functionality (TSF)	Combined functionality of all hardware, software, and firmware of a TOE that must be relied upon for the correct enforcement of the SFRs.
TOE security functions interface (TSFI)	A set of interfaces, whether interactive (man-machine interface) or machine (machine-machine interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.
Trusted channel	A means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSF.
Trusted path	A means by which a user and a TSF can communicate with necessary confidence to support the TSF.
Unauthenticated User	An identified user not being authenticated and having rights as identified in the component FIA_UAU.1//UserAuth.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE (includes both authenticated and unauthenticated entities).

Table 10: Glossary

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

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

Table 11: Acronyms

10.2 References

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

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

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

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

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

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

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

[ECCBP] ECC Brainpool Standard Curves and Curve Generation, v1.0, 19.10.2005 / ECC Brainpool, http://www.ecc-brainpool.org/ecc-standard.htm

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

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

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

[FIPS 197] FIPS PUB 197, Advances Encryption Standard (AES) / National Institute of Standards and Technology (N11IST), USA, 26th November 2001

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

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

[ISO 19790:2012]	ISO/IEC 19790:2012(E): Information Technology — Security Techniques — Security requirements for cryptographic modules / International Organization for Standardization, Geneva, Switzerland, 15th August 2012
[ANSSI]	ANSSI: "Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français" in: Journal Officiel de la République Française (JORF), n° 0241 du 16 octobre 2011 page 17533 text n° 30 (Announcement about elliptic curve parameters set by the French government). NOR: PRMD1123151V. Available: https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000024668816
[NIST SP 800-38A]	NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques / National Institute of Standards and Technology (NIST), USA, December 2001
[NIST SP 800-38B]	NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication / National Institute of Standards and Technology (NIST), USA, May 2005
[NIST SP 800-108]	NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions (Revised) / National Institute of Standards and Technology (NIST), USA; October 2009
[PKCS#1]	PKCS#1: RSA Cryptography Standard v2.2, 27thOctober 2012 / RSA Laboratories, http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm
[PKCS#3]	PKCS#3: Diffie-Hellman Key Agreement Standard, v1.4; 1st November 1993 / RSA Laboratories, http://www.rsasecurity.com/rsalabs/pkcs
[RFC 2104]	RFC 2104: HMAC: Keyed-Hashing for Message Authentication, Internet Engineering Task Force (IETF), February 1997
[PP_CMTS]	EN 419 221-5:2018 Protection Profiles for TSP Cryptographic Modules – Part 5: Cryptographic Module for Trust Services, v1.0, registered under the reference ANSSI-CC-PP-2016/05-M01, 18 May 2020
[PP_QSCD]	prEN 419 241-2: Trustworthy Systems Supporting Server Signing Part 2: Protection Profile for QSCD for Server Signing; CEN April 2019
[Regulation]	REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC
[SOG-IS-Crypto]	SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, v1.3, February 2023
[Shamir]	A. Shamir, How to share a secret, Communications of the ACM, 22 (1979), 612-613
[TR03111]	Technical Guideline TR-03111, Elliptic Curve Cryptography, Version 1.11, April 2009 / Bundesamt für Sicherheit in der Informationstechnik (BSI)

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