EVIDEN

rue Jean Jaurès
78340 Les Clayes-sous-Bois
FRANCE

Trustway

Trustway Proteccio™

SECURITY TARGET

Document Reference : PCA4_0142
Date : November 26, 2024
Version : 1.09

Summary

Summary 2

Illustration table 6

Chapter 1. Introduction 7

1.1 Introduction 7

1.2 References... 7

1.3 Glossary 10

Chapter 2. ST introduction 12

2.1 ST identification...12

2.2 ST overview 12

2.3 CC conformance 14

2.4 PP conformance...14

2.5 Enhanced Qualification conformance 14

Chapter 3. TOE Description 15

3.1 Product type 15

3.2 Architecture 17

3.3 Life cycle 20

3.4 TOE boundary 22

3.5 TOE functionalities 22

3.5.1 Cryptographic operations 22

3.5.2 Algorithmes cryptographiques 23

3.5.3 Key sizes supported by the TOE...24

3.5.4 Key management...24

3.5.5 Roles..25

3.5.6 Administration 28

3.5.7 TOE installation 28

3.5.8 TOE personalization...28

3.5.9 CIK activation...29

3.5.10 Test of critical functions 29

3.6 Protection of the network link between applications and TOE......................29

3.6.1 TLS secure channel between client applications and the TOE 29

3.6.2 Additional secure channel between client applications and the TSF ..30

3.7 TOE delivery 30

3.8 Configuration recommendation 32

3.8.1 PP 419221-5 compliant cryptographic configuration 32

3.9 Non-TOE hardware/software/firmware required by the TOE.......................34

Chapter 4. Security Problem Definition 35
4.1 Assets 35
4.2 Subjects 35
4.3 Threats 36
4.3.1 General 36
4.3.2 T.KeyDisclose — Unauthorised disclosure of secret/private key 36
4.3.3 T.KeyDerive — Derivation of secret/private key 36
4.3.4 T.KeyMod — Unauthorised modification of a key 36
4.3.5 T.KeyMisuse — Misuse of a key 36
4.3.6 T.KeyOveruse — Overuse of a key 37
4.3.7 T.DataDisclose — Disclosure of sensitive client application data 37
4.3.8 T.DataMod — Unauthorised modification of client application data .. 37
4.3.9 T.Malfunction — Malfunction of TOE hardware or software 37
4.4 Organisational Security Policies 37
4.4.1 P.Algorithms — Use of approved cryptographic algorithms 37
4.4.2 P.KeyControl — Support for control of keys 38
4.4.3 P.RNG — Random Number Generation 38
4.4.4 P.Audit — Audit trail generation 38
4.5 Assumptions 39
4.5.1 A.ExternalData — Protection of data outside TOE control 39
4.5.2 A.Env — Protected operating environment 39
4.5.3 A.DataContext — Appropriate use of TOE functions 39
4.5.4 A.UAuth — Authentication of application users 40
4.5.5 A.AuditSupport — Audit data review 40
4.5.6 A.AppSupport — Application security support 40
Chapter 5. Security Objectives 41
5.1 General 41
5.2 Security Objectives for the TOE 41
5.2.1 General 41
5.2.2 OT.PlainKeyConf — Protection of confidentiality of plaintext secret keys 41
5.2.3 OT.Algorithms — Use of approved cryptographic algorithms 41
5.2.4 OT.KeyIntegrity — Protection of integrity of keys 42
5.2.5 OT.Auth — Authorization for use of TOE functions and data 42
5.2.6 OT.KeyUseConstraint — Constraints on use of keys 42
5.2.7 OT.KeyUseScope — Defined scope for use of a key after authorization 43
5.2.8 OT.DataConf — Protection of confidentiality of sensitive client application data 43
5.2.9 OT.DataMod — Protection of integrity of client application data 43
5.2.10 OT.ImportExport — Secure import and export of keys 44
5.2.11 OT.Backup — Secure backup of user data 44
5.2.12 OT.RNG — Random number quality 44
5.2.13 OT.TamperDetect — Tamper Detection 44
5.2.14 OT.FailureDetect — Detection of TOE hardware or software failures 44

5.2.15 OT.Audit — Generation of audit trail 45
5.3 Security Objectives for the Operational Environment 45
5.3.1 General 45
5.3.2 OE.ExternalData — Protection of data outside TOE control 45
5.3.3 OE.Env — Protected operating environment 45
5.3.4 OE.DataContext — Appropriate use of TOE functions 46
5.3.5 OE.Uauth — Authentication of application users 46
5.3.6 OE.AuditSupport — Audit data review 46
5.3.7 OE.AppSupport — Application security support 47

Chapter 6. Extended Components Definitions 48

6.1 Generation of random numbers (FCS_RNG) 48
6.1.1 General 48
6.1.2 Family behaviour 48
6.1.3 Component levelling 48
6.2 Basic TSF Self Testing (FPT_TST_EXT.1) 49
6.2.1 General 49
6.2.2 Family behaviour 49
6.2.3 Component levelling 49

Chapter 7. Security Requirements 50

7.1 General 50
7.2 Typographical Conventions 50
7.3 SFR Architecture 50
7.3.1 SFR Relationships 50
7.3.2 SFRs and the Key Lifecycle 52
7.4 Security Functional Requirements 54
7.4.1 General 54
7.4.2 Cryptographic Support (FCS) 54
7.4.3 Identification and authentication (FIA) 61
7.4.4 User data protection (FDP) 66
7.4.5 Trusted path/channels (FTP) 72
7.4.6 Protection of the TSF (FPT) 74
7.4.7 Security management (FMT) 76
7.4.8 Security audit data generation (FAU) 84
7.5 Security Assurance Requirements 87
7.5.1 General 87
7.5.2 Refinements of Security Assurance Requirements 89
7.5.3 Configuration for PP-compliance 94

Chapter 8. Rationales 95

8.1 Security Objectives Rationale 95
8.1.1 Security Objectives Coverage 95
8.1.2 Security Objectives Sufficiency 95
8.2 Security Requirements Rationale 97

8.2.1 Security Requirements Coverage 97
8.2.2 SFR Dependencies 100
8.2.3 Rationale for SARs 103

Illustration table

Figure 1 - Bull Trustway Proteccio™ front panel 15
Figure 2 - Bull Trustway Proteccio™ rear panel 15
Figure 3 - Trustway Proteccio™ Architecture 17
Figure 4 - Trustway Proteccio™ front panel 18
Figure 5 - Trustway Proteccio™ rear panel 19
Figure 6 - TOE and environment general overview 22
Figure 7 - Generation of Random numbers - Component Levelling 48
Figure 8 - Basic TSF Self Testing – Component Levelling 49
Figure 9 - Architecture of User, TSF Protection and Audit SFRs 51

Chapter 1. Introduction

1.1 Introduction

The aim of this document is to describe the security target of the general purpose hardware security module (HSM) developed and manufactured by Bull Trustway, integrated in a secure communications appliance called Trustway Proteccio™. The appliance is connected to the host system through a Gigabit Ethernet interface. It comprises a network processor (ComExpress) and a cryptographic processor (FPGA).

This security target is conformant with Common Criteria Version 3.1.

1.2 References

1. **Règles et recommandations concernant le choix et dimensionnement des mécanismes cryptographiques** version 2.04, 1er Janvier 2020.
2. **Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques Version** 2.00, June 8th 2012.
3. **Règles et recommandations concernant les mécanismes** Version 1.0, 13 Janvier 2010.
4. **ISO/IEC 15408-1:200991,** Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model
5. **ISO/IEC 15408-2:200892,** Information technology — Security techniques — Evaluation criteria for IT security — Part 2: Security functional components
6. **ISO/IEC 15408-3:200893,** Information technology — Security techniques — Evaluation criteria for IT security — Part 3: Security assurance components
7. **CEN/ISSS WS/E-Sign; Area D1, CWA 14167-1 : 2003**: Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures
8. **prTS419221-2 : 2015** (Cryptographic Module for CSP Signing Operations with Backup – Protection Profile).
9. **prTS419221-3 : 2015** (Cryptographic Module for CSP key generation services).
10. **Règlement (UE) n ° 910/2014 du Parlement européen et du Conseil** du 23 juillet 2014 sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur et abrogeant la directive 1999/93/CE
11. **ETSI SR 002 176 - Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures** V1.1.1 (2003-03)
12. **European Telecommunications Standards Institute Technical Specification, ETSI TS 101 456 Policy requirements for certification authorities issuing qualified certificates** V1.2.1 2002-04

13. **FIPS 46-3 Data Encryption Standard (DES)**
    October 25, 1999
14. **Recommendation for Random Number Generation Using Deterministic random bit Generators** NIST Special Publication 800-90A Rev1, June 2015
15. **FIPS PUB140-3 Security requirements for cryptographic modules**
    March 22, 2019
16. **FIPS 180-4 Secure hash standard**
    2015-08
17. **FIPS PUB 186-4 Digital Signature Standard**
    July 2013
18. **RFC 1321The MD5 Message-Digest algorithm**
    April 1992
19. **RFC 2104 HMAC: Keyed-Hashing for message Authentication**
    February 1997
20. **ISO 9797-1Message Authentication Codes (MACs) part 1 - Mechanisms using a block cipher**
    Second edition 2011-03
21. **PKCS#1 RSA Cryptography Standard V2.2**
    October 2012 (RFC8017; November 2016)
22. **PKCS#8 Private-Key information syntax standard V1.2**
    (RFC5208; May 2008)
23. **PKCS#11Cryptographic Token interface standard V2.40**
    14 April 2015
24. **86A276FH - Trustway Proteccio - Installation and user guide** (English version) and
    86F276FH - Trustway Proteccio - **Manuel d'installation et d'utilisation** (French version)
    Version 71, November 2024
25. **86A275FH – Trustway Proteccio – Developer's guide**
    Version 70, October 2024
26. **NF EN 419221-5 : Protection profiles for Trust Service Provider Cryptographic modules - Part 5 : cryptographic Module for Trust Services** – may 2018
27. Common Methodology for Information Technology Security Evaluation, Evaluation Methodology (Version 3.1 Revision 4, September 2012), **CCMB-2012-09-004 [CEM]**
28. **CWA 14167-194,** Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures
29. **CWA 14170:2004,** Security requirements for signature creation applications
30. **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
31. **ETSI/TS 119 312**, Electronic Signatures and Infrastructures (ESI); Cryptographic Suites

32. **EN 419241-1**, Trustworthy Systems Supporting Server Signing — Part 1: General System Security Requirements
33. **EN 419241-2**, Trustworthy Systems Supporting Server Signing — Part 2: Protection profile for QSCD for Server Signing
34. **SOG-ISCrypto** SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, v1.0, May 2016
35. **AIS 31**, Version 2.1, december 2011
36. **86XY05GA,** Enabling PP5 Compliance mode. Version 3, August 2024
37. **86A202GA,** Migration_to_V193_and_higher. Version 3 July 2024

1.3 Glossary

Acronym	Definition
AES	Advanced Encryption Standard
Assets	Entities that the owner of the TOE presumably places value upon
Assurance	Grounds for confidence that a TOE meets the SFRs
Augmentation	Addition of one or more requirement(s) to a package
CC	Cryptographic component or Common Criteria
Certificate	Electronic attestation which links the SVD to a person and confirms the identity of that person
CIK	Crypto Ignition Key
CGA	Certificate Generation Application
CPLD	Complex Programmable Logic Device
CRC	Cyclic Redundancy Check
CSP	Certification Service Provider
CSP_SCD	Signature Creation Data used by a CSP
CSP_SVD	Signature Verification Data used by a CSP
DH	Diffie Hellman
DTBS	Data To Be Signed
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography or Error Correcting Code
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
Evaluation	Assessment of a PP, an ST or a TOE, against defined criteria
HMAC	Keyed-Hash Message Authentication Code
HSM	Hardware Security Module
IT	Information Technology
MCS	Secure Microcontroller
Evaluation Assurance level (EAL)	Set of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale, that form an assurance package
Security Objective	Statement of an intent to counter identified threats and/or satisfy identified organisation security policies and/or assumptions
OSP	Organisational Security Policy
Protection Profile (PP)	Implementation-independent statement of security needs for a TOE type
RAD	Reference Authentication Data
RSA	Rivest Shamir Adelman
SAR	Security Assurance Requirements

Acronym	Definition
Security Target	Implementation-dependent statement of security needs for a specific identified TOE
SF	Security Function
SFP	Security Function Policy
SFR	Security functional requirements
SHA	Secure Hash Algorithm
SCA	Signature Creation Application
SCD	Signature-creation data
SMC	Smartcard
SO	Security Officer
SOF	Strength of Function
SSCD	Secure Signature Creation Device
ST	Security Target
SVD	Signature Verification Data
TDM	Trustway Domain Management
TOE	Target of Evaluation: set of software, firmware and/or hardware possibly accompanied by guidance
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE
User data	Data created by and for the user that does not affect the operation of the TSF
VAD	Verification Authentication Data

Table 1-1. Acronyms

Chapter 2. ST introduction

2.1 ST identification

Title: **BULL Trustway HSM - Security Target**

Author: BULL SAS

TOE versions:

- **Trustway Proteccio™ EL/HR/XR :**
  - CDROM : 4.05.xx (the minor "xx" version of the CDROM is out of the scope of this ST)
  - System version: X194
  - Security Module version : V194
- **Trustway Proteccio™ EL :**
  - Hardware : 76681604-004D/76681604-004E/76681604-004G/76681604-005/76681604-105/76681604-115/76681604-116/76681604-126/76681604-226/76681604-227
  - MCS version : 1.03/1.04
- **Trustway Proteccio™ HR :**
  - Hardware : 76681610-004D/76681610-004E/76681610-004G/76681610-005/76681610-105/76681610-115/76681610-116/76682063-015/76682506-026/76681610-126/76681610-226/76682506-226/76682506-227/76681610-227
  - MCS version : 1.03/1.04
- **Trustway Proteccio™ XR :**
  - Hardware : 76682802-221
  - MCS version : 4.05

TOE commercial name: **Trustway Proteccio™**

Associated User and Development Guides: **24, 25, 36, 37**.

2.2 ST overview

The aim of this document is to describe the Security Target of the Bull Trustway HSM, integrated in a secure communications appliance.

Bull Trustway HSM is intended to be used as a cryptographic security module that can be used to produce key material and digital signatures for qualified certificates but also as a general-purpose hardware security module for key management and for various cryptographic operations (encryption, signature, message hash, cryptographic key wrapping …).

The main objectives of this ST are:

- To describe the Target-of-Evaluation (TOE).

- To describe the security environment of the TOE including the assets to be protected and the threats to be countered by the TOE and its environment.
- To describe the security objectives of the TOE and its supporting environment.
- To specify the security requirements which include the TOE security functional requirements and the TOE security assurance requirements.
- To specify the TOE summary specification, which includes the TOE security functions specifications and the assurance measures.

2.3 CC conformance

The ST is compliant to Part 2 [5] extended and Part 3 [6] of Common Criteria v3.1 rev5.

The assurance level for this ST is **EAL4**, augmented with:

- ADV_IMP.2 (Complete mapping of the implementation representation of the TSF),
- ALC_CMC.5 (Advanced Support),
- ALC_DVS.2 (Sufficiency of security measures),
- ALC_FLR.3 (Systematic Flaw Remediation),
- AVA_VAN.5 (Advanced methodical vulnerability analysis).

2.4 PP conformance

The ST claims a strict compliance to Protection Profile prTS 419221-5:2018 (Protection profiles for TSP Cryptographic modules - Part 5).

The Security Problem Definition (SPD), objectives and SFR are strictly identical to those in the PP.

2.5 Enhanced Qualification conformance

The ST is compliant to the French "Enhanced" qualification process [1] and thus conforms to the associated referential for "Enhanced" strength level edited by ANSSI:

- Cryptographic referential [1]
- Key management architecture referential [2]
- Authentication referential [3]

Chapter 3. TOE Description

3.1 Product type

Bull HSM is a high performance network-attached hardware security module that is part of a general purpose HSM appliance commercially available under the brand 'Trustway Proteccio'.

It is contained in its own secure enclosure that provides physical resistance to tampering and zeroisation of plaintext key material and security parameters in the event a tamper signal is received.

There are three models of Trustway Proteccio™:

- An entry level model (EL) with a Com Express module using an ATOM processor and an ARRIA2GX125 FPGA.
- A high range model (HR) with a Com Express module using a Core 2 Duo or Core I3 processor and an ARRIA2GX260 FPGA.
  - An extreme range model (XR) with a Com Express module using a Core I5 and a 10AX066H2F34E1HG FPGA.

Figures 1 and 2 shows the Trustway Proteccio™ appliance:

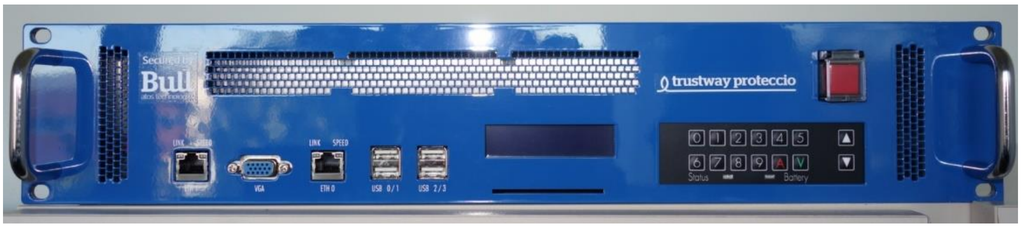


Figure 1 - Bull Trustway Proteccio™ front panel

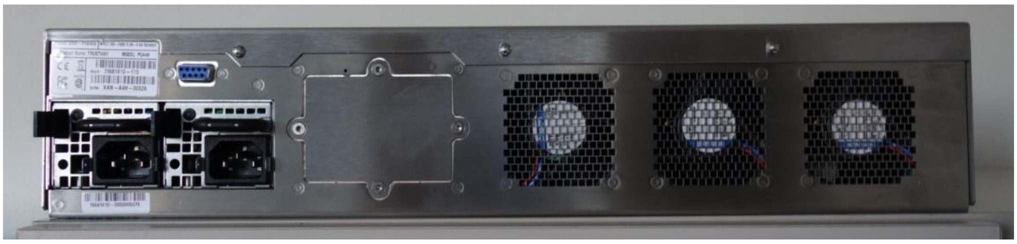


Figure 2 - Bull Trustway Proteccio™ rear panel

The HSM provides cryptographic functions for:

- Encryption and decryption;
- Digital signature and verification;
- Key management (including key generation and secure key storage).

The operating system supported into the appliance is Linux.

The operating systems supported on the client side are:

- Linux;
- Windows.

The HSM (TOE) is a cryptographic module suitable for use by trust service providers supporting electronic signature and electronic sealing operations, certificate issuance and revocation, time stamp operations, and authentication services (including support of authentication of client applications or authorized users of secret keys, and support of authentication for electronic identification), as identified by the (EU) No 910/2014 regulation of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market (eIDAS).

The Trustway Proteccio™ HSM cryptographic API on the client side is PKCS#11[25]. Proprietary APIs are also proposed to implement the HSM management services and other custom functions.

Bull HSM has a "Trusted Path" external interface to enable the connection of a device including keyboard/display/smartcard reader. This interface is internal to the appliance and the device keyboard/display/smartcard reader is integrated into the appliance.

Bull HSM has an opening detection mechanism that triggers the internal security alarm, making it difficult to open the enclosure without detection.

Critical component of Bull HSM are protected by a hard opaque potting material (resin) as stated in the FIPS 140-3 standard.

The HSM has, while in power on state, an emergency erase button which provokes its depersonalization.

The protection of secret elements is provided by a CIK mechanism at power on. The CIK activation mode can be configured during the personalization phase. Two modes are possible: smart card CIK – automatic CIK (for a start/restart without operator intervention and without the need of a smart card).

The HSM can be partitioned in virtual HSM while guaranteeing the strong compartmentalization of cryptographic key structures. The compliance mode with PP 419221-5 must be enabled on a Virtual HSM. Within one physical HSM, it is possible to configure only some virtual HSMs in PP compliance mode.

The HSM generates an audit record of all events related to the TOE start-up and initialisation, key management (generation, destruction …) and security (notification of physical attacks, unsuccessful self-tests …). There are 3 different audit files:

- An audit file related to the whole equipment Trustway Proteccio™, associated to the role Auditor;
- A security audit file, associated to the role Auditor;
- An audit file related to each virtual HSM, associated to the role HSM Auditor.

3.2 Architecture

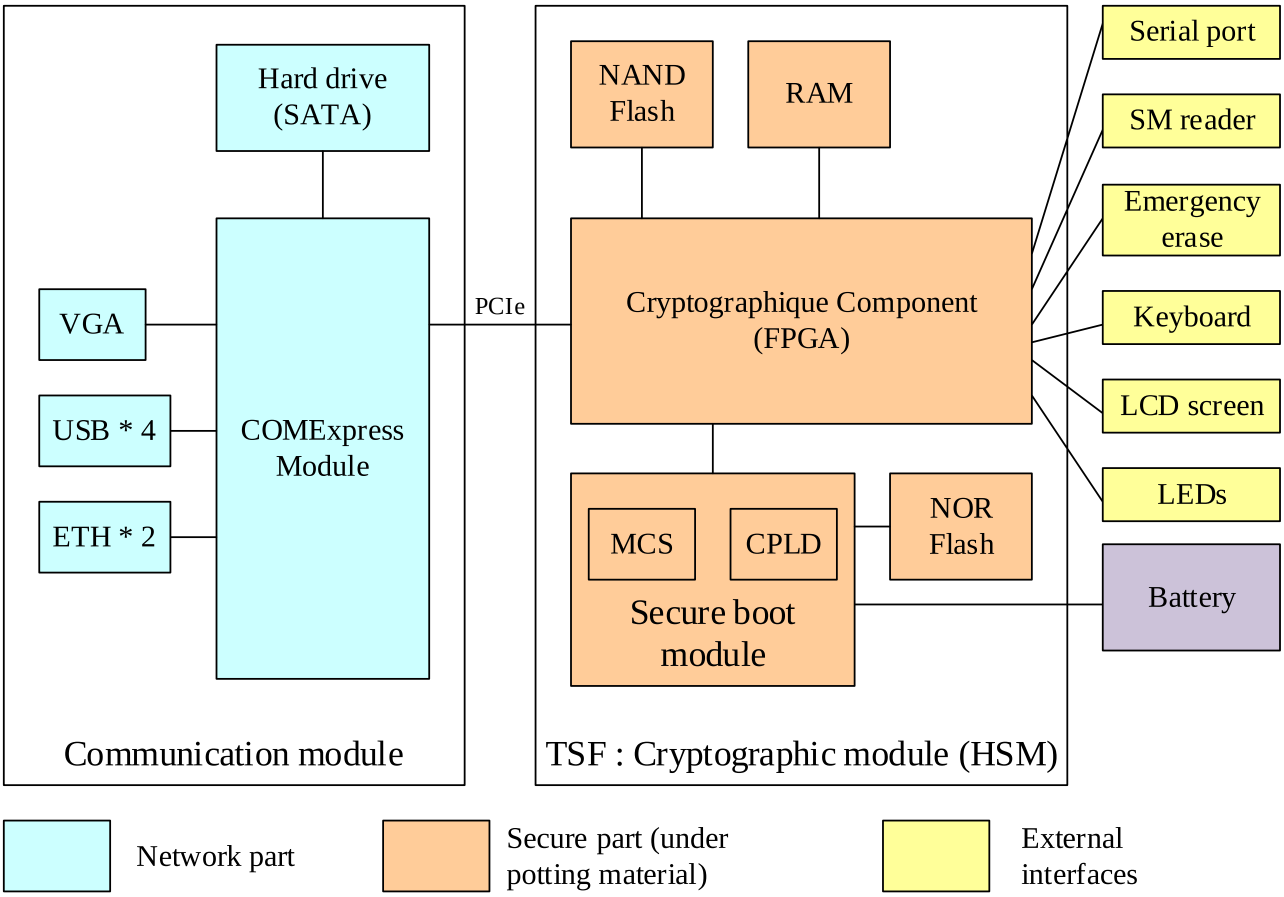


Figure 3 - Trustway Proteccio™ Architecture

Trustway Proteccio™ is a 2U high, 19" rack-mounted, secure network appliance with integrated power supply and interfaces in the front panel. It is connected to the host system through one or two ETHERNET Gigabit interfaces and integrates a network processor (ComExpress) and a cryptographic processor (FPGA).

The appliance contains:

- An electronic board (mother board), which includes:
  - A network processor implemented under the form of a Com Express module with an INTEL processor running the Linux operating system,
  - An Ethernet component 10/100/1000 with PCI Express interface,
  - An FPGA cryptographic component called CC, implementing three NiosII processors and all symmetrical and asymmetrical cryptographic IPs,
  - An ATMEL microcontroller (MCS) used for the secure boot, secure loading of the FPGA bitstream, alarm and tamper management as well as secure key storage,
  - DDR2/4 ECC protected RAM (Error Correction Code), storing the code, data and keys for the CC,

- 128 MB NOR FLASH including the FPGA bitstream and the CC software,
- 2 GB NAND FLASH to store the black keys,
- A CPLD to share the access to the NOR flash and the LCD screen between the CC and MCS.

- A 2.5-inch SATA hard drive connected to the ComExpress module,
- A 3.5V lithium battery for the secure storage of the base key in the MCS and the alarms and tamper management,
- An ATX AC/DC power supply of at least 100 W,
- 2 or 3 fans respectively for EL and HR/XR models.

The external interfaces are:

<u>Front:</u>

- 2 RJ45 Ethernet Interfaces, of which only one is used for cryptographic operations (eth0)
- 1 VGA Interface linked to the communication module
- 4 USB 2.0 interfaces (keyboard, mouse, external drive...)
- 1 smart card reader linked to the CC
- 1 LCD display linked to the CPLD
- 1 16-key keyboard linked to the CC
- 1 emergency stop button linked to the CC
- 1 status two-colour LED (Ready/Error/Alarm) linked to the CC
- 1 battery weak-status LED linked to the CC

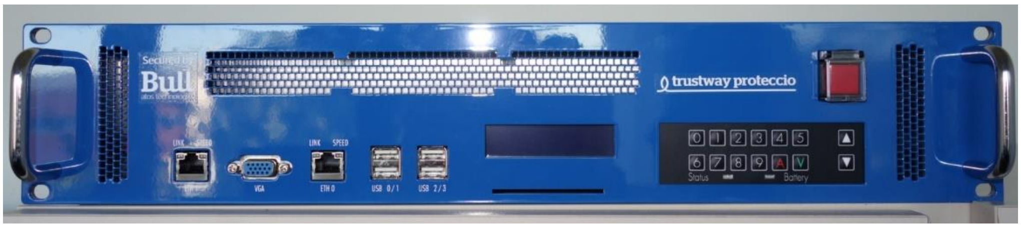


Figure 4 - Trustway Proteccio™ front panel

Rear:

- The 220V power supply connector
- 1 switch for 220V supply
- 1 DB9 serial link connector

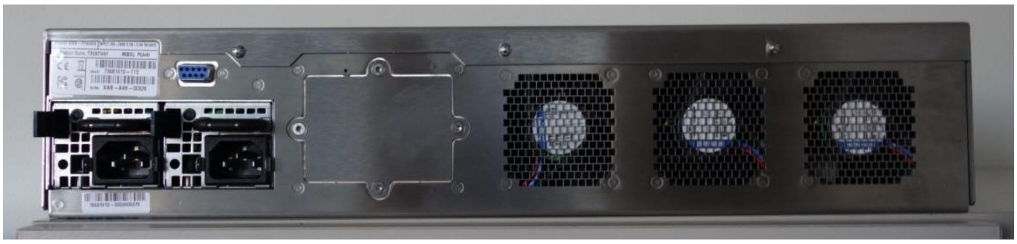


Figure 5 - Trustway Proteccio™ rear panel

The Smart card used is the IDeal Citiz 2.17 from IDEMIA (certified EAL5+, Certification Report ANSSI-CC-2019/04, ANSSI Enhanced Qualification 17950/ANSSI/SDE/PSS/BQA).

3.3 Life cycle

Trustway Proteccio™ life cycle can be divided into 9 phases:

Phase		Phase Responsibility	Phase Environment
Phase 1	Development	The development team (Bull) is in charge of the hardware design and the embedded software development and signing	This phase is executed in the development environment (under the responsibility of the developer)
Phase 2	Software signature	All signatures use asymmetric Elliptic curves (ECC) signatures with EC-KCDSA protocol Bull Les Clayes sous Bois	This phase is executed in Bull premises (under the responsibility of the integrator)
Phase 3	Manufacturing	The HSM manufacturing process is performed by subcontractors: • Manufacture of printed circuit motherboard (SOMACIS); • Components assembly on the motherboard and motherboard tests (ASTEEL); • Enclosure manufacturing (OTIMA) ; • Repeating the tests within the enclosure (ASTEEL).	Production
Phase 4	Tests and preparation by Bull	Bull manufactory (Angers) performs additional tests and prepares the HSM for the next phase (that will occur upon client purchase order).	This phase is executed in Bull Angers premises
Phase 5	Pre-personalization	This phase ensures the FPGA and COMExpress module software update (with the operational version), and the injection of pre-personalization elements. At the end of this step, Trustway Proteccio™ is ready to be delivered to the client, for personalization.	This phase is executed in Bull Angers premises
Phase 6	Delivery to the client	The Trustway Proteccio™ and the associated guides (documentation) is sent to the client.	
Phase 7	Personalization	Personalization by the client.	

Phase		Phase Responsibility	Phase Environment
Phase 8	Embedded software update	If needed, the end user (security officer) can update the HSM embedded software.	These phases are executed within the end user environment (under the responsibility of the end user)
Phase 9	TOE use	This last phase is executed by the end user.	

The evaluation (ALC class) perimeter circumscribes to phases 1 to 6 and does not include the personalization, configuration and embedded software update processes, executed into the end user environment.

Upon detection of an intrusion attempt, the TOE must be returned to Bull logistic centre (Angers)to be re-personalized (phase 5) in order to assure service continuity.

3.4 TOE boundary

The boundary of the TOE described in this ST encompasses the following:

- The cryptographic FPGA component.
- The smart card reader, housed into the appliance, which provides a trusted path for the communication of critical security parameters (authentication data) to the cryptographic module.
- The Linux operating system, which includes a specific driver, the PKCS #11 cryptographic API (under the form of a Linux library), which provide the programming and communications interface normally used to access the cryptographic module.
- The network interface allowing the communication between the client applications (including the administration application) and the TOE

- User and Administrative Guidance documentation for the TOE, provided on a CD-ROM.

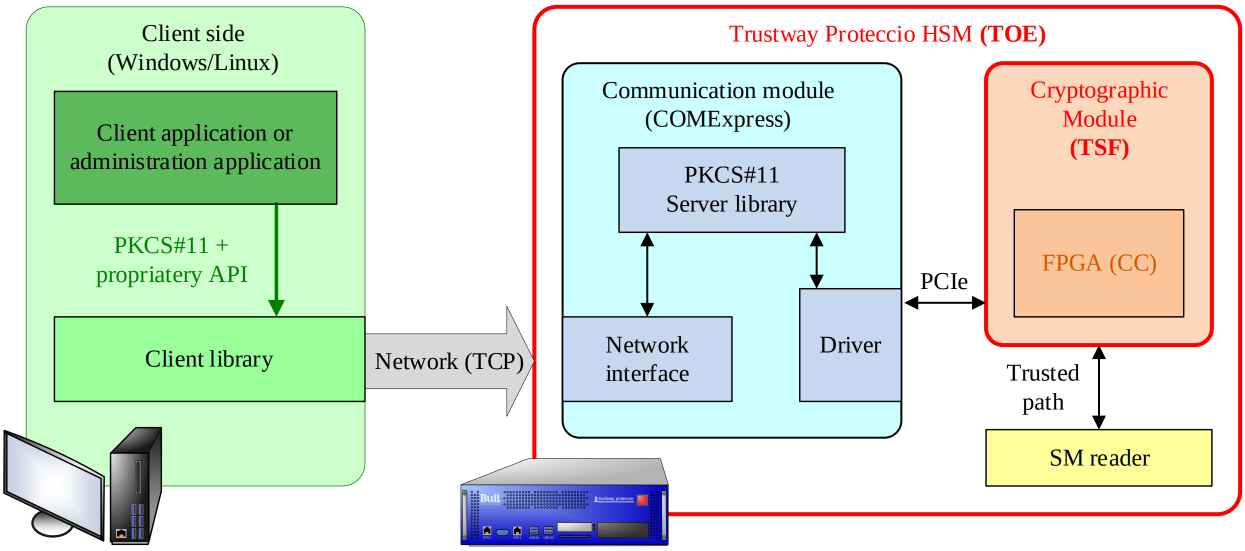

Client side
(Windows/Linux)
Client application or administration application
PKCS#11 + propriatery API
Client library
Network (TCP)
Trustway Proteccio HSM (TOE)
Communication module (COMExpress)
PKCS#11 Server library
Network interface
Driver
PCIe
Cryptographic Module (TSF)
FPGA (CC)
Trusted path
SM reader

Figure 6 - TOE and environment general overview

3.5 TOE functionalities

This section describes all the TOE functionalities. For external functionalities (user services), the ones covered by this certification (within the TSF) are listed in **bold format**.

3.5.1 Cryptographic operations

The TOE supports PKCS#11 API for the following operations:

- **Signature and verification functions**
- Encryption and decryption functions
- **Digest functions**
- Wrap and unwrap functions
- Key derivation functions

- **Key management functions (generation, storage, save/restore, destruction).**

The TOE implements specific PKCS#11 functions, such as C_CreateObject, allowing the introduction of secret, public and private keys.

3.5.2 Algorithmes cryptographiques

Bull HSM is intended to be used as a general purpose cryptographic resource implementing a set of cryptographic algorithms. Some are provided as external cryptographic services to end-users, others are only used as internal mechanisms.

3.5.2.1 Cryptographic services (external)

The TOE implements the following cryptographic algorithms :

- Symmetric encryption/decryption :
  - AES, DES, 2DES, 3DES, modes ECB et CBC,
  - AES-GCM ;
- Asymmetric encryption/decryption:
  - **RSA** (RSA-PKCS, **RSA-PKCS-PSS**, RSA-PKCS-OAEP) ;
- Signature/Verification :
  - **RSA**, MD5-RSA, SHA1-RSA, **SHA256-RSA, SHA384-RSA, SHA512-RSA**,
  - **ECDSA**, **ECDSA-SHA256, ECDSA-SHA384, ECDSA-SHA512**;
- Message authentication/Verification :
  - HMAC MD5, HMAC SHA-1, HMAC SHA256, HMAC SHA384, HMAC SHA512,
  - DES MAC, DES3 MAC,
  - AES MAC, AES-CMAC, AES-GMAC ;
- Digest :
  - **SHA256, SHA384, SHA512**, SHA-1, MD5 ;
- Key derivation :
  - Dedicated mechanism for SNMP dialogue with the TDM,
  - Through AES and DES encryption mode ECB and CBC.
  - ECDH
- Dedicated mechanisms for TLS negotiation:
  - Master key derivation
  - Session keys derivation

3.5.2.2 Internal cryptographic mechanisms

The TOE implements the following algorithms for strict internal usage:

- EC-KCDSA : for software signature verification at start-up, secure software update, software signature and cryptographic configuration signature;

- Key derivation (following ANSI X9.31 standard) with SHA256 for the hash algorithm;
- Retail MAC ISO/IEC 9797-1 compliant with algorithm 3 and filling 2 (integrity protection for secure channel) ;
- Intel (Altera) native AES-256 bitstream encryption ;
- ECDH for MCS-FPGA secure channel key exchange;
- AES mode CTR decryption for executable code during runtime.

3.5.3 Key sizes supported by the TOE

The TOE supports the following key sizes:

- DES : 64 bits
- DES2 : 128 bits
- TDES : 168 bits
- AES : 128, 192, 256 bits
- Generic Secret : 32 à 512 bits
- **RSA** : 512 to 4096 bits key-pairs (step 128) – **2048 to 4096 in PP compliance mode**
- **ECDSA** : 192 to 521 bits **– 256 to 521 bits in PP compliance mode**
- EC-KCDSA : 256 bits

Note :
The virtual HSM cryptographic configuration allows for further restriction in terms of algorithms and key length. For instance, in the PG083 compliant configuration, the RSA minimum key size is 1024 bits and ECC minimum key size is 192 bits.

3.5.4 Key management

Bull HSM provides a high level of key management and storage.

Key generation is performed by a hardware based random number generator generating a physical seed followed by a software post-treatment compliant with NIST SP800-90.

The cryptographic keys are managed in black (bus, memory) in the HSM. They are managed in red only within the FPGA. No key is stored in plaintext in the external memories linked to the FPGA.

The destruction of the keys complies with FIPS140-2, Section 4.7.6, Key zeroisation.

Note :
When a virtual HSM is in the *PP 419221-5* compliant cryptographic configuration, the secret keys **R.SecretKey** (symmetrical and private) must be associated with a secret key user (key owner) through authentication data, in order to be used for cryptographic operations.

3.5.5 Roles

In its operational environment, the TOE handles directly or indirectly the user categories (roles) described below.

Authentication of **Security Officer**, **Auditor**, **Crypto-officer** and **HSM Auditor** is performed on a trusted path (serial connection with smart card reader) using a smart card.

Authentication of **Crypto-user** is performed with a password. The login operation is executed by means of C_Login PKCS#11 function.

Authentication of secret key users (key owners) is performed via password:

- When generating the secret key in the HSM (generation or derivation), the authentication data (password) is provided as an additional (and mandatory) parameter of the associated PKCS#11 request, by the client application. During the transfer to the TOE, this authentication data Is protected by the secure channel between the client application and the TSF.
- In order to perform a cryptographic operation with a secret key, the key owner must authenticate himself using a dedicated request, allowing him to provide the authentication data (password) via the client application. This authentication data Is protected by the secure channel between the client application and the TSF.

Note :

In the rest of the document, the term **administrator** will be used to for the security officer, the auditor, the crypto-officer and the HSM auditor.
Similarly, the term **auditor** will be used for the auditor and the HSM auditor.

Note :
The TOE possesses several configurations with clearly identified roles with their own privileges.
Using the TOE in the "PP compliant" configuration shall ensure the conformity to the SFRs of the Protection Profile.

3.5.5.1 Security Officer (S.Admin – SO Master)

The Security Officer is authorised to execute the following functions:

- Create its own smart card for further authentication;
- Create virtual HSMs;
  - Create the security officer for each virtual HSM.
  - Create the auditor for each virtual HSM.
- Suppress a non-personalized virtual HSM;
- Update the HSM embedded software;
- Update the system software;
- Introduce the software license keys (virtual HSM, …);
- Modify the network configuration.

3.5.5.2 Auditor (S.Admin – Audit Master)

The Auditor is authorised to execute the following operations:

- Create its own smart card for further authentication;
- Read general audit data (events log and security log) generated by the TOE and exported for audit review in the TOE environment;
- Generate and export the log integrity key;
- Read the PKCS#11log file;
- Get the token status

3.5.5.3 Virtual HSM security officer (S.Admin – SO)

A virtual HSM must have one and only one virtual HSM security officer role.

The SO is authorised to execute the following operations:

- Personalize the virtual HSM;
- Depersonalize the virtual HSM ;
- Create the PKCS#11 user for the virtual HSM;
- Choose the user password;
- Configure the start mode for the virtual HSM;
- Individual activation/deactivation of all supported algorithms;
- When in "PP compliance mode":
    - Assign secret keys;
    - Reset authentication data for non-assigned keys;
    - Unblock secret keys.

3.5.5.4 HSM Auditor (S.Admin - Audit)

A virtual HSM must have one and only one user in the HSM Auditor role.

The HSM Auditor is authorised to execute the following operations:

- Read audit data generated by the virtual HSM and exported for audit review in the TOE environment.

3.5.5.5 PKCS #11 user (S.User)

A virtual HSM must have one and only one pkcs#11 user.

The user is not a physical entity, it can be represented by one or more client application upon knowledge of the associated password.

The user can access private objects only after authentication by C_Login function.

3.5.5.6 Secret key user (S.User – key owner)

The secret key user is the physical person authorised to use a PP 419211-5 secret key (R.SecretKey). Such a secret key can only have one user (key owner).

The secret key user is in possession of the authentication data associated with the secret key (password). He can therefore perform cryptographic operations with his secret key (generation, destruction, usage and authentication data modification).

The secret key user interacts with the HSM through a client application acting on his behalf.

The secret key user must also be logged as PKCS#11 user in order to access his key.

3.5.5.7 Client application (External client application)

The client application is a PKCS#11 application linked with the Trustway client library in order to interact with the HSM.

This library is in charge of managing both the TLS channel and the secure channel with the TSF.

The client application is in charge of transmitting the authentication data (password) to the HSM on behalf of the secret key users (cf. FMT_SMR.1).

3.5.5.8 Corresponding table for roles between the PP and the HSM

PP Subject	HSM Role
S.Application	Client application
S.User	the secret key users + PKCS#11 user
S.Admin	SO Master, SO, Audit Master, Audit

Table 3-1. Roles correspondence between the PP and the HSM

3.5.6 Administration

Product administration is performed trough a Java application.

The administration application can be used by several role defined above: SO Master, SO, Audit Master, Audit (authenticated via smartcard) PKCS#11 users and secret key users (authenticated by password).

Administration covers:

- Secure embedded software loading process (SO Master);
- Virtual HSM creation by the security officer, using a specific authentication mechanism which reconstructs a shared secret number in sections by reading M out of N eligible smart cards, which will be used when the operations of backup/restore keys will be executed;
- Virtual HSM personalization/depersonalization (crypto-officer);
- Token cryptographic configuration (supported algorithms, cryptographic operations authorised to the Crypto-user, number and length of cryptographic objects);
- Exploration/delete of PKCS#11 objects;
- Secure backup and restore of cryptographic keys;

3.5.7 TOE installation

The installation method selected for the TOE is based on the threshold scheme principle.

The generation of the N smart cards needed to install the virtual HSM must be done prior to personalize the virtual HSM.

Initially the Security Officer configures N and M using an administrative application or a custom client application on the client PC.

☞ **Note :** N and M may be configured with the value 1 (only one installation card).

In a second step the N smart cards are generated, each owner of the N smart cards choosing its PIN.

☞ **Note :** New installation Shamir smart cards, corresponding to new virtual HSMs, can be generated at any time, under the control of the Security Officer

3.5.8 TOE personalization

A virtual HSM must be personalized before its first use. It allows its association to a particular user, by the use of specific secrets.

The virtual HSM depersonalization imposes the Crypto-officer to be authenticated and needs the reintroduction of the secrets generated during the personalization phase to be able to use it.

Once the virtual HSM is personalized, the PP compliance mode can be activated cf. 3.8 Configuration recommendation.

3.5.9 CIK activation

CIK activation mechanism can be configured during the personalization phase. The possible choices are:

- smart card CIK, based on the threshold scheme principle;
- Automatic CIK (allowing the start/restart without operator intervention and without the need of a smart card).

3.5.10 Test of critical functions

3.5.10.1 Black key decryption

Black key decryption is self-verified in normal operation, by implementing the following principle:

- The cryptographic integrity of keys is verified with all its attributes;
- The elements to be decrypted contain sensitive values and a CRC of these values;
- After decryption, the CRC is checked.

3.5.10.2 Periodic tests and fault management

The software security mechanisms involve a set of periodic tests that constantly monitor the proper operation and integrity of the sensitive functions of the card, to wit:

- The AES, DES, 3DES, RSA, MD5 and SHA/SHA256/SHA384/SHA512 cryptographic operations;
- The random number generator.

All these tests are executed during the start phase

3.6 Protection of the network link between applications and TOE

3.6.1 TLS secure channel between client applications and the TOE

The TOE uses version 3.0.9 of OpenSSL library to implement TLS v1.2 protocol aiming to protect the network link between the applications (client and administration applications) and the TOE.

An auto-signed server certificate is generated by Trustway Proteccio™:

- Server certificate generation (ECC key pair on secp521r1 curve);
- Server certificate signature: ECDSA with SHA384 (secp521r1 curve).

The ciphersuite used by TLS is ECDHE-ECDSA-AES256-GCM-SHA384 (secp521r1 curve).

The previous ciphersuite (DHE-RSA-AES256-SHA256) is still supported for historical compliance, under the control of the Security Officer.

The certificate can be configured by the organization using the TOE services.

3.6.2 Additional secure channel between client applications and the TSF

The TSF implements a secure channel with the client library (and thus the client application), dedicated to protect sensitive information in reference to FTP_TRP.1/External.

A “server” authentication public key is generated and exported by the HSM under SO Master control.

A “client” authentication public key is generated and exported by the client, under its responsibility, and with means at the client discretion.

The authentication public keys are traded between the client and the HSM in order to activate the secure channel.

The cryptographic protocol used for secure channel establishment is NIST C(2e, 2s, ECC CDH) (cf. SP-800 56A) on secp521r1 curve

3.7 TOE delivery

TOE is delivered with the necessary environment to its correct functioning:

- The Bull Trustway appliance, which includes :
    - ○ A custom Linux operating system running on the communication module
    - ○ A custom Linux library, providing the PKCS#11 API and the custom APIs. This library provides all programming and communication interfaces used to interact with the cryptographic module
    - ○ The embedded cryptographic software running on the cryptographic module
- The client library, delivered as a dynamic Windows or Linux library, depending on the system configuration. It runs in the client environment on the client station and provides the programming interface to the client application.
- The Java administration application for the TOE, which runs on the client environment.

The TOE supports access by multiple users. Each user can establish one or more sessions with the HSM. Cryptographic requests received by the communication module and transmitted to the cryptographic module for treatment. Inversely, responses from the cryptographic module are passed to the communication module and transmitted to the client.

In PP 419221-5 compliance mode, the interface is not strictly compliant with the PKCS#11 API. Proprietary requests must be used to correctly manage secret keys (R.secretKey). Only the secret key users are authorised to manipulate secret keys.

The TOE also provides a local interface (through a VGA screen directly connected to the HSM front panel). It allows the local administration of the equipment:

- By the correctly authenticated Security Officer :
    - – Network configuration

  - TLS configuration
  - TSF secure channel configuration
  - Firmware update (cryptographic module)
  - System update (communication module)
  - TLS certificate display
- By the correctly authenticated Auditor :
  - PKCS#11 logs activation and extraction
  - Diagnostics execution
  - Log management activation for PP compliance

3.8 Configuration recommendation

PP compliance requires several usage restrictions in regards to the usual usage of the HSM:

- The TLS channel must be activated with both client and server authentication
- The TSF secure channel must be activated
- The "PP compliant" cryptographic configuration (see below) must be activated on the target virtual HSM.

3.8.1 PP 419221-5 compliant cryptographic configuration

3.8.1.1 Configuration

Virtual HSM configuration recommendation using the Trustway administration application.

Cryptographic Configuration

Options

Maximum number of objects
Only enable ECC and RSA keys**.**

Keys
Only enable ECC and RSA keys

Minimum key sizes
RSA: 2048 bits minimum[1]
ECDSA: 256 bits minimum

Operations

Only the following mechanisms shall be enabled (checked):

Encrypt/Decrypt
Uncheck all mechanisms

Sign/Hash

- CKM_RSA_PKCS_PSS
- CKM_ECDSA

Wrap/Unwrap
Uncheck all mechanisms

Key Generation

- CKM_RSA_PKCS_KEY_PAIR_GEN
- CKM_EC_KEY_PAIR_GEN

Key Creation
Uncheck all mechanisms

Key Derivation

- Uncheck all mechanisms

Digest

- CKM_SHA256
- CKM_SHA384
- CKM_SHA512

Key restoration

[1] la taille minimum des clé RSA est de 2048 bits jusqu'à 2030, à partir de 2031 la taille minimale sera 3072.

- RESTORE_WITH_AUTH_DATA

3.8.1.2 Key attributes restrictions :

Public keys :
CKA_PRIVATE=TRUE

Private keys :
CKA_PRIVATE=TRUE,
CKA_SENSITIVE=TRUE
If the CKA_ASSIGNABLE attribute is set to TRUE:

- CKA_EXTRACTABLE=FALSE
- CKA_MODIFIABLE=FALSE

☞ **Note :**
To be assigned, a key must have been generated with CKA_ASSIGNABLE=TRUE.

3.9 Non-TOE hardware/software/firmware required by the TOE

The TOE is delivered with a client PKCS11 library which shall be installed on the client PC/server. This library is mandatory for the interface between client applications and the TOE.

Chapter 4. Security Problem Definition

This chapter describes the Security Problem Definition of the reference PP [26].

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 (only private 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 shall 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 shall 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 an administrator (hence to control access to privileged administrator functions such as TOE backup, export of audit data) or to authorize a user for access to secret and private keys (R.SecretKey). This asset includes copies of authentication/authorization data made for external storage and/or backup purposes. The integrity of the RAD shall be protected; its confidentiality shall 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 PP 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/authorization 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

4.3.1 General

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 authorized for the relevant action, but who may present themselves as either a completely unknown user, or as one of the subjects in 6.2 (but in this case the attacker will not have access to the authentication or authorization data for the subject).

4.3.2 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.

4.3.3 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.

4.3.4 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 authorization, 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[11].

4.3.5 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[12]), without necessarily obtaining access to the value of the key.

4.3.6 T.KeyOveruse — Overuse of a key

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

4.3.7 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.

4.3.8 T.DataMod — Unauthorised modification of client application data

An attacker modifies data (R.ClientData such as DTBS/R, authentication/ authorization 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.

4.3.9 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 realized. 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

4.4.1 P.Algorithms — Use of approved cryptographic algorithms

The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognized authorities as appropriate for use by TSPs.

Application Note 1

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 910/2014 [10] and an exemplary list of algorithms and parameters is given in ETSI/TS 119 312 [8] or SOG-IS-Crypto [11].

4.4.2 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[2]), shall be implemented in such a way that the secret keys can be reliably protected by the legitimate owner against use by others, and in such a way that the use of the secret keys by the TOE can be confined to a set of authorized cryptographic functions.

Application Note 2

This policy is intended to ensure that the TOE can be used for qualified electronic seals and qualified electronic signatures as in Regulation 910/2014 [10], but recognizes that not all keys are used for such purposes. Therefore, although the TOE needs to 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.

4.4.3 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/authorization data, or seed data for another random number generator that is used for these purposes.

4.4.4 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

The cryptographic module TOE is 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. This is described in the user' manuals.

[2] 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.

4.5 Assumptions

4.5.1 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 shall 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).

4.5.2 A.Env — Protected operating environment

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

4.5.3 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.

4.5.4 A.UAuth — Authentication of application users

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

4.5.5 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

As noted for P.Audit in 4.4.4, the TOE exist as part of a larger system and the System Auditor is a role within this larger system.

4.5.6 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, cryptoperiods and key renewal, and key/certificate revocation.

Chapter 5. Security Objectives

5.1 General

This clause 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.

5.2 Security Objectives for the TOE

5.2.1 General

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

5.2.2 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 authorized users of the keys and administrators of the TOE cannot directly access the plaintext value of a secret key.

5.2.3 OT.Algorithms — Use of approved cryptographic algorithms

The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognized 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

In PP-compliance mode of operations, all cryptographic functions are compliant with ANSSI PG083

See note under P.Algorithms (4.4.1) on relevant references for digital signatures within the European Union.

5.2.4 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 authorization 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.

5.2.5 OT.Auth — Authorization for use of TOE functions and data

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

— administrators of the TOE;

— users of TOE cryptographic functions (client applications using secure channels);

— users of secret keys.

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

Application Note 6

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. However, use of a secret key always requires prior authorization.

5.2.6 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.

5.2.7 OT.KeyUseScope — Defined scope for use of a key after authorization

The TOE is required to define and apply clearly stated limits on when authorization and reauthorization are required in order for a secret key to be used[3]. For example, the TOE may allow secret keys to be used for a specified time period or number of uses after initial authorization, or for may allow the key to be used until authorization is explicitly rescinded. As another example, the TOE may implement a policy that requires re-authorization before every use of a secret key.

Application Note 7

Such limits on the use of a key after initial authorization are termed "re-authorization conditions" in this ST. All types of secret need to be re-authorized after explicit rescinding of previous authorization for access to the secret key .

5.2.8 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/authorization 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

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.

5.2.9 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/authorization data or public key certificates) during transmission between the client application and the TOE.

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

Application Note 9

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.

5.2.10 OT.ImportExport — Secure import and export of keys

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 shall be exported only in encrypted form (it is not sufficient to rely on properties of a secure channel to provide the protection: the key itself shall be encrypted). The TOE also allows individual secret keys under its 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.

5.2.11 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 authorized 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.

5.2.12 OT.RNG — Random number quality

Random numbers generated and provided to client applications for use as keys, authentication/authorization 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.

5.2.13 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.

5.2.14 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.

5.2.15 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.

5.3 Security Objectives for the Operational Environment

5.3.1 General

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.

5.3.2 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).

5.3.3 OE.Env — Protected operating environment

The TOE shall operate in a protected environment that limits physical access to the TOE to authorized 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).

5.3.4 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.

5.3.5 OE.Uauth — Authentication of application users

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

5.3.6 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

As noted for P.Audit in 4.4.4, the TOE exists as part of a larger system and the System Auditor is a role within this larger system.

5.3.7 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, cryptoperiods and key renewal, and key/certificate revocation.

Chapter 6. Extended Components Definitions

The following extended components are defined in the PP [26].

6.1 Generation of random numbers (FCS_RNG)

6.1.1 General

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

6.1.2 Family behaviour

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

6.1.3 Component levelling

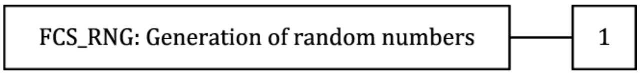


Figure 7 - Generation of Random numbers - Component Levelling

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].

6.2 Basic TSF Self Testing (FPT_TST_EXT.1)

6.2.1 General

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

6.2.2 Family behaviour

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

6.2.3 Component levelling

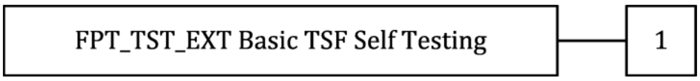


Figure 8 - Basic TSF Self Testing – Component Levelling

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 authorized 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].

Chapter 7. Security Requirements

7.1 General

This clause gives the security functional requirements (SFR) and the security assurance requirements (SAR) for the TOE and the environment.

Security functional requirements components given in 7.4 "*Security Functional Requirements*" are drawn from Common Criteria Part 2 [5]. Some security functional requirements represent extensions to [5]. Operations for assignment, selection and refinement have been made.

The TOE security assurance requirements statements given in 7.5 "*Security Assurance Requirement*" are drawn from the security assurance components from Common Criteria Part 3 [6].

7.2 Typographical Conventions

The following conventions are used in the definitions of the SFRs:

- Refinements made by this ST are marked in **bold** text.
- Selections and assignments made by this ST are underlined.
- Selections, assignments and refinements made by the PP are marked in ***bold italic*** text.
- Iterations are denoted with "/<iteration identifier>"

7.3 SFR Architecture

7.3.1 SFR Relationships

Figure 9 and Figure 10 give a graphical presentation of the connections between the Security Functional Requirements (SFRs) from 7.4 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 7.4 defines the SFRs grouped by the abstract class and family groupings Common Criteria Part 2 [5].

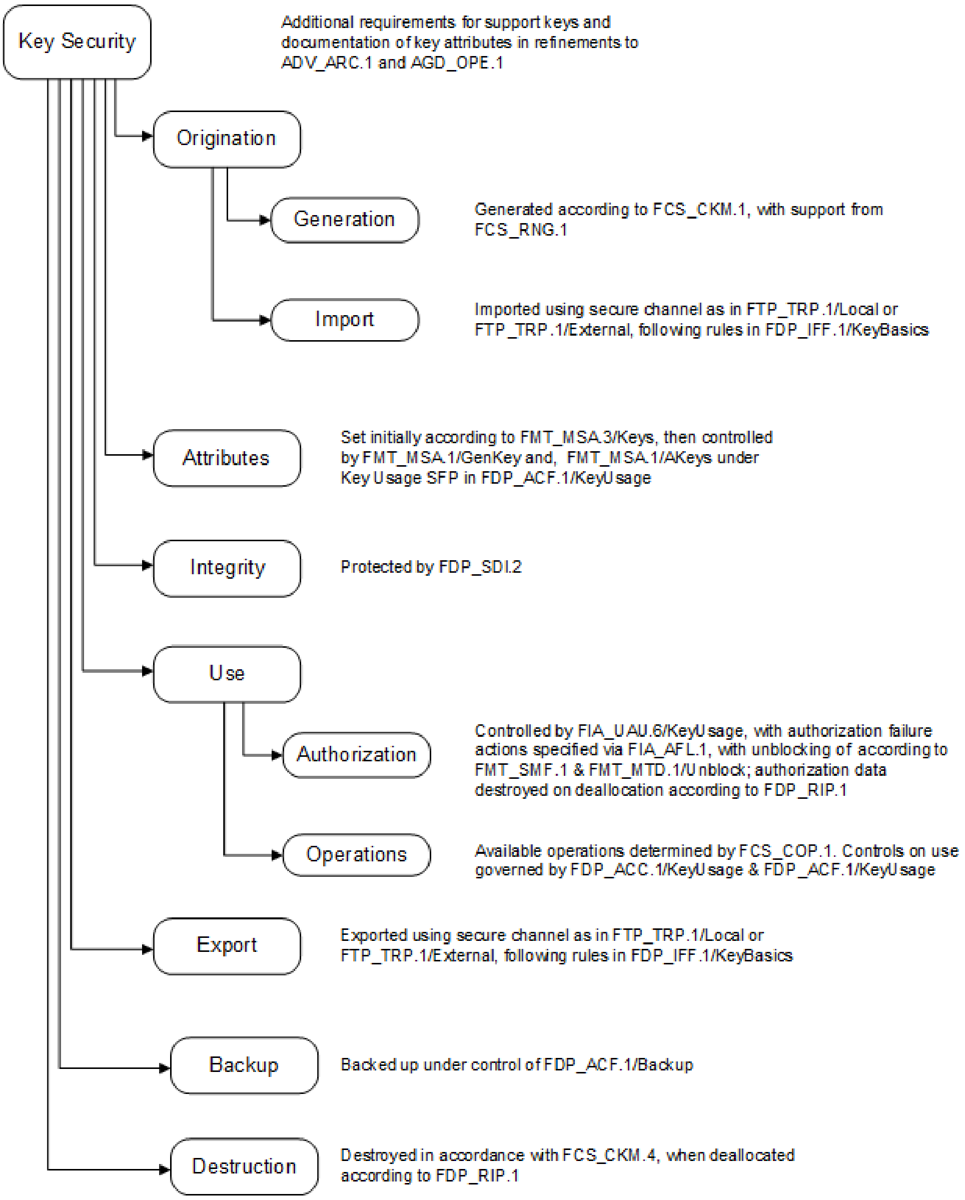
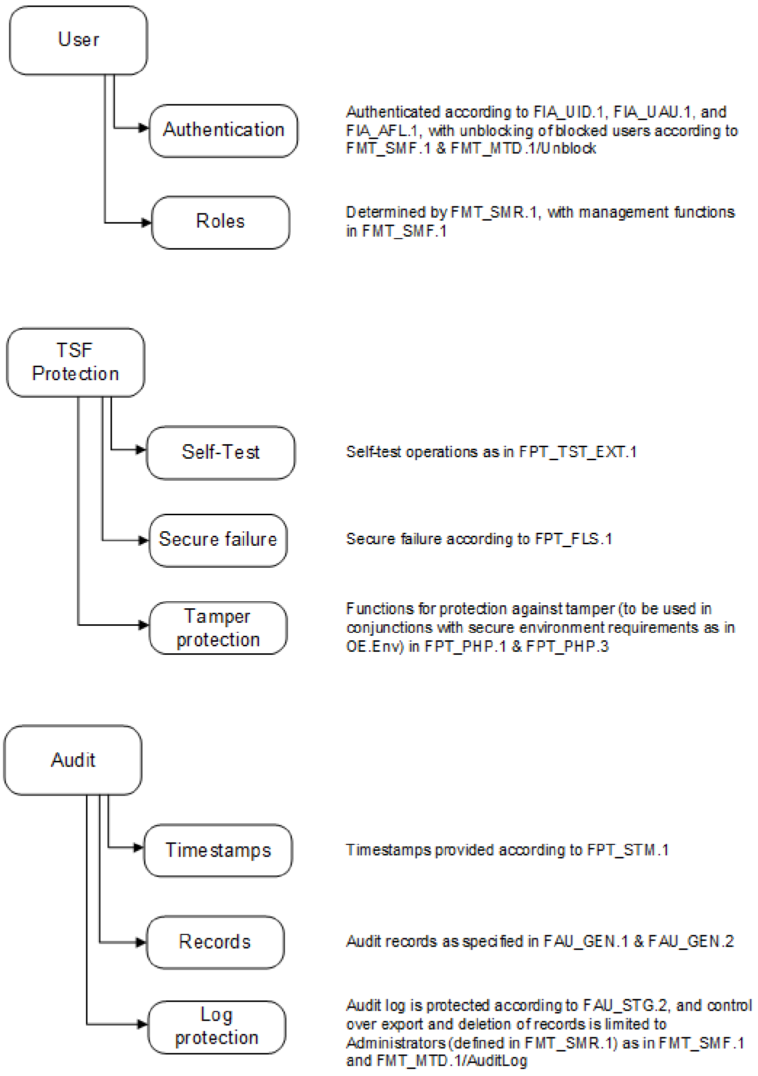

Key Security
Additional requirements for support keys and documentation of key attributes in refinements to ADV_ARC.1 and AGD_OPE.1
Origination
Generation
Generated according to FCS_CKM.1, with support from FCS_RNG.1
Import
Imported using secure channel as in FTP_TRP.1/Local or FTP_TRP.1/External, following rules in FDP_IFF.1/KeyBasics
Attributes
Set initially according to FMT_MSA.3/Keys, then controlled by FMT_MSA.1/GenKey and, FMT_MSA.1/AKeys under Key Usage SFP in FDP_ACF.1/KeyUsage
Integrity
Protected by FDP_SDI.2
Use
Authorization
Controlled by FIA_UAU.6/KeyUsage, with authorization failure actions specified via FIA_AFL.1, with unblocking of according to FMT_SMF.1 & FMT_MTD.1/Unblock; authorization data destroyed on deallocation according to FDP_RIP.1
Operations
Available operations determined by FCS_COP.1. Controls on use governed by FDP_ACC.1/KeyUsage & FDP_ACF.1/KeyUsage
Export
Exported using secure channel as in FTP_TRP.1/Local or FTP_TRP.1/External, following rules in FDP_IFF.1/KeyBasics
Backup
Backed up under control of FDP_ACF.1/Backup
Destruction
Destroyed in accordance with FCS_CKM.4, when deallocated according to FDP_RIP.1

Figure 9 - Architecture of User, TSF Protection and Audit SFRs



Authenticated according to FIA_UID.1, FIA_UAU.1, and FIA_AFL.1, with unblocking of blocked users according to FMT_SMF.1 & FMT_MTD.1/Unblock

Determined by FMT_SMR.1, with management functions in FMT_SMF.1

Self-test operations as in FPT_TST_EXT.1

Secure failure according to FPT_FLS.1

Functions for protection against tamper (to be used in conjunctions with secure environment requirements as in OE.Env) in FPT_PHP.1 & FPT_PHP.3

Timestamps provided according to FPT_STM.1

Audit records as specified in FAU_GEN.1 & FAU_GEN.2

Audit log is protected according to FAU_STG.2, and control over export and deletion of records is limited to Administrators (defined in FMT_SMR.1) as in FMT_SMF.1 and FMT_MTD.1/AuditLog

Figure 10 — Architecture of Key Protection SFRs

7.3.2 SFRs and the Key Lifecycle

The generic lifecycle for a key is illustrated in Figure 11. This 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 then be processed (export, use in a cryptographic function, backup, or destruction). The SFRs related to each of these aspects are then described below Figure 11.

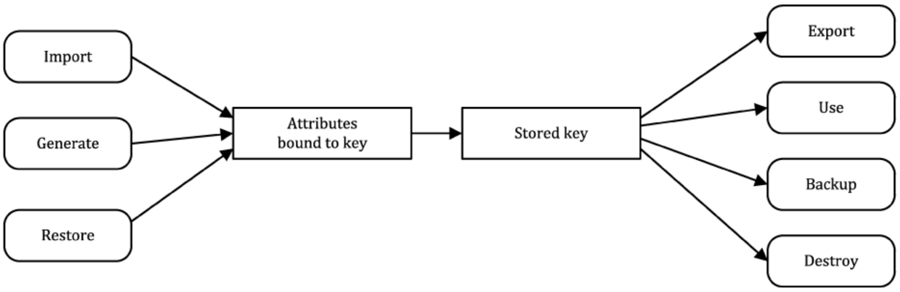


Figure 11 — Generic Key Lifecycle and Related SFRs

Import:

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

Generate:

- FCS_CKM.1 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 audit of generation (and of failure of RNG)

Restore:

- DP_ACF.1/Backup requires only an Administrator can restore from a backup, all backups shall preserve confidentiality and integrity of keys (as appropriate to key type) and their attributes, and any restore shall 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 and FMT_MSA.1/AKeys define requirements on key attribute modification
- FAU_GEN.1 requires audit 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 audit of integrity errors detected

Export:

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

Use:

- FIA_AFL.1 requires blocking of access to a key on reaching an authorization failure threshold (FDP_IFF.1/KeyBasics and FMT_MTD.1/Unblock define requirements on unblocking)
- FDP_ACF.1/KeyUsage requires authorization 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 authorization before initial use of a key and describes any additional requirements for re-authorization conditions such as expiry of a time period or number of uses of a key (or when the authorization period has been explicitly ended)
- FDP_RIP.1 requires protection of authorization data on deallocation
- 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 audit of authorization failure (and blocking or unblocking)

Backup:

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

Destroy:

- FDP_RIP.1 requires key to be protected on deallocation
- FCS_CKM.4 requires key zeroisation on deallocation
- FAU_GEN.1 requires audit of key destruction

7.4 Security Functional Requirements

7.4.1 General

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

7.4.2 Cryptographic Support (FCS)

7.4.2.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.

Dependencies:

[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm: listed in the ‟ Key generation algorithm” column of table FCS_*CKM.1 Cryptographic key generation* and specified cryptographic key sizes: listed in the ‟ Cryptographic key sizes” column of table FCS_*CKM.1 Cryptographic key generation* that meet the following standards: listed in the ‟ Standards” column of table FCS_*CKM.1 Cryptographic key generation.*

Key generation algorithm	Cryptographic key sizes	Standards	Enabled in the TSF (scope of the certification)	SFR iteration
RSA key pairs	1024 to 4096 bits (step 128)	FIPS PUB 186-4 for key sizes of 2048 bits, 3072 bits and 4096 bits)	Yes (for sizes of 2048, 3072 and 4096 bits)	/RSA
AES keys	128, 192, 256 bits	FIPS PUB 197	No	--
AES keys for internal use	256 bits	FIPS PUB 197	Yes	/INTERNAL-AES
ECC key pairs	192 to 521 bits	FIPS PUB 186-3 and ANSI X9.62	Yes	/ECC
Generic Secret keys	32 to 512 bits	PKCS#11 v2.11	No	--
HMAC-SHA256 keys for internal use	256 bits	PKCS#11 v2.11	Yes	/INTERNAL-SHA
Diffie Hellman for internal use	shared key: 2048 bits, private key: 2048 bits, public key: 2048 bits	FIPS PUB 186-3	No	--
EC-KCDSA key pairs	256 bits	"The Korean Certificate-based Digital Signature Algorithm" (1998)	Yes	/INTERNAL-ECC

Table 7-1. FCS_CKM.1 Cryptographic key generation

Application Note 12

Key generation is linked to the setting of security attributes of a key (including the link to a subject who owns the key, via the setting of authorization data) as in FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys.

All keys generated by the TSF for external use (by key owners) are compliant with PG083. Note that, for RSA keys, the minimum key size is 2048 until end of year 2030. From 2031, the minimum key size will be 3072.

In Table "FCS_CKM.1 Cryptographic key generation", key generation mechanisms without an SFR iteration are not in the scope of the certification.

For RSA keys, only 2048 bits, 3072 bits and 4096 bits are in the scope of the certification.

7.4.2.2 FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies:

[FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **zeroisation**[4] that meets the following: actions listed in table *Cryptographic key destruction* :

key destruction method	cryptographic keys	conditions
FIPS 140-2 Level 3	R.SecretKey	When the HSM is operational
Reset of internal FPGA RAM	R.SecretKey present in the internal FPGA RAM	During an HSM alarm (including tamper detection)
Overwriting memory with '0's	Encrypted user keys are erased by overwriting the corresponding memory areas.	When the HSM is operational
Zeroisation of the encrypting key	R.SecretKey encrypted by a support key (or a chain of keys)	Upon detection of specific tampers (opening the box,…)

Table 7-2. Cryptographic key destruction

Application Note 13

Secure destruction methods are described above for all secret keys.

7.4.2.3 FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

Dependencies:

[FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1

[4] [assignment: *cryptographic key destruction method*]

The TSF shall perform a list of cryptographic operations described in the "Cryptographic operation" column of Table 7.3 : *Cryptographic operations* within the TSF (scope of the certification) in accordance with a specified cryptographic algorithm described in the "Algorithm" column of Table 7.3 and cryptographic key sizes described in the "Key Size" column of Table 7.3 that meet the following: a list of standards described in the "Standards" column of Table 7.3.

Cryptographic operation	Algorithm	Key Size (in bits)	Standards	Enabled in the TSF (scope of the certification)	SFR iteration
Digital signature generation and verification	*RSA PSS*	*2048 to 4096*	*PKCS #1 PSS*	*Yes*	*/SIGN-VERIFY*
	ECDSA	*256 to 521*	*FIPS PUB 186-4 and ANSI X9.62*	*Yes*	
	EC-KCDSA	*256 to 521*	*"The Korean Certificate-based Digital Signature Algorithm" (1998)*	*Yes (for internal use)*	*/INTERNAL-SIGN-VERIFY*
SMC secure messaging – message	*AES CBC* *AES CMAC*	*128*	*ISO 9797-1 (MAC algorithm 3, padding mode 2) with AES-CMAC*	*Yes (for internal use)*	*/INTERNAL-SMC*
	SHA256 (key derivation)	*256*	*X9.31*	*Yes*	
Message digest	*SHA256*	*N/A*	*FIPS PUB 180-2*	*Yes*	*/DIGEST*
	SHA384	*N/A*	*FIPS PUB 180-2*	*Yes*	
	SHA512	*N/A*	*FIPS PUB 180-2*	*Yes*	
Key storage (internal or external)	*AES mode CBC (encryption/decryption) and HMAC-SHA256 (integrity)*	*Resp. 256 and 256*	*Encryption/ decryption : FIPS PUB 197* *Integrity : FIPS PUB 198*	*Yes*	*/KEY-STORAGE*
Client secure channel	*ECDH (key agreement) and AES-256-GCM (SC confidentiality and integrity)*	*Resp. 521 and 256*	*Key agreement : FIPS SP800-56A and 56C* *SC confidentiality and integrity : FIPS SP800-38D*	*Yes*	*/SEC-CHANNEL*
Software update	*AES-CBC (confidentiality), ECKCDSA with SHA256 (digital signature verification)*	*Resp. 256 and 256*	*FIPS PUB 197, "The Korean Certificate-based Digital Signature Algorithm" (1998)*	*Yes*	*/SW-UPDATE*
Internal integrity protection	*HMAC-SHA256*	*256*	*FIPS PUB 198*	*Yes (for internal use)*	*/INTERNAL-INTG*

Table 7-3. Cryptographic operations within the TSF (scope of the certification)

Application Note 14

The above Table: "Cryptographic operations within the TSF (scope of the certification)" includes all cryptographic functions that are intended to support TSP operations. Corresponding iterations of FCS_COP.1 are described in the "SFR iteration" column of the Table.

All mechanisms included in the TSF are compliant with ANSSI PG083.
Note that, for RSA keys, the minimum key size is 2048 until end of year 2030. From 2031, the minimum key size will be 3072.

The below Table: "Cryptographic operations not in the TSF (out of the scope of the certification)" describe the cryptographic mechanisms that are supported, in addition, by the HSM in standard mode of operations (not in PP-compliance mode). Those mechanisms are disabled, as client-usable services, when in PP-compliance mode.

Cryptographic operation	Algorithm	Key Size (in bits)	Standards	Enabled in the TSF (scope of the certification)
Digital signature generation and verification	*RSA*	*1024 to 4096*	*PKCS #1 v1.5*	*No*
Message authentication generation and verification	*DES MAC, DES MAC-GENERAL*	*64*	*FIPS PUB 113*	*No*
	DES3 MAC, DES3 MAC-GENERAL	*192*	*FIPS PUB 113*	*No*
	AES MAC, AES MAC-GENERAL, AES CMAC, AES CMAC-GENERAL, AES GMAC	*128, 192, 256*	*FIPS PUB 197 and FIPS PUB 113*	*No*
	MD5 HMAC, MD5 HMAC GENERAL	*40 to 192*	*FIPS PUB 198*	*No*
	SHA-1 HMAC, SHA-1 HMAC GENERAL, SHA256 HMAC, SHA256 HMAC GENERAL, SHA384 HMAC, SHA384 HMAC GENERAL, SHA512 HMAC, SHA512 HMAC GENERAL	*40 to 512*	*FIPS PUB 198*	*No*
Asymmetric encryption and decryption	*RSA*	*1024 to 4096*	*PKCS#1 V1.5 and OAEP (PKCS#1 v2.1 2002)*	*No*
Symmetric encryption and decryption	*DES (ECB and CBC mode)*	*64*	*FIPS PUB 46-3*	*No*
	DES3 (ECB and CBC mode)	*192*	*FIPS PUB 46-3*	*No*
	DES3 CBC	*128*	*FIPS PUB 46-3 REV01 26/11/2001 and ANSSI cryptographic referential*	*No*
	AES (ECB, CBC and GCM mode)	*128,192, 256*	*FIPS PUB 197 and FIPS SP800-38D*	*No*
Symmetric decryption	*AES (CTR mode)*	*256*	*FIPS PUB 197*	*No*
Message digest	*MD5*	*N/A*	*RFC 1321*	*No*

	SHA1	*N/A*	*FIPS PUB 180-2*	*No*
Secret keys wrapping and unwrapping	*RSA*	*1024 to 4096*	*PKCS#1 v1.5 and OAEP (PKCS#1 v2.1 2002)*	*No*
Private keys wrapping and unwrapping	*AES (mode CBC and CBC-PAD)*	*128,192, 256*	*FIPS PUB 197*	*No*
Key Derivation (Trustway VPN SNMP dialogue).	*SHA256*	*256*	*RFC 5996 – Internet Key Exchange Protocol Version 2 (IKEv2)*	*No*
Elliptic Curve Diffie Hellman	*ECDH*	*192 to 521*	*ANSI X9-63-2001/RFC5903*	*No*

Table 7-4. Cryptographic operations not in the TSF (out of the scope of the certification)

7.4.2.4 FCS_RNG.1/HYBRID Generation of random numbers

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1/HYBRID

The TSF shall provide a hybrid deterministic, random number generator that implements PG083-compliant security capabilities.

FCS_RNG.1.2/HYBRID

The TSF shall provide octets of bits that meet PG083 quality metrics

Application Note 15

The RNG is used for all key generation mechanisms described in FCS_CKM.1for both internal and external usage.

The RNG is used for all other random number generation in the HSM, in particular during ECC signature (FCS_COP.1/SIGN-VERIFY, FCS_COP.1/INTERNAL-SIGN-VERIFY)

7.4.3 Identification and authentication (FIA)

7.4.3.1 FIA_UID.1

7.4.3.1.1 FIA_UID.1/S.USER-S.ADMIN Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1/S.USER-S.ADMIN

The TSF shall allow

(1) Self-test according to FPT_TST_EXT.1

(2) The following list of additional TSF-mediated actions on behalf of the user **(Refinement: S.User or S.Admin)** to be performed before the user is identified:

- detection of the secure blocking state (FPT_FLS.1),
- detection of violation of physical integrity (FPT_PHP.2),
- identification (FIA_UID.1)
- query of the HSM status
- query of the virtual HSMs configuration
- generation, usage, modification or destruction of public PKCS#11 objects (with CKA_PRIVATE attribute set to False)

FIA_UID.1.2/S.USER-S.ADMIN

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

7.4.3.1.2 FIA_UID.1/KEY-OWNER Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1/KEY-OWNER

The TSF shall allow

(1) Self-test according to FPT_TST_EXT.1

(2) The following list of additional TSF-mediated actions on behalf of the user **(Refinement: Key Owner)** to be performed before the user is identified:

- All user-allowed operations except:
  - Generation, usage of a **R.SecretKey** (including assigned keys)
  - Modification of the authorization data of a **R.SecretKey** (including assigned keys)

FIA_UID.1.2/KEY-OWNER

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

The expected behaviour of the TSF is different depending on the user. The SFR FIA_UID.1 is therefore refined in two instances:

- FIA.UID.1 has been iterated and refined FIA_UID.1/S.USER-S.ADMIN is a refinement of the SFR in the case of users that are not key owners.
- FIA_UID.1/KEY_OWNER is a refinement of the SFR in the case of key owners.

7.4.3.2 FIA_UAU.1

7.4.3.2.1 FIA_UAU.1/S.USER-S.ADMIN Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1/S.USER-S.ADMIN

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) The following list of additional TSF-mediated actions on behalf of the user **(Refinement: S.User or S.Admin)** to be performed before the user is authenticated.

- detection of the secure blocking state (FPT_FLS.1),
- detection of violation of physical integrity (FPT_PHP.2),
- identification (FIA_UID.1)
- query of the HSM status
- query of the virtual HSMs configuration
- generation, usage, modification or destruction of public PKCS#11 objects (with CKA_PRIVATE attribute set to False)

FIA_UAU.1.2/S.USER-S.ADMIN

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

7.4.3.2.2 FIA_UAU.1/KEY-OWNER Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FIA_UAU.1.1/KEY-OWNER

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) The following list of additional TSF-mediated actions on behalf of the user **(Refinement: Key Owner)** to be performed before the user is authenticated.

- All user-allowed operations except:
  - Generation, usage of a **R.SecretKey** (including assigned keys)
  - Modification of the authorization data of a **R.SecretKey** (including assigned keys)

FIA_UAU.1.2/KEY-OWNER

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

The expected behaviour of the TSF is different depending on the user. The SFR FIA_UAU.1 is therefore refined in two instances:

- FIA.UID.1 has been iterated and refined FIA_UAU.1/S.USER-S.ADMIN is a refinement of the SFR in the case of users that are not key owners.
- FIA_UAU.1/KEY_OWNER is a refinement of the SFR in the case of key owners.

Identification and authentication are done simultaneously which explains the similarity of the TSF-mediated actions between FIA_UID.1.1 and FIA_UAU.1.1.

7.4.3.3 FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1

The TSF shall detect when <u>the number indicated in the "Number of consecutive unsuccessful authentication" column of Table *Authentication failure handling*</u> of unsuccessful authentication ***or authorization*** attempts occur related to ***consecutive failed authentication or authorization attempts***.

FIA_AFL.1.2

When the defined number of unsuccessful authentication ***or authorization*** attempts has been <u>met</u>, the TSF shall ***block access*** to <u>the functionality described in the "Functionality blocked" column of table *Authentication failure handling*</u> ***until*** unblocked by <u>the condition indicated in the "Unblocking condition" column of table *Authentication failure handling*</u>.

Role performing authentication	Number of consecutive unsuccessful authentication or authorization attempts (cf. FIA_AFL.1.1)	Functionality blocked (cf. FIA_AFL.1.2)	Unblocking condition (cf. FIA_AFL.1.2)	SFR iteration
Administrator (S.Admin)	5	All action related to the specific role as described in 3.5.5 Roles	N/A	/ADMIN-CAP
Key Owner (S.User)	5	Access to the associated **R.SecretKey** (usage, modification of authorization data)	The authenticated SO of the virtual HSM can reset the authorization data if the key is not an assigned key The authenticated SO of the virtual HSM can unblock the key by resetting the number of failed authentication attempts	/KEY-OWNER

Table 7-5. *Authentication failure handling*

Application Note 18

The unblocking of functionality blocked as described in each iteration of FIA_AFL.1.2 is described in a corresponding iteration of FMT_MTD.1 (cf. 9.4.7).

For Administrator authentication(FIA_AFL.1.2/ ADMIN-CAP), the number of authentication failures handling (5) applies to each authentication smart card. The Administrator (Master Security Officer, Master Auditor, HSM Security Officer, HSM Auditor) can generate any number of smart cards. If all the attempts with all the smart cards are unsuccessful, the identity will be blocked for authentication

7.4.3.4 FIA_UAU.6/KeyAuth Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1/KeyAuth

The TSF shall ***authorize and re-authorize*** the user ***for access to a secret key*** under the conditions

1) ***Authorization in order to be granted initial access to the key; And***
2) Re-authorization of **R.SecretKey (including assigned keys)** under the following conditions:
    - after explicit rescinding of previous authorization for access to the secret key **(using a specific API described in the developer's guide)**

- Authorization on every subsequent access to the key **(including assigned keys) until the client application, which has gained authorization to use the key, is terminated.**

Application Note 19

Note that any use of a key requires an initial authorization by presentation of the correct authorization data.

The TOE allows unlimited uses of a secret key after initial authorization, until :

- Invoking the dedicated API that allows the key owner to explicitly rescind the authorization to access his key (via the client application).
- The client application used by the key owner terminates. This induces the rescinding of previous authorization.

It is the responsibility of the client application to make appropriate use of any 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 shall 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 authorization for use of a key could allow the creation of more than one signature using the authorized key). Note that in order to make qualified electronic signatures under Regulation (EU) 910/2014 [7] then the user/application shall be able to precisely control the signatures that can be made under each authorization.

Actions taken by the TOE in the case of successive authorization failures are specified in FIA_AFL.1.

7.4.4 User data protection (FDP)

7.4.4.1 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*** on

1) ***subjects: all***
2) ***information: keys***

3) ***operations: all***

7.4.4.2 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*** 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.*** **This assignment is trivially met as key import/export is not supported by the TSF**
4) ***The value of the Export Flag of a key.***

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***
6) ***Unblocking access to a key shall not allow any subject other than those authorized to access the key at the time when it was blocked***

Application Note 20

FDP_IFF.1.2/KeyBasics 1) to 5) are trivially met as key import/export is not supported by the TSF.

Unblocking a key as in FDP_IFF.1.2/KeyBasics (6) restores the ability of subjects to authorize for access to a key by presenting the correct authorization data. As noted for FMT_MTD.1/Unblock, the subject who unblocks the key will not be able to use the key as a result of the unblocking (unless of course they are able to supply the correct authorization data). This is a part of ensuring that sole control of secret keys can be achieved.

FDP_IFF.1.3/KeyBasics

The TSF shall enforce ***the following additional information flow control rules: <u>none</u>***.

FDP_IFF.1.4/KeyBasics

The TSF shall explicitly authorise an information flow based on the following rules:

1) **<u>External storage of secret keys under continuing TOE control shall only be allowed provided that the secret key value is encrypted and linked to its attributes with integrity protection of both</u>**
2) **<u>Only specific support keys shall be used to perform the encryption and integrity protection of secret keys for external storage</u>**

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 authorization 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***

Application Note 21

The TOE does not provide facilities to import or export keys, the relevant part of the SFR is therefore trivially satisfied.

FDP_IFF.1.4/KeyBasics is refined as the TOE provides secure external storage for secret keys.

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. 4.4.1.3).

The Operational Guidance specifies how any attributes not supplied with an imported key are set when the key is imported (or alternatively how such keys are rejected).

7.4.4.3 FDP_ACC.1/KeyUsage Subset access control

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*** on :

1) ***subjects: all***
2) ***objects: keys***
3) ***operations: all***

7.4.4.4 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*** to objects based on the following:

1) ***whether the subject is currently authorized to use the secret key***
2) ***whether the subject is currently authorized to change the attributes of the secret key***
3) ***the cryptographic function that is attempting to use the secret key***

Application Note 22

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

Whether a subject is currently authorized to change the attributes of a secret key is determined by the iterations of FMT_MSA.1 in 7.4.7.

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 authorized subject, and only as permitted in the Key Attributes Modification Table***

2) ***Only subjects with current authorization 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***

Application Note 23

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 authorized either by presenting the correct authorization data for the key as part of the request for the operation or else the authorization has previously been presented by the subject and the current use of the key does not yet require re-authorization according to FIA_UAU.6/KeyAuth (meaning that the current usage is therefore within the usage constraints since the last authorization 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***.

FDP_ACF.1.4/KeyUsage

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: ***none***

Application Note 24

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. 2.4.1.3)

7.4.4.5 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*** on:

1) ***Subjects: all***
2) ***Objects: keys***
3) ***Operations: backup, restore***.

Application Note 25

The TOE does not provide backup and restore of the TSF therefore FDP_ACC.1/Backup and FDP_ACF.1/Backup are trivially met.

The TOE provides external storage mechanism for all user keys. Externally stored key are protected in confidentiality and integrity with secure binding of all its attributes to the key. The external storage mechanism is supported by FCS_COP.1/KEY-STORAGE.

7.4.4.6 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

FDP_ACF.1.1/Backup

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

1) ***whether the subject is an administrator*** .

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 authorized 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.***

FDP_ACF.1.3/Backup

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

FDP_ACF.1.4/Backup

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

Application Note 26

The TOE does not provide backup and restore of the TSF therefore FDP_ACC.1/Backup and FDP_ACF.1/Backup are trivially met.
The TOE provides external storage mechanism for all user keys. Externally stored key are protected in confidentiality and integrity with secure binding of all its attributes to the key. The external storage mechanism is supported by FCS_COP.1/KEY-STORAGE.

7.4.4.7 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*** on all ***keys (including security attributes)***, based on the following attributes: ***integrity protection data***.

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.***

Application Note 27

This SFR is supported by FCS_COP.1/ *KEY-STORAGE* with SHA256 algorithm.

7.4.4.8 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 ***deallocation of the resource from*** the following objects:

- ***authorization data***
- ***secret keys***

Application Note 28

Authorization data is not stored persistently in the TOE. This data is held a minimal time before deallocation according to FDP_RIP.1.

7.4.5 Trusted path/channels (FTP)

7.4.5.1 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 client applications*** that is logically distinct from other communication paths and provides ensured ***authentication*** of its end points and protection of the communicated data from ***modification and disclosure***.

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

FTP_TRP.1.3/Local The TSF shall require the use of the trusted path for all communication between the local client application and the TSF.

Application Note 29

The TOE and local client applications are located within the physical boundary of the same hardware appliance. Indeed, for OEM appliances, local applications run on the communication module that is connected to the TSF via an internal PCIe bus. The trusted path is therefore mapped to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment).

7.4.5.2 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 external client applications*** that is logically distinct from other communication paths and provides ensured ***authentication*** of its end points and protection of the communicated data from ***modification and disclosure***.

FTP_TRP.1.2/External

The TSF shall permit remote external client applications to initiate communication via the trusted path.

FTP_TRP.1.3/External

The TSF shall require the use of the trusted path between remote external client applications and the TSF with:

1) Authentication, confidentiality and integrity protection for confidentiality-requiring TSF sensitive data within cryptographic and security-related services:

- All authentication data (for key owners and PKCS#11 user)

- Plaintext data in encryption services

1) Authentication and integrity protection for integrity-requiring TSF sensitive data within cryptographic and security-related services:

- Attribute templates in key creation, derivation or modification services

- Data in signing services

- Ciphered data in encryption services

Return code in signature verification services

Application Note 30

This SFR is supported by:

- FCS_COP.1/*SEC-CHANNEL*

The term "remote external client applications" refers to any application that uses the Trustway Proteccio™ PKCS#11 client library, runs on a remote target (PC, server…) and communicates with the HSM via a network.

7.4.6 Protection of the TSF (FPT)

7.4.6.1 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

The TOE provides timestamps suitable for supporting the time in an audit record for FAU_GEN.1.

7.4.6.2 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 startup (or power-on) and*** periodically during normal operation to demonstrate the correct operation of the TSF:

- ***At initial start-up (or power-on)***:
  - ***Software/firmware integrity test* for the entire TOE**
    - **First, the TSF performs signature verification on its own software/firmware (cryptographic module)**
    - **Second, the TSF performs signature verification on the software of the communication module (Linux)**
  - ***Cryptographic algorithm tests***
  - ***Random number generator tests***
- Periodically during normal operation :
  - Cryptographic algorithm tests
  - random number generator tests

Application Note 32

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

Software/firmware signature verification is performed by FCS_COP.1/ *INTERNAL-SIGN-VERIF.*

7.4.6.3 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.

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

Physical design of the TOE allows the TOE administrator to verify the physical integrity of the TOE as part of a routine inspection procedure (physical seal, and other physical security measures described in the operational guidance).

7.4.6.4 FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1

The TSF shall resist the physical tampering scenarios described in Table: *Resistance to physical attack* to the list of TSF devices/elements described in Table: *Resistance to physical attack* by responding automatically such that the SFRs are always enforced.

Physical tampering scenarios	TSF devices/elements	Response
Opening the TOE cover to access the TSF	TSF	HSM alarm (depersonalisation of the TOE)
Modification of the temperature	TSF	HSM alarm (reboot of the TOE)
Modification of the voltage	TSF	HSM alarm (reboot of the TOE)

Table 7-6. *Resistance to physical attack*

Application Note 34

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 7.7.2 Physical security general requirements and 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 7.5.2.4.)

7.4.6.5 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)***
3) ***Failures of critical TOE hardware components (including the RNG) occur***
4) ***Corruption of TOE software occurs***
5) None.

Application Note 35

The Operational Guidance includes 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.

7.4.7 Security management (FMT)

For the purposes of specifying a minimum set 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 recognized by having their 'Assigned Flag' attribute set to 'assigned'), and general keys (keys that have their 'Assigned Flag' attribute set to 'non-assigned').

Assigned Keys represent a type of key that can be more easily mapped to requirements for sole control 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[5]. In particular, an Administrator cannot avoid the need to provide the current authorization data in order to use such a key, nor can an Administrator change the authorization 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[6].

In the FMT_MSA SFRs specified for keys below, the permitted values of assignments have been restricted to identify a minimum set of attributes that shall be mapped to their implementation in a 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 shall 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;
- authorization 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, FDP_ACF.1/KeyUsage, and FDP_ACF.1/Backup. Authorization data is required only for secret keys;
- re-authorization conditions: the constraints on uses of the key that can be made before reauthorization is required according to FIA_UAU.6/KeyAuth, and which determines whether a subject is currently authorized to use a key as in FDP_ACF.1/KeyUsage. The types of secret key to which re-authorization conditions apply, and the details of the re-authorization conditions for a specific TOE are described in FIA_UAU.6/KeyAuth in 9.4.3;
- key usage: the cryptographic functions that are allowed to use the key as 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 this PP as ‘true’ (meaning export is allowed) and ‘false’ (meaning export is not allowed) but may be mapped to other suitable binary values in TOE implementations;

[5] 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 4.4.1.

[6] 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 9.5.2).

- assigned flag: indicates whether the key has currently been assigned. Once a key has been assigned by an Administrator then its authorization data can only be changed on successful validation of the current authorization data – it cannot be changed or reset by an Administrator – and the reauthorization conditions and key usage attributes cannot be changed; allowed values are referred to in this PP as 'assigned' and 'non-assigned' but may be mapped to other suitable binary values in TOE implementations.

7.4.7.1 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 **(mapped within the TOE to SO Master, SO, Audit Master and Audit)**, *External Client Application,* Key User, Key Owner, and no other roles.

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Application Note 36

The Local Client Application role represents an identifiable subject that communicates locally with the TOE, within the same hardware appliance. The External Client Application role represents an identifiable subject that communicates remotely with the TOE over a secure channel.

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

7.4.7.2 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 authorization 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*** **(mapped in this case to the Audit Master or Audit roles)**
4) No backup and restore functions
5) No key import function
6) No key export function

Application Note 37

Key export/import is not supported by the TSF.

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

7.4.7.3 FMT_MTD.1/Unblock 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

The TSF shall restrict the ability to ***unblock*** the list of TSF data described in Table *Data unblock* to the authorized identified administrative roles described in Table *Data unblock*.

Role performing authentication	Blocked TSF data	Administrative roles	SFR iteration
Administrator (S.Admin)	All action related to the specific role as described in 3.5.5 Roles	--	/ADMIN-CAP
Key Owner (S.User)	Number of failed authentications attempts for assigned and non-assigned keys	SO of the virtual HSM	/KEY-OWNER

Table 7-7. *Data unblock*

Application Note 38

There is a distinction between administrators authorized to unblock a key and users authorized to use the key. When unblocking a secret key, the unblocking process does not allow a subject to use the key other than a subject who is authorized by presentation of the current authorization data.

For Administrator authentication(FMT_MTD.1.1/Unblock/ ADMIN-CAP), the number of authentication failures handling (5) applies to each authentication smart card. The Administrator (Master Security Officer, Master Auditor, HSM Security Officer, HSM Auditor) can generate any number of smart cards. If all the attempts with all the smart cards are unsuccessful, the identity will be blocked for authentication.

7.4.7.4 FMT_MTD.1/AuditLog 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/AuditLog

The TSF shall restrict the ability to control ***export and deletion of*** the ***audit log records*** to the ***Administrator role*** **(mapped to the Audit master and Audit roles in the TSF)**.

Application Note 39

The control of export and deletion of the audit log records helps to ensure their protection against accidental or malicious deletion.

Key export/import is not supported by the TSF.

7.4.7.5 FMT_MSA.1/GenKeys 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

The TSF shall enforce the ***Key Usage SFP*** to restrict the ability to ***modify*** the security attributes listed in the Key Attributes Modification Table to subjects, objects, and operations among subjects and General Keys specified in the Key Attributes Modification Table.

Key Attribute (MSA.1)		Modification conditions	
Key Attribute (as described in FMT)	**Key Attribute (mapping to PKCS#11 atributes)**	**Assigned Key**	**General Key**
Key identifier	CKA_UUID	Non-modifiable	Non-modifiable
Key type	CKA_CLASS CKA_KEY_TYPE	Non-modifiable	Non-modifiable
Authorization data	CKA_AUTH_DATA	Can only be modified by the **key owner** with successful validation of the authorization data	Can be modified by: - the **key owner** with successful validation of the authorization data - the **SO** (Administrator) **of the virtual HSM**
Re-authorization conditions	Not treated as a key attribute	Non-modifiable	**Non-modifiable**
--	**CKA_MODIFIABLE**	**Set to False - Non-modifiable**	**Can only be modified from true to false by the key owner**
Key usage	CKA_WRAP, CKA_UNWRAP, CKA_ENCRYPT, CKA_DECRYPT, CKA_SIGN, CKA_VERIFY, CKA_DERIVE,	Non-modifiable	**Can only be modified by the key owner if CKA_MODIFIABLE is True**
Export flag	CKA_EXTRACTABLE + CKA_SENSITIVE	CKA_EXTRACTABLE is set to False, CKA_SENSITIVE is set to True Both are non-modifiable	**Can only be modified by the key owner if CKA_MODIFIABLE is True**
Assigned flag	CKA_ASSIGNED	Non-modifiable	Can be modified only by Administrator, and only to change from non-assigned to assigned. **I.e. can only be set by the SO from False to True, if CKA_ALWAYS_SENSITIVE is True and CKA_NEVER_EXTRACTABLE is False**

Integrity protection data	Not treated as a key attribute	Cannot be modified by users (maintained automatically by TSF	Cannot be modified by users (maintained automatically by TSF

Table 7-8. *Key Attributes Modification Table*

7.4.7.6 FMT_MSA.1/AKeys 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

The TSF shall enforce the ***Key Usage SFP*** to restrict the ability to ***modify*** the security attributes listed in the Key Attributes Modification Table to subjects, objects, and operations among subjects and Assigned Keys specified in the Key Attributes Modification Table.

Application Note 40

The Key Attributes Modification Table is referenced from FMT_MSA.1/GenKeys, and FMT_MSA.1/AKeys. The required constraints on security attribute modification specified in this ST are shown in Table above.

Authorization Data and Re-authorization conditions are required for secret keys only. Re-authorization conditions include the conditions specified for FIA_UAU.6.1/KeyAuth (matching the assignments and selections made for that SFR in the Security Target).

7.4.7.7 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*** to provide restrictive, default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Keys the TSF shall allow the the authorised identified roles, according to the constraints in the Key Attributes Initialisation Table to specify alternative initial values to override the default values when an object or information is created

Key Attribute (MSA.1)		**Initialisation conditions**	
Key Attribute (as described in FMT)	**Key Attribute (mapping to PKCS#11 atributes)**	**Assigned Key**	**General Key**
Key identifier	CKA_UID	Initialized by the TSF during generation process	Initialized by the TSF during generation process
Key type	CKA_CLASS CKA_KEY_TYPE	Initialized by the Key owner during generation process	Initialized by the Key owner during generation process
Authorization data	CKA_AUTH_DATA	Initialized by the Key owner during generation process	Initialized by the Key owner during generation process
Re-authorization conditions	Not treated as a key attribute	Initialized by the **TSF** during generation process	Initialized by the **TSF** during generation process
--	CKA_MODIFIABLE	False	Initialized by the Key owner during generation process – False by default
Key usage	CKA_WRAP, CKA_UNWRAP, CKA_ENCRYPT, CKA_DECRYPT, CKA_SIGN, CKA_VERIFY, CKA_DERIVE,	Initialized by the Key owner during generation process	Initialized by the Key owner during generation process
Export flag	CKA_EXTRACTABLE + CKA_SENSITIVE	CKA_EXTRACTABLE set to false CKA_SENSITIVE set to True	Initialized by the Key owner during generation process – CKA_EXTRACTABLE is False by default - CKA_SENSITIVE is True by default
Assigned flag	CKA_ASSIGNED	Set to True by the TSF, during the assigning of the key by the SO	Set to False

Integrity protection data	Not treated as a key attribute	Initialised automatically by TSF	Initialised automatically by TSF

Table 7-9. *Key Attributes Initialisation Table*

Application Note 41

Authorization Data and Re-authorization 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.

Re-authorization conditions are intrinsic to the TSF and cannot be modified.

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.

7.4.8 Security audit data generation (FAU)

7.4.8.1 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** level of audit; and

c. ***Startup of the TOE;***

d. ***Shutdown of the TOE***

e. ***Cryptographic key generation (FCS_CKM.1);***

f. ***Cryptographic key destruction (FCS_CKM.4);***

g. ***Failure of the random number generator (FCS_RND.1);***

h. ***Authentication and authorization failure handling (FIA_AFL.1): all unsuccessful authentication or authorization attempts, the reaching of the threshold for the unsuccessful authentication or authorization attempts and the blocking actions taken;***

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);***

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);***

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

s. None**.**

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.

Application Note 42

The Operational Guidance describes the management of audit functions. Default logging actions of the TOE are also described in Operational Guidance.

7.4.8.2 FAU_GEN.2 User identity association

Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation
	FIA_UID.1 Timing of identification

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.

7.4.8.3 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 detect unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.2.3

The TSF shall ensure that ***all*** stored audit records will be maintained when the following conditions occur: ***audit storage exhaustion***.

Application Note 43

The Operational Guidance describes the management of audit records, the need for an external audit server, the situation regarding records held locally (transiently) on the TOE and those held externally, and the way in which audit records are maintained when local audit storage is exhausted. The Operational Guidance describes the protection applicable to all records created by the TOE, and identifies the obligations for the environment in maintaining audit trail protection. This protection is supported by *FCS_COP/INTERNAL-INTG.*

Control over export and deletion of the audit log records is limited to the Administrator role (Audit master for the HSM) as specified in FMT_MTD.1/AuditLog.

Key import/export is not supported by the TSF.

7.5 Security Assurance Requirements

7.5.1 General

The security assurance requirement level is EAL4 augmented with **AVA_VAN.5, ADV_IMP.2, ALC_CMC.5, ALC_DVS.2 and ALC_FLR.3** . 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 Security Target.

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)
	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)
	Implementation representation of the TSF (ADV_IMP.2)
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)
	Production support, acceptance procedures automation (ALC_CMC.5)
	Problem tracking CM coverage (ALC_CMS.4)
	Delivery procedures (ALC_DEL.1)
	Identification of security measures (ALC_DVS.1)
	Identification of security measures (ALC_DVS.2)
	Developer defined life-cycle model (ALC_LCD.1)
	Well-defined development tools (ALC_TAT.1)

	Systematic Flaw Remediation (ALC_FLR.3)
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 7-10.*Security Assurance Requirements*

7.5.2 Refinements of Security Assurance Requirements

The following refinements are made to selected assurance requirements in Table *Security Assurance Requirements*:

7.5.2.1 ADV_ARC.1 Security architecture description

Refinement:

The following specific topics shall be addressed as part of ADV_ARC.1 for this ST. 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 or client applications, but are used by the TOE to carry out its 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, authorization, 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 shall 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 shall demonstrate that sufficient entropy has been used in the generation of each support key, and the source of that entropy. The rationale shall 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 shall 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 shall 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 and FTP_TRP.1/External)

   e. All local and external interfaces used for communications with users, client applications, audit data, and stored TOE data (cf. 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). This also includes identifying the types of keys (if any) that support re-authorization 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), and any privileges available to the user type/role

h. All of the cryptographic functions provided (cf. 4.4.1.2) and whether any non-endorsed cryptographic algorithms and/or cryptographic functions are available (cf. FCS_COP.1 and 4.4.1.4)

i. The authorization methods used for keys (cf. FIA_UAU.6/KeyAuth and FDP_ACC.1/KeyUsage)

j. Description of the way in which the TOE ensures that it only holds authorization data for the minimum time possible before deallocating 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).

7.5.2.2 AGD_OPE.1 Operational user guidance

Refinement:

The following specific topics shall 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 (EU) 910/2014 [7]. 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). 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 and Annex III] for qualified electronic signatures and qualified electronic seals.

2. The use of trusted channels (cf. FTP_TRP.1/Local and 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, including their use to constrain the period and number of uses that are enabled by authorization of a key (cf. FIA_UAU.6/KeyAuth and Application Note 19).
4. Identification of any non-endorsed cryptographic algorithms and/or cryptographic functions that are available (cf. FCS_COP.1 and 4.4.1.4).
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 and Application Note 32).
7. Specific failures detected by the TOE (cf. FPT_FLS.1 and Application Note 35).
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 and Application Note 42, FAU_STG.2 and Application Note 43, FMT_MTD.1/AuditLog and Application Note 39).
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.

7.5.2.3 ATE_IND.2 Independent testing – sample

Refinement:

The following specific topics shall 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.

7.5.2.4 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 in 7.7.2 Physical security general requirements and 7.7.3 Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3.

7.5.3 Configuration for PP-compliance

In order to set the HSM in the mode of operations compliant with the PP 419221-5, three steps must be taken:

- Enabling the external secure channel between the client applications and the TSF (see section “Management of the secure channel with the cryptographic module”).
- Enabling the secure export of audit event through syslog-ng (see section “Management of secure audit export”).
- Once the two steps above have been taken: enabling the PP-compliant cryptographic configuration on the target virtual HSM (see section “PP-compliant cryptographic configuration”).

Chapter 8. Rationales

8.1 Security Objectives Rationale

8.1.1 Security Objectives Coverage

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

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstrain	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.Algorithms		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 8-1. Security Problem Definition mapping to Security Objectives

8.1.2 Security Objectives Sufficiency

8.1.2.1 General

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

8.1.2.2 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 authorization check before using a secret key. OT.KeyUseConstraint expands on this to set out requirements for the granularity of authorization.

T.KeyOveruse is concerned with the possibility that more uses may be made of an authorized key than were intended, and this is addressed by the requirements of OT.KeyUseScope which requires controls to be specified and enforced for any re-authorization conditions that the TOE allows a user to define.

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.

8.1.2.3 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 well-defined limits on the use of a key when it is authorized (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).

8.1.2.4 Assumptions

Each of the Assumptions in 6.5 is directly matched by a security objective for the operational environment in 7.3. 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

Table *TOE Security Objectives mapping to SFRs* summarizes the mapping of Security Objectives for the TOE to SFRs

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstra	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FCS_CKM.1/RSA		**x**												
FCS_CKM.1/INTERNAL-AES		**x**												

FCS_CKM.1/INTERNAL-SHA		**x**	**x**											
FCS_CKM.1/ECC		**x**												
FCS_CKM.1/INTERNAL-ECC		**x**												
FCS_CKM.4	**x**													
FCS_COP.1/SIGN-VERIFY		**x**												
FCS_COP.1/INTERNAL-SIGN-VERIFY		**x**												
FCS_COP.1/INTERNAL-SMC		**x**		**x**										
FCS_COP.1/DIGEST		**x**												
FCS_COP.1/KEY-STORAGE	**x**	**x**	**x**											
FCS_COP.1/SEC-CHANNEL		**x**		**x**			**x**							
FCS_COP.1/SW-UPDATE		**x**												
FCS_COP.1/INTERNAL-INTG		**x**												**x**
FCS_RNG.1/HYBRID											**x**			
FIA_UID.1/S.USER-S.ADMIN				**x**										
FIA_UID.1/KEY-OWNER				**x**										
FIA_UAU.1/S.USER-S.ADMIN				**x**										
FIA_UAU.1//KEY-OWNER				**x**										
FIA_AFL.1/ADMIN-CAP				**x**										
FIA_AFL.1/KEY-OWNER				**x**										
FIA_UAU.6/KeyAuth				**x**		**x**								
FDP_IFC.1/KeyBasics	**x**				**x**				**x**					
FDP_IFF.1/KeyBasics					**x**									
FDP_ACC.1/KeyUsage					**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**
FMT_SMF.1				**x**										**x**
FMT_MTD.1/Unblock/ADMIN-CAP				**x**										
FMT_MTD.1/Unblock/KEY-OWNER				**x**										
FMT_MTD.1/AuditLog														**x**

FMT_MSA.1/GenKeys					x									
FMT_MSA.1/AKeys					x									
FMT_MSA.3/Keys					x									
FAU_GEN.1														x
FAU_GEN.2														x
FAU_STG.2														x

Table 8-2. Security Requirements Coverage

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. FCS_COP.1/KEY-STORAGE ensures the confidentiality of keys during storage (internal and external). FCS_COP.1/KEY-STORAGE relies on FCS_CKM.1/INTERNAL-AES for key generation.

OT.Algorithms is addressed by the need to use endorsed standards for FCS_COP.1 (all iterations) and the use of an appropriate random number generator in FCS_CKM.1 (all iterations). Note that the refinements to assurance components in 7.5.2 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 integrityprotected 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 under FDP_IFF.1/KeyBasics). FCS_COP.1/KEY-STORAGE ensures the integrity and linking of key attributes to its value during storage (internal and external). FCS_COP.1/KEY-STORAGE relies on FCS_CKM.1/INTERNAL-SHA for key generation.

OT.Auth is addressed by FIA_UID.1, FIA_UAU.1 and FIA_AFL.1 for administrator authentication (with FMT_MTD.1/Unblock and its dependencies on FMT_SMR.1 and FMT_SMF.1 ensuring that appropriate roles and unblocking for authorization and authentication failures are also provided). Authorization for external client applications is provided by the requirements for authentication of end points in FTP_TRP.1/Local and FTP_TRP.1/External. Authorization for the use of secret keys is addressed by FIA_UAU.6/KeyAuth. Administrator authentication is supported by FCS_COP.1/INTERNAL-SMC, FTP_TRP.1/External is supported by FCS_COP.1/SEC-CHANNEL.

OT.KeyUseConstraint is addressed by the requirements for well-defined (and securely initialised) key attributes in FMT_MSA.1/GenKeys, FMT_MSA.1/AKeys, 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 authorization 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 re-authorization conditions for use of a secret key specified in 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, and by 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. Backup is not implemented in the TOE.

OT.RNG is addressed by the requirement in FCS_RNG.1/HYBRID 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.

OT.FailureDetect 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 Table *Dependencies Rationale*. 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

Requirement	Dependencies	Fulfilled by
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1/SIGN-VERIFY FCS_COP.1/ INTERNAL-SIGN-VERIFY FCS_COP.1/INTERNAL-SMC FCS_COP.1/DIGEST FCS_COP.1/KEY-STORAGE FCS_COP.1/SEC-CHANNEL FCS_COP.1/SW-UPDATE FCS_COP.1/INTERNAL-INTG FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1/RSA FCS_CKM.1/INTERNAL-AES FCS_CKM.1/ECC FCS_CKM.1/INTERNAL-SHA FCS_CKM.1/INTERNAL-ECC See also note below on key attributes during import or export.
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1/RSA FCS_CKM.1/INTERNAL-AES FCS_CKM.1/ECC FCS_CKM.1/INTERNAL-SHA FCS_CKM.1/INTERNAL-ECC FCS_CKM.4 See also note below on key attributes during import or export.
FCS_RNG.1	No dependencies	
FIA_UID.1	No dependencies	
FIA_UAU.1	FIA_UID.1	FIA_UID.1/S.USER-S.ADMIN FIA_UID.1/KEY-OWNER
FIA_AFL.1/ADMIN-CAP	FIA_UAU.1	FIA_UAU.1/S.USER-S.ADMIN
FIA_AFL.1/KEY-OWNER	FIA_UAU.1	FIA_UAU.1/KEY-OWNER
FIA_UAU.6/KeyAuth	No dependencies	
FDP_IFC.1/KeyBasics	FDP_IFF.1	FDP_IFF.1/KeyBasics
FDP_IFF.1/KeyBasics	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/KeyBasics FMT_MSA.3/Keys
FDP_ACC.1/KeyUsage	FDP_ACF.1	FDP_ACF.1/KeyUsage
FDP_ACF.1/KeyUsage	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/KeyUsage FMT_MSA.3/Keys
FDP_ACC.1/Backup	FDP_ACF.1	FDP_ACF.1/Backup

FDP_ACF.1/Backup	FDP_ACC.1 FMT_MSA.3	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.
FDP_SDI.2	No dependencies	
FDP_RIP.1	No dependencies	
FTP_TRP.1/Local	No dependencies	
FTP_TRP.1/External	No dependencies	
FPT_STM.1	No dependencies	
FPT_TST_EXT.1	No dependencies	
FPT_FLS.1	No dependencies	
FMT_SMR.1	FIA_UID.1	FIA_UID.1/S.USER-S.ADMIN FIA_UID.1/KEY-OWNER
FMT_MTD.1/Unblock	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1/AuditLog	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/GenKeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/AKeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
FMT_MSA.3/Keys	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/GenKeys, FMT_MSA.1/AKeys FMT_SMR.1
FAU_GEN.1	FPT_STM.1	FPT_STM.1

FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1 FIA_UID.1/S.USER-S.ADMIN FIA_UID.1/KEY-OWNER
FAU_STG.2	FAU_GEN.1	FAU_GEN.1 FCS_COP.1/INTERNAL-INTG
FMT_MTD.1/Unblock	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1

Table 8-3. *SFR Dependencies*

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 7.5.2 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 (for FMT_MSA.3) also requires that the initialisation of any attributes on import is described in the Security Target

Note :

The TOE does not allow import/export of keys.

8.2.3 Rationale for SARs

The assurance level for this Security Target is EAL4 augmented with **AVA_VAN.5, ADV_IMP.2, ALC_CMC.5, ALC_DVS.2, ALC_FLR.3**.

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialized 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 this Security Target is just such a product. Augmentation results from the selection of **ADV_IMP.2, ALC_CMC.5, ALC_DVS.2, ALC_FLR.3, AVA_VAN.5**. All the dependencies are satisfied by other assurance components in the EAL4 assurance package

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.

8.2.3.1 AVA_VAN.5 Advanced methodical vulnerability analysis

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.

8.2.3.2 ADV_IMP.2 Complete mapping of the implementation representation of the TSF

Regarding the implementation representation of the TSF, the assurance level is augmented to ADV_IMP.2 to comply with the ANSSI Enhance Qualification requirements.

8.2.3.3 ALC_CMC.5 Advanced Support

Regarding the support for the TOE, the assurance level is augmented to ALC_CMC.5 as it is a dependency for ADV_IMP.2.

8.2.3.4 ALC_DVS.2 Sufficiency of security measures

Regarding the security measures, the assurance level is augmented to ALC_DVS.2 to comply with the ANSSI Enhance Qualification requirements.

8.2.3.5 ALC_FLR.3 Systematic Flaw Remediation

Regarding the flaw remediation management, the assurance level is augmented to ALC_FLR.3 to comply with the ANSSI Enhance Qualification requirements.

END OF DOCUMENT