

# EVIDEN

rue Jean Jaurès  
78340 Les Clayes-sous-Bois  
FRANCE

# Trustway

## Trustway Proteccio®

## SECURITY TARGET LITE

**Document reference:** PCA4\_0152

**Date :** July 29, 2024

**Version :** 1.3



## Revision

Revision	Author	Date	Reason
1.0	BULL SAS	October 5th 2023	Document creation
1.1	BULL SAS	November 6th 2023	Format corrections
1.2	BULL SAS	May 31 <sup>th</sup> 2024	Format corrections
1.3	BULL SAS	July 29 <sup>th</sup> 2024	Format corrections

Table 1-1. Revisions

# Summary

<b>Revision.....</b>	<b>2</b>
<b>Summary.....</b>	<b>3</b>
<b>Chapter 1. Introduction .....</b>	<b>6</b>
1.1 Introduction .....	6
1.2 References.....	6
1.3 Glossary .....	8
<b>Chapter 2. ST introduction .....</b>	<b>10</b>
2.1 ST identification.....	10
2.2 ST overview .....	10
2.3 CC conformance .....	12
2.4 PP conformance.....	12
2.5 Qualification conformance .....	12
<b>Chapter 3. TOE description.....</b>	<b>13</b>
3.1 Product type.....	13
3.2 Architecture .....	15
3.3 Life cycle .....	17
3.4 TOE boundary .....	19
3.5 TOE functionalities .....	19
3.5.1 Cryptographic operations .....	19
3.5.2 Cryptographic algorithms .....	19
3.5.3 Key sizes supported by the TOE .....	20
3.5.4 Key management.....	20
3.5.5 Roles.....	20
3.5.6 TOE roles .....	20
3.5.7 Administration .....	22
3.5.8 TOE installation.....	22
3.5.9 TOE personalization.....	23
3.5.10 CIK activation.....	23
3.5.11 Test of critical functions .....	23
3.6 Protection of the network link between applications and TOE.....	24
3.7 TOE usage .....	24
<b>Chapter 4. TOE Security Environment .....</b>	<b>26</b>
4.1 Assets to protect.....	26
4.1.1 TOE services.....	26
4.1.2 TOE internal data .....	26
4.1.3 Data shared between the TOE and its environment .....	27
4.2 Threats.....	28
4.3 Organisational Security Policies .....	32

4.4	Assumptions .....	32
<b>Chapter 5.</b>	<b>Security Objectives .....</b>	<b>34</b>
5.1	Security Objectives for the TOE .....	34
5.2	Security Objectives for the Environment .....	38
<b>Chapter 6.</b>	<b>Security Requirements .....</b>	<b>40</b>
6.1	Security Functional Requirements .....	40
6.1.1	Security audit (FAU) .....	40
6.1.2	Cryptographic support (FCS) .....	42
6.1.3	User data protection (FDP) .....	47
6.1.4	Identification and authentication (FIA) .....	51
6.1.5	Security management (FMT) .....	52
6.1.6	Privacy (FPR) .....	54
6.1.7	Protection of the TOE Security Functions (FPT) .....	55
6.1.8	Trusted path (FTP) .....	58
6.2	TOE Security Assurance Requirements .....	59
<b>Chapter 7.</b>	<b>TOE summary specification .....</b>	<b>60</b>
7.1	TOE Security functions .....	60
7.1.1	Audit Data Generation (SF.AUDIT) .....	60
7.1.2	Authentication (SF.AUTHENTICATION) .....	60
7.1.3	Access control (SF.ACCESS_CONTROL) .....	62
7.1.4	HSM management .....	62
7.1.5	Cryptographic operations (SF.CO) .....	62
7.1.6	Secure loading (SF.SL) .....	63
7.1.7	Security mechanisms (SF.SM) .....	64
7.1.8	Backup and Recovery (SF.BACKUP) .....	65
<b>Chapter 8.</b>	<b>PP claims .....</b>	<b>67</b>
8.1	Reference PP .....	67
8.2	PP addition .....	67
8.2.1	Threats .....	67
8.2.2	Assumptions .....	67
8.2.3	Security objectives .....	67
8.2.4	Security objectives for the environment .....	67
8.2.5	Security Functional Requirements .....	67
<b>Chapter 9.</b>	<b>Rationale .....</b>	<b>69</b>
9.1	Introduction .....	69
9.2	Security Objectives Rationale .....	69
9.2.1	Security Objectives Coverage .....	69
9.2.2	Security Objectives Sufficiency .....	72
9.3	Security Requirements Rationale .....	79
9.3.1	Security Requirement Coverage .....	79
9.3.2	Security Requirements Sufficiency .....	80
9.4	TOE Summary Specification Rationale .....	85

9.4.1	TOE Security functions Coverage .....	85
9.4.2	TOE Security functions Sufficiency.....	87
9.5	Dependency Rationale .....	92
9.5.1	Functional and Assurance Requirements Dependencies .....	92
9.5.2	Justification of unsupported Dependencies .....	96
9.6	Rationale for Assurance Level 4 Augmented .....	97
9.6.1	AVA_VAN.5 Advanced methodical vulnerability analysis .....	97
9.6.2	ADV_IMP.2 Complete mapping of the implementation representation of the TSF .....	97
9.6.3	ALC_DVS.2 Sufficiency of security measures .....	98
9.6.4	ALC_FLR.3 Systematic Flaw Remediation .....	98

# 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, 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, 1<sup>er</sup> 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 d'authentification** Version 1.0, 13 Janvier 2010.
4. **Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model**; Version 3.1, Revision 5, April 2017, CCMB-2017-04-001
5. **Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements**; Version 3.1, Revision 5, April 2017, CCMB-2017-04-002
6. **Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements**; Version 3.1, Revision 5, April 2017, CCMB-2017-04-003
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), en ce qui concerne certaines des exigences de sécurité.
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**  
Référence : 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. **86F276FH - TrustWay Proteccio – Installation\_and\_user\_guide**  
septembre 2023
25. **86F275FH – TrustWay Proteccio – Developer’s Guide\_ATOS**  
June 2023

## 1.3 Glossary

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



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

**Table 1-1. Acronyms**

## Chapter 2. ST introduction

### 2.1 ST identification

Title: **BULL Trustway HSM – Security Target Lite**

Author: BULL SAS

TOE versions:

- **Trustway Proteccio EL/HR/XR:**
  - CDROM : 3.06.xx (the minor “xx” version of the CDROM is out of the scope of this ST)
  - System version: X170
  - Security Module version : V167
- **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.**

### 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 is based on the [Protection Profile prTS 419221-2:2015](#) (Cryptographic Module for CSP Signing Operations with Backup – Protection Profile).

The ST is also based on the Protection Profile [prTS 419221-3:2015](#) (Cryptographic Module for CSP key generation services), for some of the security requirements.

## 2.5 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 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.

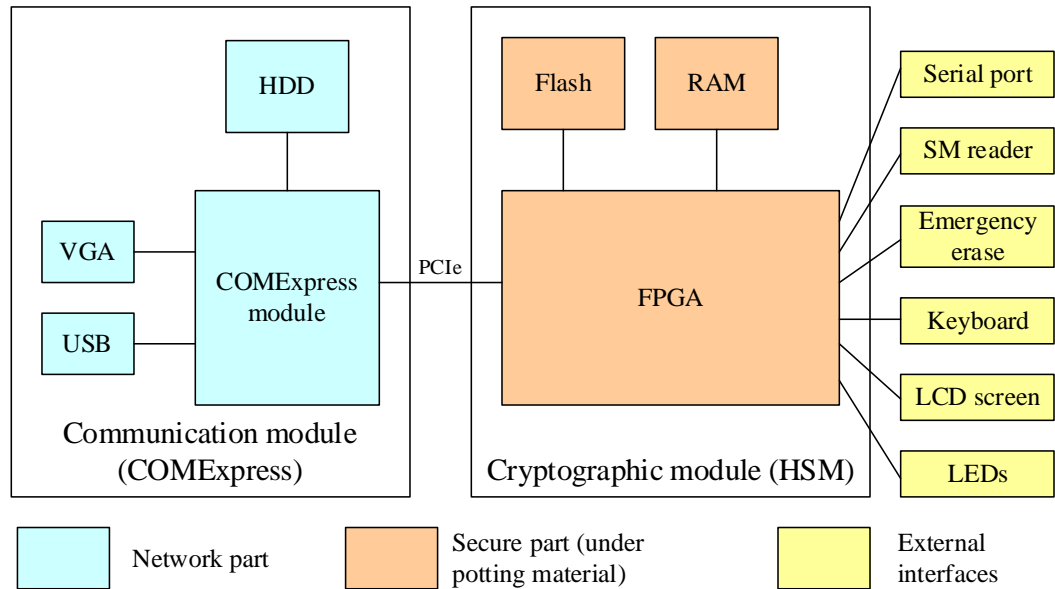
The protection of secret elements is provided by a CIK mechanism at power on. The CIK activation mode can be configured during the personalization phase. Two modes are possible: smart card CIK – automatic CIK (for a start/restart without operator intervention and without the need of a smart card).

The HSM can be partitioned in virtual HSM while guaranteeing the strong compartmentalization of cryptographic key structures.

The 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
  - DDR2/4 RAM
  - NAND FLASH to store the black keys
- A 2.5-inch SATA hard drive connected to the ComExpress module
- A 3.5V lithium battery
- An ATX AC/DC power supply of at least 100 W
- 2 or 3 fans.

The external interfaces are:

**Front:**

- 2 RJ45 Ethernet Interfaces, of which only one is active (eth0)
- 1 VGA Interface
- 4 USB 2.0 interfaces (keyboard, mouse, external drive...)
- 1 smart card reader
- 1 LCD display

- 1 16-key keyboard
- 1 emergency stop button
- 1 status two-colour LED (Ready/Error/Alarm)
- 1 battery weak-status LED



**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 7 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 into the development environment (under the responsibility of the developer)
Phase 2	Manufacturing	The HSM manufacturing process is performed by subcontractors: <ul style="list-style-type: none"> <li>• Manufacture of printed circuit motherboard (SOMACIS),</li> <li>• Enclosure manufacturing (ATOS),</li> <li>• Assembly, programming (manufacturing specific version) and test of components and integration into the enclosure (ASTEEL),</li> <li>• Repeat the tests into the enclosure (ASTEEL).</li> </ul>	Production
Phase 3	Pre-personalization	This phase comprises 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.	These phases are executed in BULL site (under the responsibility of Bull BILS)
Phase 4	Delivery to the client	The Trustway Proteccio is sent to the client.	
Phase 5	Personalization	Personalization by the client.	These phases are executed into the end user environment (under the responsibility of the end user)
Phase 6	Embedded software update	If needed, the end user (security officer) can update the HSM embedded software.	
Phase 7	TOE use	This last phase is executed by the end user.	

The evaluation perimeter circumscribes to phases 1 to 4, 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 to be re-personalized (phase 3) 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.

## 3.5 TOE functionalities

### 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 Cryptographic algorithms

Bull HSM is intended to be used as a general purpose cryptographic resource implementing a set of cryptographic algorithms:

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-SHA1 ;
- 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

### 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, 256 bits
- Generic Secret : 32 à 512 bits
- RSA : 512 to 4096 bits key-pairs (step 128)
- ECDSA : 192 à 521 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 into the FPGA.

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

### 3.5.5 Roles

### 3.5.6 TOE roles

The TOE supports 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.

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.

### 3.5.6.1 Security Officer (SO)

The Security Officer (SO) 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.

### 3.5.6.2 Auditor (Master Auditor)

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;
- Read the PKCS#11log file;
- Get the token status.

### 3.5.6.3 Crypto-officer (HSM Security Officer)

A virtual HSM must have one and only one user in the Crypto-officer role.

The Crypto-officer is authorised to execute the following operations:

- Personalize the virtual HSM;
- Depersonalize the virtual HSM ;
- Create the Crypto-user for the virtual HSM;
- Choose the user password;
- Configure the start mode for the virtual HSM;
- Individual activation/deactivation of all supported algorithms;
- Modification of cryptographic configuration parameters.

### 3.5.6.4 HSM Auditor

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.
- Suppress the audit data for the virtual HSM

### 3.5.6.5 Crypto-user (HSM user)

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

The Crypto-user can access private objects only after authentication by C\_Login function.

The Crypto-user is authorised to perform cryptographic operations.

## 3.5.7 Administration

Product administration is performed by a Java application.

Administration covers:

- Secure embedded software loading process;
- 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;
- Consultation/export of audit records (auditor or HSM auditor).

## 3.5.8 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

### 3.5.9 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.

### 3.5.10 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).

### 3.5.11 Test of critical functions

#### 3.5.11.1 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.

#### 3.5.11.2 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

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).

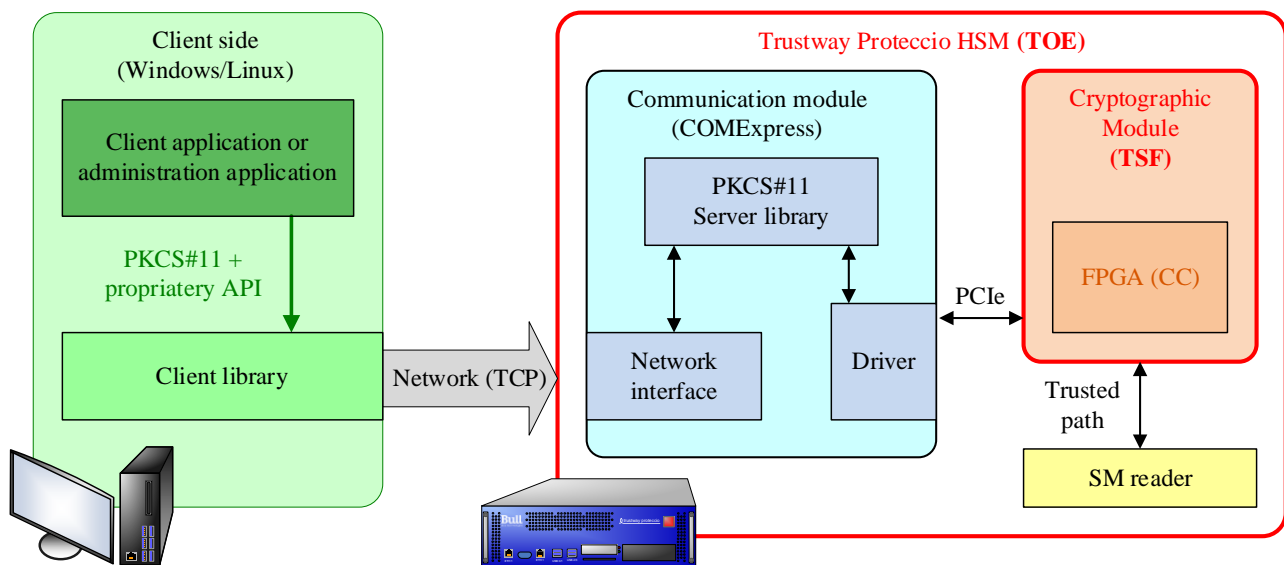
The ciphersuite used by TLS is ECDHE-ECDSA-AES256-GCM-SHA384 (secp521r1 curve).

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.

## 3.7 TOE usage

The TOE is responsible for protecting the CSP\_SCD and other cryptographic keys against disclosure, compromise and unauthorised modification and for ensuring that the TOE services are only used in an authorised way.



**Figure 6 - TOE and environment general overview**

As shown in Figure 6, end-users will communicate with the client application, which in turn will call TOE services on behalf of the end-user, via the client library. The client application is responsible for passing any user data in a correct way to the TOE. Different mechanisms may be used to protect the user data on its way from the originating user to the TOE, but all those mechanisms are not part of the TOE functionality and therefore not defined in this Security Target

The TOE provides authentication, access control and audit for users of its services.



The client application in the TOE environment is the intermediary between the TOSE cryptographic services and the end-users. It is therefore the responsibility of the client application to identify, authenticate and control access of its end-users gaining access to the TOE services provided for the Crypto-user role

The TOE provides an appropriate interface and communication path (called trusted path) between human users and the TOE for authentication and management services. The trusted path transmits identification, authentication and management data of TOE roles in a secure way to the TOE.

The TOE supports backup and restoration of cryptographic keys, with the TSF data needed to re-establish an operational state after recovery from a failure. Backup and restoration is done using cryptographic protocols and mechanisms that protect the confidentiality of the backup data and detect loss of the integrity of the backup data. Measures must be taken within the non-IT environment to ensure the availability of the backup data.

The TOE is delivered to the customer complete with the most important components of the environment. These environmental components are the following:

- Bull appliance platform including:
  - A Linux operating system with a modified hardened Kernel;
  - A specific driver and the PKCS #11 Cryptographic API software (provided as a Linux server library), allowing the access to the cryptographic module;
  - The embedded cryptographic software;
- The client library (provided as a Windows or Linux dynamic library), running under the client server environment and allowing the client application to call the TOE services on behalf of the end-user (Crypto-user).
- A Java application for administration of the TOE.

The TOE supports access by multiple users. Each user establishes one or more sessions with the cryptographic module, by which requests for services are transmitted to the cryptographic module and responses received.

The TOE offers a local application (available through the front VGA interface) allowing its local administration:

- By the Security Officer, after authentication:
  - Network configuration;
  - TLS configuration;
  - Embedded cryptographic software update;
  - System (Linux) software update.
  - Display of the TLS certificates
  - Syslog configuration
- By the Auditor:
  - Allow PKCS#11 traces;
  - Execute diagnostic test

## Chapter 4. TOE Security Environment

### 4.1 Assets to protect

The primary assets that need to be protected by the TOE are the following:

#### 4.1.1 TOE services

##### **R.SERVICES (I, A)**

TOE services are:

- Generation and management cryptographic keys;
- Usage of cryptographic keys for cryptographic operations;
- Backup and restore of TSF data;
- Secure update of the embedded software that implements the TOE services (also protected in confidentiality);
- Identity and role management;
- Internal Audit.

These services have to be protected in Integrity and Availability.

#### 4.1.2 TOE internal data

##### **R. USER\_DATA (C, I, A)**

Confidential user data (CSP\_SCD, other user related cryptographic keys (private, secret), data objects) must be protected in confidentiality, integrity and availability.

##### **R.USER\_PUB\_KEYS (I)**

Public keys used for signature verification, public keys and certificates used for encryption, must be protected in Integrity.

##### **R.DTBSR\_DS (Au)**

The result of the electronic signature over the R.DTBS\_REPRESENTATION, produced by the cryptographic module, has to be protected in authenticity.

**R.TSF\_DATA (C, I, A)**

TSF data (especially VAD and RAD) and other sensitive system data not related to a user or role (system configuration data, audit data) which have to be protected in confidentiality, integrity and availability.

**R.USERMGMT\_DATA (I)**

Non-confidential user / role related data (identifier, access control lists, role definitions, etc.). Those data have to be protected in integrity.

**R.CODE\_HSM (C, I, Au)**

Software embedded into the HSM, protected in confidentiality, integrity and authenticity

### 4.1.3 Data shared between the TOE and its environment

**R.BACKUP\_DATA (C, I)**

Backup data exported by the TOE to the TOE environment and restored in the TOE (R.USER\_DATA et R.USER\_PUB\_KEYS). This data needs to be protected in integrity and confidentiality by the TOE. Availability of this data has to be ensured in the TOE environment.

**R.BACKUP\_KEY (C, I)**

Cryptographic keys used to ensure the integrity and confidentiality of R.BACKUP\_DATA before exporting it outside the TOE. They must be protected in integrity and confidentiality by the TOE.

**R.DTBS\_REPRESENTATION (I)**

Data to be signed with cryptographic keys must be protected in integrity.

## 4.2 Threats

The expected attackers are qualified so as to have HIGH attack potential, in accordance with the security assurance given by AVA\_VAN.5 "*Advanced methodical vulnerability analysis*". Attackers can either be authorised users (security officer, Auditor, Crypto officer, HSM Auditor, Crypto user) or non-authorised users acting intentionally or by negligence.

The following threads have been added to those stated in the PP:

T.INSECURE\_CHANNEL – Sensitive data from applications sent by a non-authorised user to the TOE could be manipulated during the network transmission.

T.TRUSTED\_PATH – Sensitive authentication data sent by unauthorised personnel to the TOE could be manipulated.

T.KEYS\_DERIVE – Expands the thread T.CSP\_SCD\_DERIVE to include all key material manipulated by the TOE.

T.KEYS\_DISCLOSE – Expands the thread T.CSP\_SCD\_DISCLOSE to include all key material manipulated by the TOE.

T.KEYS\_ALTERATION – Expands the thread T.CSP\_SCD\_ALTERATION to include all key material manipulated by the TOE.

T.MISUSE\_OPERATION – Expands the thread T.MISUSE\_OPERATION to include all the cryptographic operations.

T.CRYPTO\_FORGERY – Expands the thread T.SIGNATURE\_FORGERY to include all the cryptographic operations.

### T.BACKUP

The attacker might manipulate R.BACKUP\_DATA in the TOE environment in order to restore altered R.BACKUP\_DATA that will alter R.USER\_DATA, R.USERMGMT\_DATA or R.TSF\_DATA

*Threatened assets: R.USER\_DATA, R.USERMGMT\_DATA, R.TSF\_DATA*

### T.BACKUP\_KEY\_Alteration

An attacker, or an auditor or crypto officer, might modify or alter R.BACKUP\_KEY by interaction with the TOE logical internal functions, or within the TOE environment, in order to:

- Invalidate the cryptographic checksum of R.BACKUP\_DATA or
- Invalidate decryption of R.BACKUP\_DATA.

An agent such as Crypto-user, auditor or crypto-officer, might also modify or alter R.BACKUP\_KEY in the TOE environment within the session of a Crypto-officer whose responsibility is to load this data in the crypto module.

*Threatened assets: R.BACKUP\_KEY*

### T.BACKUP\_KEY\_Derive

An attacker or an auditor or crypto officer, might derive all or part of R.BACKUP\_KEY using knowledge about the R.BACKUP\_KEY operations (generation, usage, destruction), even during legitimate use of R.SERVICES.

*Threatened assets: R.BACKUP\_KEY*

### T.BACKUP\_KEY\_Disclose

An attacker or an auditor or crypto officer might disclose all or part of R.BACKUP\_KEY over physical or logical TOE interface by bypassing the export control mechanisms.

*Threatened assets: R.BACKUP\_KEY*

### **T.BAD\_SW**

The attacker might try to load malicious software into the TOE in order to modify or gain illicit access to R.USER\_DATA, R.TSF\_DATA, R.USERMGMT\_DATA, R.SERVICES or R.CODE\_HSM.

*Threatened assets: R.USER\_DATA, R.TSF\_DATA, R.USERMGMT\_DATA, R.SERVICES, R.CODE\_HSM*

### **T.KEYS\_ALTERATION**

When the user cryptographic key or other cryptographic keys are distorted, cryptographic operations using these keys are invalid. For example, DTBS signed with the distorted cryptographic key (e.g. qualified certificates or CRLs) will be invalid.

Although the use of a distorted cryptographic key can be detected, the impacts for the organisation issuing the signed data using the cryptographic key (e.g. qualified certificates) can be high.

*Threatened assets: R.USER\_DATA*

### **T.CSP\_SVD\_ALTERATION**

The attacker might alter R.USER\_PUB\_KEYS when R.USER\_PUB\_KEYS is exported from the TOE.

Although the use of a distorted R.USER\_PUB\_KEYS can be detected, the impacts for the organisation issuing the signed data using the CSP\_SCD (e.g. qualified certificates) can be high.

*Threatened assets: R.USER\_PUB\_KEYS*

### **T.KEYS\_DERIVE**

The attacker might derive all or part of R.USER\_DATA using knowledge about the R.USER\_DATA operations (generation, usage and destruction), R.DTBS or R.USER\_PUB\_KEYS, even during legitimate use of R.SERVICES.

*Threatened assets: R.USER\_DATA*

### **T.KEYS\_DISCLOSE**

The attacker might disclose all or part of R.USER\_DATA over physical or logical TOE interface by bypassing the export control mechanisms.

*Threatened assets: R.USER\_DATA, R.TSF\_DATA*

### **T.DATA\_MANIPUL**

The attacker might manipulate R.DTBS\_REPRESENTATION within the TOE environment before it is passed to the TOE. This may result in the effect that the TOE signs data without the approval of the user under whose control the data is submitted to the TOE.

When performed within the client application such manipulations may not be detectable by the TOE itself and therefore this threat needs to be countered within the TOE environment.

Manipulation of data in the TOE environment within the session of a Crypto-officer may also result in a compromise of the security of the TOE. The backup of user data and TSF data might be lost.

*Threatened assets: R. DTBS\_REPRESENTATION*

### **T.INSECURE\_INIT**

The attacker, (e.g. unauthorised personnel, authorised personnel without using adequate organisational controls) may initialise the TOE with insecure R.TSF\_DATA or R.USERMGMT\_DATA.

*Threatened assets: .TSF\_DATA, R.USERMGMT\_DATA*

### **T.MALFUNCTION**

An internal malfunction of TOE functions may result in:

- the modification of R.DTBS\_REPRESENTATION,
- misuse of R.SERVICES,
- disclosure or alteration of R.USER\_DATA
- denial of R.SERVICES for authorised users
- alteration of R.TSF\_DATA or R.USERMGMT\_DATA

This includes the destruction of the TOE as well as hardware failures, which prevent the TOE from performing its services.

This includes also the destruction of the TOE by environmental failure.

Finally, this includes any kind of physical tampering that induces erroneous behaviour from the underlying hardware or software of the TOE.

Technical failure may result in an insecure operational state violating the integrity and availability of the TOE services.

The correct operation of the TOE also depends on the correct operation of critical hardware components. Critical components might be:

- the central processing unit
- a coprocessor for accelerating cryptographic operations
- a physical random number generator
- storage devices used to store the R.USER\_DATA or the DTBS-Representation
- physical I/O device drivers.

*Threatened assets: R.DTBS\_REPRESENTATION, R.SERVICES, R.USER\_DATA, R.TSF\_DATA, R.USERMGMT\_DATA*

### **T.MISUSE\_OF\_TOE**

The attacker (e.g. CSP personnel) may misuse the TOE R.SERVICES to forge R.USER\_DATA, R.USER\_PUB\_KEYS, R.USERMGMT\_DATA or R.TSF\_DATA.

For instance, a user of the client application or the TOE misuses a TOE service, in order to falsify a R.USER\_DATA/R.USER\_PUB\_KEYS pair.

### **T.MISUSE\_OPERATION**

The attacker (user of the client application or user of the TOE) misuses R.SERVICES for cryptographic functions (i.e. signature-creation to sign forged qualified certificates with forged cryptographic keys).

### **T.PHYS\_MANIPUL**

An attacker may try to physically manipulate the TOE with the intent to:

- derive all or part of the cryptographic keys or,
- manipulate the user data (DTBS for instance) within the TOE
- misuse R.SERVICES.
- alter R.TSF\_DATA

The TOE may be physically attacked by even an authorised user of TOE services.

This threat includes also the destruction of the TOE by deliberate action.

### **T.INSECURE\_CHANNEL**

An attacker could manipulate sensitive data from applications sent by an authorised user to the TOE could be manipulated during the network transmission, and thus affect the TOE initialisation or configuration.

### **T.TRUSTED\_PATH**

An attacker could manipulate sensitive authentication data sent by an authorised personnel to the TOE, and thus affect the TOE initialisation or configuration.

### **T.CRYPTO\_FORGERY**

An attacker exploits weaknesses in R.SERVICES in order to forge the output data (e.g. into the cryptography and/or key management in the TOE, in order to forge a R.DTBS\_REPRESENTATION or digital signature in a way that is not detectable by the verifier of the signature).

### **T.KeyGeneration\_Misuse**

A user misuses a R.SERVICES, for instance the key generation service for a signing key pair, which can lead to creating or using unauthorised R.USER\_DATA et R.USER\_PUB\_KEYS

## 4.3 Organisational Security Policies

### P.ALGORITHMS

The referential for “Enhanced” strength level edited by ANSSI ([1], [2], [3]) must be followed for key generation, key management and cryptographic operations.

## 4.4 Assumptions

Some of the assumptions of this ST are either more specific than or in addition to those of the prTS419221-2 PP. The following assumption has been added to the ST for the reasons indicated.

A.ADMIN: Because of the overall complexity of the TOE, the personnel responsible for its administration (installation, configuration, audit review, etc.) must be adequately trained.

A.PROTECTION\_HOST – The operating system is a rugged and secure system that ensures the integrity of sensitive files and allows access only to authorised remote applications.

The assumption A.SECURE\_CHANNEL has been suppressed due to the presence of a trusted path between the user and the TOE for authentication and management operations.

### A.AUDIT\_SUPPORT

The organisation using the TOE services reviews the audit trail generated and exported by the TOE. The client application receives and stores the audit trail of the TOE for review by the System auditor of the organisation (Role Auditor/HSM Auditor) according to the audit procedure of the organisation.

### A.DATA\_STORE

The TOE environment ensures the confidentiality, integrity and availability of their security relevant data for TOE initialisation, start-up and operation if stored or handled outside the TOE. The TOE environment ensures the availability of the backup data. Examples of these data are verification authentication data, cryptographic key material and documentation of TOE configuration data.

### A.CRYPTOUSER\_AGENT

The client-application is assumed as user of the TOE in the Crypto-user role. The administration application can also be assumed as user of the TOE in the Crypto-user or Crypto-Officer role. Other users authorised for the TOE Crypto-user services may be not be known to the TOE itself. The TOE environment performs identification and authentication for these individual users and allows successfully authenticated users to use the client application as their agent for the Crypto-user services.

### A.TRUSTED\_ENVIRONMENT

The HSM operates in a secure environment with policy for trustworthiness of operating personnel and physical security of the environment.

### A.CORRECT\_DTBS



DTBS-representation submitted to the TOE are assumed to be correct. This requires that the DTBS (e.g. the certificate content data) have been initialised correctly and maintains this correctness until it is passed to the TOE. This requires the DTBS to be correctly defined during the registration process, be transferred with integrity protection between the systems involved in the process (e.g. registration and certificate generation), be processed in a correct way by the client application, being hashed correctly (in the case the hashing is done by the client application and not by the TOE) and passed correctly to the TOE.

The TOE environment will probably use its own mechanisms to ensure this correctness during processing and transmission. This will for example include mechanisms that can be used to verify the integrity and authenticity of user data when passed between different entities within the TOE environment.

#### **A.ADMIN**

Authorised administrators are non-hostile, appropriately trained and follow all administrator guidance. In particular, authorised administrators are authenticated before performing any action, through a trusted path based on a smart card authentication procedure.

#### **A.PROTECTION\_HOST**

The operating system is a rugged and secure system that ensures the integrity of sensitive files and allows access only to authorised remote applications.

## Chapter 5. Security Objectives

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

### 5.1 Security Objectives for the TOE

The following objectives have been added or represent refinements of the PP objectives:

O.KEYS\_SECURE – expands the objective O.CSP\_SCD\_Secure to include all key material handled by the TOE.

O.CRYPTO\_SECURE – expands the objective O.Sign\_Secure to include all the cryptographic operations handled by the TOE.

O.SECURE\_LOADING – has been added to counter T.BAD\_SW.

O.TRUSTED\_PATH – has been added to counter T.TRUSTED\_PATH.

O.DEPERSONALIZATION – has been added to provide depersonalization means, allowing the assurance that all the sensitive elements introduced during the personalization phase and the user keys are erased.

#### O.AUDIT

The TOE shall audit the following events:

- TOE initialisation
- TOE start-up
- R.USER\_DATA generation
- R.USER\_DATA destruction
- Unsuccessful authentication (R.TSF\_DATA)
- Modification of TOE management data (R.USERMGMT\_DATA )
- Creation of new users/roles (R.USERMGMT\_DATA)
- Suppression users/roles (R.USERMGMT\_DATA)
- Access and suppression (zeroization) of audit data
- Export and restore of R.BACKUP\_DATA
- Self-test execution at start-up, initialisation, on demand and during maintenance
- Unsuccessful self-test operations (R.TSF\_DATA)
- Unsuccessful restore of backup data
- Unsuccessful restore attempt
- Generation, export, import and destruction of R.BACKUP\_KEYS
- TOE software update
- Tamper detection event
- Creation/suppression of virtual HSMs
- Personalization/depersonalization of virtual HSMs

The integrity of the audit trail shall be ensured. The TOE shall provide the management function for the audit to the Auditor only. The TOE shall export the audit data only upon request the Auditor. When applicable, audit data shall be associated with the identity of the user/role that caused the event.

There are three different audit files:

- An audit file related to the whole equipment Trustway Protecchio, associated to the role Auditor (Master Auditor);
- A security audit file, associated to the role Auditor (Master Auditor);
- An audit file related to each virtual HSM, associated to the role HSM Auditor.

The only possible action for these audit files is the consultation by the respective role.

*Threats countered* T.BACKUP, T.BAD\_SW, T.MALFUNCTION, T.INSECURE\_INIT, T.MISUSE\_OF\_TOE, T.PHYS\_MANIPUL, T.KeyGeneration\_Misuse.

## O. KEYS\_SECURE

The confidentiality and integrity of the user cryptographic keys (R.USER\_DATA) shall be ensured during their whole lifetime.

The TOE shall ensure cryptographic secure user cryptographic keys (R.USER\_DATA) generation, use and management. This includes protection against disclosing completely or partly the R.USER\_DATA and other cryptographic keys through any physical or logical TOE interface.

The TOE implements secure cryptographic algorithms and parameters for the generation of all cryptographic keys (including R.USER\_DATA / R.USER\_PUB\_KEYS pairs) chosen from ANSSI referential for "Enhanced" strength level [2].

*Threats countered* T.KEYS\_DERIVE, T.KEYS\_DISCLOSE, T.KEYS\_ALTERATION, T.MISUSE\_OPERATION

## O.CHECK\_OPERATION

The TOE shall perform regular checks to verify that its components operate correctly. This includes integrity checks and authenticity (when required) of TOE software, firmware, internal TSF data or user data during initial start-up, at installation, maintenance and during exploitation.

*Threats countered* T.BACKUP, T.BAD\_SW, T.KEYS\_ALTERATION, T.MALFUNCTION, T.PHYS\_MANIPUL, T.BACKUP\_KEY\_Manipulation

## O.RBAC

The TOE shall restrict the access to its services, depending on the user role, to those services explicitly assigned to this role. Assignment of services to roles shall be either done by explicit action of a Security-officer or by default. Roles may also be predefined in personalisation phase.

*Threats countered* T.BACKUP, T.BAD\_SW, T.INSECURE\_INIT, T.MISUSE\_OF\_TOE, T.MISUSE\_OPERATION, T.KeyGeneration\_Misuse

## O.ATTACK\_RESPONSE

The TOE shall detect attempts of physical tampering and securely destroy the R.USER\_DATA and other cryptographic keys in this case.

*Threats countered* T.KEYS\_ALTERATION, T.PHYS\_MANIPUL

## O.SECURE\_STATE

The TOE shall enter a secure state whenever it detects a failure or an integrity error of software, firmware, internal TSF data or user data. The secure state shall prevent the loss of confidentiality of the R.USER\_DATA and other cryptographic keys.

*Threats countered* T.BACKUP, T.BAD\_SW, T.KEYS\_ALTERATION, T.PHYS\_MANIPUL

## O.PROTECT\_EXPORTED\_DATA

The TOE shall apply integrity and confidentiality protection measures to all assets listed in the asset list requiring integrity or confidentiality protection when they are exported from the TOE e.g. for the purpose of backup and restore.

The TOE implements secure cryptographic algorithms and parameters for the encryption and data integrity protection chosen from ANSSI referential for "Enhanced" strength level [2].

*Threats countered* T.BACKUP, T.KEYS\_DISCLOSE, T.KEYS\_ALTERATION

## O.CRYPTO\_SECURE

The TOE performs secure cryptographic operations.

In particular, the TOE creates signatures such as the advanced signature in qualified certificates that

- Do not reveal the cryptographic keys and
- Cannot be forged without knowledge of the R.USER\_DATA.

The TOE implements secure cryptographic algorithms for all cryptographic operations (including the signing operation) chosen from ANSSI referential for "Enhanced" strength level [1].

*Threats countered* T.KEYS\_DERIVE, T.KEYS\_DISCLOSE, T.MISUSE\_OPERATION, T.CRYPTO\_FORGERY

## O.USER\_AUTHENTICATION

The TOE shall be able to identify and authenticate the users acting with a defined role, before allowing any access to TOE protected assets. Identification and authentication shall be user-based.

Authentication for the Security-Officer, Crypto-Officer, Auditor and HSM Auditor relies on smartcards. Those authentications are performed via the smartcard reader on the HSM front panel (trusted path).

*Threats countered* T.BACKUP, T.BAD\_SW, T.INSECURE\_INIT, T.MISUSE\_OF\_TOE, T.MISUSE\_OPERATION, T.KeyGeneration\_Misuse

## O.TRUSTED\_PATH

The TOE shall supply a trusted communication path with human users physically independent from application path. This trusted path will ensure that the identification and authentication data of TOE users are transmitted correctly and in a confidential way to the TOE. It is materialized by the smartcard reader on the front panel of the HSM (trusted path).

*Threats countered* T.TRUSTED\_PATH

## O.SECURE\_LOADING

The TOE shall supply a secure loading process to update the TOE embedded software. The loading operation must be performed by applying integrity and confidentiality protection measures to protect from any loading of malicious software. Loading operation shall be audited and performed under Security-Officer control.

*Threats countered* T.BAD\_SW

### **O.DEPERSONALIZATION**

The TOE shall supply a depersonalization process, allowing the assurance that all the sensitive elements introduced during the personalization phase and the user keys are erased.

*Threats countered* T.INSECURE\_INIT, T.MISUSE\_OF\_TOE, T.KEYS\_ALTERATION

### **O.BACKUP\_SECURE**

TOE must generate a cryptographic checksum and protect R.BACKUP\_DATA in confidentiality in a way that does not reveal R.BACKUP\_KEY. Generation of the checksum and the encrypted version of R.BACKUP\_DATA shall not be possible without the knowledge of R.BACKUP\_KEY. The checksum verification and the decryption of the protect R.BACKUP\_DATA shall not be possible without the knowledge of R.BACKUP\_KEY.

*Threats countered* T.BACKUP, T.BACKUP\_KEY\_Manipulation.

### **O.BACKUP\_KEY\_SECURE**

Confidentiality and integrity of R.BACKUP\_KEY must be ensured during its whole life-cycle.

TOE must ensure in a secure way, the generation, usage and management of R.BACKUP\_KEY. This includes its protection against disclosure through physical and logical interfaces of the TOE.

*Threats countered* T.BACKUP\_KEY\_Manipulation.

## 5.2 Security Objectives for the Environment

The following security objectives relate to the TOE environment. This includes the client and administration application as well as the procedures for the secure operation of the TOE.

The following security objectives for the environment have been added or represent refinements of the PP objectives:

O.ENV\_ADMIN – Authorised administrators are non-hostile, appropriately trained and are authenticated through a trusted path based on a smart card authentication procedure before performing any action.

O.ENV\_SECURE\_CHANNEL – Data passing between the applications and the TOE are protected in confidentiality and in integrity.

O.ENV\_PROTECTION\_HOST – The operating system must be a rugged and secure system that ensures the integrity of sensitive files and allows access only to authorised remote applications.

### O.ENV\_APPLICATION

The applications which use the TOE shall perform the necessary security checks on the data passed to the TOE.

Security controls in the TOE environment shall also prevent unauthorised manipulation of data submitted to the TOE.

The applications (client and administration) shall also perform the required user identification and authentication as well as local access control.

### O.ENV\_AUDIT

The environment ensures the availability of the generated and exported by the TOE audit trails and provides a review of the audit trail recorded by the TOE.

### O.ENV\_SECURE\_CHANNEL

The TOE environment shall ensure the confidentiality and integrity of all data transferred between the applications (client applications and administration application) and the TOE.

### O.ENV\_PERSONNEL

The personnel using the TOE services shall be aware of civil, financial and legal responsibilities, as well as the obligations they have to face, depending on their role. The personnel shall be trained on correct usage of the TOE.

### O.ENV\_PROTECT\_ACCESS

The TOE shall be protected by physical, logical and organisational protection measures, in order to prevent any TOE modification, as well as any protected assets disclosure. Those measures shall restrict the TOE usage to authorised persons only.

### O.ENV\_RECOVERY

Recovery plans and procedures shall exist that allow a secure and timely recovery in the case of a major problem with the TOE (i.e. if TOE is blocked in its secure state after a failure, service discontinuity or detected physical tampering). These procedures shall ensure that the confidentiality and integrity of TOE assets and their associated backup keys are always maintained, and especially during recovery and that the recovery does not result in a situation that allows personnel to extend the TOE services they are allowed to use.

#### **O.ENV\_SECURE\_INIT**

Procedures and controls in the TOE environment shall be defined and applied that allow to securely set-up and initialise the TOE for all cryptographic operations including the generation of signatures for qualified certificates or certificate status information. This includes the secure key generation / key import as well as the initial configuration of other TSF data like roles, users and user authentication information. The TOE shall be installed (initialised) with a secure installation procedure using secret data supplied by one or several administrators and entered on a trusted path using split knowledge mechanisms.

#### **O.ENV\_SECURE\_OPER**

Procedures and controls in the TOE environment shall be defined that allow operating the TOE within a CA system in compliance with the requirements of the EU directive and the Policy for certification authorities issuing qualified certificates.

#### **O.ENV\_ADMIN**

Authorised administrators are non-hostile, appropriately trained and follow all administrator guidance.

#### **O.ENV\_PROTECTION\_HOST**

The operating system must be a rugged and secure system that ensures the integrity of sensitive files and allows access only to authorised remote applications.

## Chapter 6. Security Requirements

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

Security functional requirements components given in section 6.1 "TOE security functional requirements" are drawn from Common Criteria part 2 [5]. Some security functional requirements represent extensions to part 2 (nevertheless only components of the reference PP).

The TOE security assurance requirements statements given in section 6.2 "TOE Security Assurance Requirement" are drawn from the security assurance components from Common Criteria part 3 [6].

### 6.1 Security Functional Requirements

This chapter gives the security functional requirements of the PP 8. It also contains some additional requirements.

According to CC part 1, the refinements provided in this section are operations of the security functional requirements and therefore are mandatory parts. The application notes are optional part of the PP and contain additional supporting information that is considered relevant or useful for the construction, evaluation, or use of the TOE but they are not mandatory to fit.

#### 6.1.1 Security audit (FAU)

##### 6.1.1.1 FAU\_GEN.1 Audit data generation

**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) **The following specifically auditable events:**
  - 1) **Initialisation of the TOE,**
  - 2) **Start-up after power up,**
  - 3) **Shutdown of the TOE,**
  - 4) **Software update of the TOE,**
  - 5) **Cryptographic key generation (FCS\_CKM.1): CSP-SCD/CSP-SVD pair generation,**
  - 6) **Cryptographic key distribution (FCS\_CKM.2): entry of R.BACKUP\_KEY,**
  - 7) **Cryptographic key destruction (FCS\_CKM.4): CSP-SCD destruction, destruction of R.BACKUP\_KEY,**
  - 8) **Failure of the random number generator (FCS\_RND.1)**
  - 9) **Unsuccessful recovery (FDP\_BKP.1): Unsuccessful recovery because of detection of modification of the backup data**
  - 10) **Authentication failure handling (FIA\_AFL.1): the reaching of the threshold for the unsuccessful authentication attempts and the actions,**
  - 11) **Timing of authentication (FIA\_UAU.1): all unsuccessful use of the authentication mechanism,**



- 12) **Management of security attributes (FMT\_MSA.1) /(all instantiations): all modifications of the values of security attributes,**
- 13) **Static attribute initialisation (FMT\_MSA.3): modifications of the default setting of permissive or restrictive rules, all modifications of the initial values of security attributes,**
- 14) **Management of TSF data (FMT\_MTD.1/ACCESS\_CONTROL): All modifications to the values of TSF data,**
- 15) **Management of TSF data (FMT\_MTD.1/AUDIT: Export of audit data, Clear (zeroisation) of audit data,**
- 16) **Failure with preservation of secure state (FPT\_FLS.1): Failure detection of the TSF and secure state,**
- 17) **Inter-TSF detection of modification (FPT\_ITI.1): The detection of modification of imported backed-up TSF data,**
- 18) **Notification of physical attack (FPT\_PHP.2): Detection of intrusion (security audit file).**
- 19) **TSF testing (FPT\_TST.1): Execution of the TSF self-tests during initial start-up, at the request of the authorised user, at the conditions installation and maintenance and the results of the tests, unsuccessful self-test operations.**

**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, 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 ST, **identity of the user and sequence data.**

**Refined by adding:**

Date and time of the event may be given by the sequence data correlated to time of export the audit data to the TOE environment. The sequence data shall be a sequence number of the audit event data or time stamp.

**Application note:**

Each audit event record includes date and time. Some events are related to the operation of the equipment and independent of the user, some events are explicitly linked to the roles and other events are implicitly related to the roles (see [7.1.1.1](#)).

### 6.1.1.2 **FAU\_GEN.2 User identity association**

**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.

### 6.1.1.3 **FAU\_STG.2 (TOE) Guarantees of audit data availability**

**FAU\_STG.2.1 (TOE)** The TSF shall protect the stored audit records from unauthorised deletion.

**FAU\_STG.2.2 (TOE)** The TSF shall be able to **prevent** modifications to the audit records in the audit trail.

**FAU\_STG.2.3 (TOE)** The TSF shall ensure that **the last 255** audit records will be maintained when the following conditions occur: **audit storage exhaustion**.

**Application note:**

When storage exhaustion occurs in the general audit file or in the virtual HSM specific audit file, the new audit data overwrite the oldest audit data to guaranty service continuity. The security audit file cannot be erased.

## 6.1.2 Cryptographic support (FCS)

### 6.1.2.1 FCS\_CKM.1 Cryptographic key generation

**FCS\_CKM.1.1** The TSF shall generate cryptographic keys in accordance with *the* specified cryptographic key generation *algorithms listed below* and specified cryptographic key sizes **specified for each algorithm** that meet the following **standards noted for each algorithm**:

1. **RSA 1024 to 4096 bits (step 128) key pairs (in accordance with FIPS PUB 186-4 for key sizes of 2048bits and 3072bits).**
2. **DES 64 bits in accordance with FIPS PUB 46-3.**
3. **DES2 128 bits in accordance with FIPS PUB 46-3.**
4. **TDES 192 bits in accordance with FIPS PUB 46-3.**
5. **AES 128, 192, 256 bits in accordance with FIPS PUB 197.**
6. **ECC 192 to 521 bits in accordance with FIPS PUB 186-4 and ANSI X9.62.**
7. **Generic Secret 32 to 512 bits in accordance with PKCS#11 v2.11.**
8. **EC-KCDSA key pair generation in accordance with "The Korean Certificate-based Digital Signature Algorithm" (1998), 256 to 521 bits**

### 6.1.2.2 FCS\_CKM.1 (backup) Cryptographic key generation

**FCS\_CKM.1.1 (backup)** The TSF shall generate cryptographic keys in accordance with *the* specified cryptographic key generation *algorithms listed below* and specified cryptographic key sizes **specified for each algorithm** that meet the following **standards noted for each algorithm**:

1. **AES 256 bits in accordance FIPS PUB 197.**

### 6.1.2.3 FCS\_CKM.2 (backup\_keys) Cryptographic key distribution

**FCS\_CKM.2.1 (backup\_keys)** The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **key entry** that meets the following: **secure proprietary electronic key distribution method**.

**Refinement**

All encrypted secret or private keys entered into the TOE shall be encrypted and respect the organisational Security Policy **P.Algorithms**. Key entry shall be performed using either manual or electronic methods.

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

- (1) in encrypted form or
- (2) using split knowledge procedures.

Manually-entered keys shall be verified during entry into the TOE for accuracy.

Secret and private keys established using electronic methods shall be entered in encrypted form.

If split knowledge procedures are used:

- (1) The TOE shall separately authenticate the crypto-officer entering each key component.
- (2) At least two key components shall be required to reconstruct the original cryptographic key.

**Application note:**

Due to the SFR FPT\_FLS.1 and FPT\_PHP.3 with their refinements the TOE would not store permanently any private or secret key because this key will be erased after detection of failure or physical tampering. The TSF shall import all secret backup key(s) to restore the TOE to an operational status at a previous point in time. The import of encrypted keys requires a clear key to decrypt these keys in the TOE. Therefore FCS\_CKM.2 ensures that the master key under which all other keys are encrypted for import into the TOE shall be imported by split knowledge procedures. Note that according to FDP\_BKP.1.4 the R.USER\_DATA shall be exported for backup and imported for restore in encrypted form only.

### 6.1.2.4 FCS\_CKM.2 (Other\_keys) Cryptographic key distribution

**FCS\_CKM.2.1 (Other\_keys)** The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **key entry** that meets the following: **keys are entered using the PKCS#11 API**.

### 6.1.2.5 FCS\_CKM.4 Cryptographic key destruction

**FCS\_CKM.4.1** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **zeroisation** that meets the following: **FIPS 140-2 Level 3**.

**Application note:**

The TSF will destroy the R.USER\_DATA and all other plaintext secret or private keys, if the TSF required by FPT\_PHP.2 detects physical tampering.

### 6.1.2.6 FCS\_COP.1 Key Derivation (SNMP dialogue) Cryptographic operation

**FCS\_COP.1.1/ Key Derivation (SNMP dialogue).**The TSF shall perform **key derivation** in accordance with a specified cryptographic algorithm **SHA256** and cryptographic key sizes **256 bits** that meet the following: **RFC 5996 – Internet Key Exchange Protocol Version 2 (IKEv2)**.

### 6.1.2.7 FCS\_COP.1 (SIGN/VERIFY) Cryptographic operation

**FCS\_COP.1.1 (SIGN/VERIFY)** The TSF shall perform **digital signature generation and verification** in accordance with *the* specified cryptographic *algorithms listed below* and cryptographic key sizes *specified for each algorithm* that meet the following: **standards noted for each algorithm:**

1. **RSA, RSA key pairs 1024 to 4096 bits (PKCS #1 v1.5),**
2. **RSA PSS, RSA key pairs 1024 to 4096 bits (PKCS #1 PSS),**

3. **RSA with MD5, SHA-1, SHA-256, SHA-384, SHA-512, RSA key pairs 512 to 4096 bits (PKCS #1 v1.5),**
4. **RSA PSS with SHA-1 SHA-256, SHA-384, SHA-512, RSA key pairs 512 to 4096 bits (PKCS #1 PSS),**
5. **ECDSA, ECC key pairs 192 to 521bits (FIPS PUB 186-4 and ANSI X9.62),**
6. **ECDSA with SHA-1, ECC key pairs 192 to 521bits (FIPS PUB 186-4 and ANSI X9.62),**
7. **EC-KCDSA-SHA256, EC-KCDSA key pairs 256 bits (The Korean Certificate-based Digital Signature Algorithm FIPS PUB 186-43 and ANSI X9.62).**

**Note:**

Curve parameters are sent through the PKCS#11 API.

### 6.1.2.8 FCS\_COP.1 (MESSAGE AUTHENTICATION/VERIFY) Cryptographic operation

**FCS\_COP.1.1 (MESSAGE AUTHENTICATION/VERIFY)** The TSF shall perform **Message authentication generation and verification** in accordance with the specified cryptographic algorithms **listed below** and cryptographic key sizes **specified for each algorithm** that meet the following: **standards noted for each algorithm:**

1. **DES MAC, DES MAC-GENERAL, DES key 64 bits (FIPS PUB 113),**
2. **DES3 MAC, DES3 MAC-GENERAL, DES3 keys 192 bits (FIPS PUB 113),**
3. **AES MAC, AES MAC-GENERAL, AES CMAC, AES CMAC-GENERAL, AES GMAC, AES keys 128, 192, 256 bits (FIPS PUB 197 and FIPS PUB 113),**
4. **MD5 HMAC, MD5 HMAC GENERAL, Generic Secret keys 40 to 192 bits (FIPS PUB 198),**
5. **SHA-1 HMAC, SHA-1 HMAC GENERAL, SHA256 HMAC, SHA256 HMAC GENERAL, SHA384 HMAC, SHA384 HMAC GENERAL, SHA512 HMAC, SHA512 HMAC GENERAL, Generic Secret keys 40 to 512 bits (FIPS PUB 198).**

### 6.1.2.9 FCS\_COP.1 ECDH Cryptographic operation

**FCS\_COP.1.1/ ECDH.** The TSF shall perform **Elliptic Curve Diffie Hellman** in accordance with a specified cryptographic algorithm **EC-DH** and cryptographic key sizes **192 to 521 bits** that meet the following: **ANSI X9-63-2001/RFC5903.**

### 6.1.2.10 FCS\_COP.1 (ENCRYPT/DECRYPT) Cryptographic operation

**FCS\_COP.1.1 (RSA ENCRYPT/DECRYPT)** The TSF shall perform **asymmetric encryption and decryption** in accordance with a specified cryptographic algorithm **RSA** and cryptographic key sizes **1024 to 4096 bits (step 128)** that meet the following: **PKCS#1 V1.5 and OAEP (PKCS#1 v2.1 2002).**

**FCS\_COP.1.1 (DES ENCRYPT/DECRYPT)** The TSF shall perform **symmetric encryption and decryption** in accordance with a specified cryptographic algorithm **DES (ECB and CBC mode)** and cryptographic key sizes **64 bits** that meet the following: **FIPS PUB 46-3**.

**FCS\_COP.1.1 (DES3 ENCRYPT/DECRYPT)** The TSF shall perform **symmetric encryption and decryption** in accordance with a specified cryptographic algorithm **DES3 (ECB and CBC mode)** and cryptographic key sizes **192 bits** that meet the following: **FIPS PUB 46-3**.

**FCS\_COP.1.1 (AES ENCRYPT/DECRYPT)** The TSF shall perform **symmetric encryption and decryption** in accordance with a specified cryptographic algorithm **AES (ECB, CBC and GCM mode)** and cryptographic key sizes **128 bits, 192 bits and 256 bits** that meet the following: **FIPS PUB 197**.

**FCS\_COP.1.1 (ENCRYPT DES3 WITH DES2 KEY).** The TSF shall perform **encryption** in accordance with a specified cryptographic algorithm **DES3 CBC** and cryptographic key sizes **128 bits** that meet the following: **FIPS PUB 197 REV01 26/11/2001 and ANSSI cryptographic referential**.

#### 6.1.2.11 **FCS\_COP.1 (DIGEST) Cryptographic operation**

**FCS\_COP.1.1 (DIGEST)** The TSF shall perform **message digest** in accordance with *the specified cryptographic algorithms listed below*:

1. **MD5 (RFC 1321),**
2. **SHA-1 (FIPS PUB 180-2),**
3. **SHA-256 (FIPS PUB 180-2),**
4. **SHA-384 (FIPS PUB 180-2),**
5. **SHA-512 (FIPS PUB 180-2).**

#### 6.1.2.12 **FCS\_COP.1 (WRAP/UNWRAP) Cryptographic operation**

**FCS\_COP.1.1 (RSA WRAP/UNWRAP)** The TSF shall perform **secret keys wrapping** in accordance with a specified cryptographic algorithm **RSA** and cryptographic key sizes **1024 to 4096 bits (step 128)** that meet the following: **PKCS#1 v1.5 and OAEP (PKCS#1 v2.1 2002)**.

**FCS\_COP.1.1 (AES WRAP/UNWRAP)** The TSF shall perform **private keys wrapping** in accordance with a specified cryptographic algorithm **AES** and cryptographic key sizes **128 and 256 bits** that meet the following: **PKCS#8**.

#### 6.1.2.13 **FCS\_COP.1 (BACKUP\_ENC) Cryptographic operation**

**FCS\_COP.1.1 (BACKUP\_ENC)** The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **AES CBC** and cryptographic key sizes **256 bits** that meet the following: **FIPS PUB 197**.

#### 6.1.2.14 **FCS\_COP.1 (BACKUP\_INT) Cryptographic operation**

**FCS\_COP.1.1 (BACKUP\_INT)** The TSF shall perform **calculation and verification of cryptographic checksums** in accordance with a specified cryptographic algorithm **HMAC-SHA** and cryptographic key sizes **256 bits** that meet the following: **FIPS PUB 198**.

**6.1.2.15 FCS\_RND.1 Quality metrics for random numbers**

**FCS\_RND.1.1** The TSF shall provide a mechanism for generating random numbers that meet with ***ANSI cryptographic referential for "enhanced" strength level [1] and FIPS 140-2 tests criteria.***

**FCS\_RND.1.2** The TSF shall be able to enforce the use of TSF-generated random numbers for **FCS\_CKM.1.**

## 6.1.3 User data protection (FDP)

### 6.1.3.1 FDP\_ACC.1 (CRYPTO) Subset access control

**FDP\_ACC.1.1 (CRYPTO)** The TSF shall enforce the **Crypto-SFP** on:

- User,
- Private keys,
- Public keys,
- R.BACKUP\_KEY
- Data to be processed including DTBS representation,
- Generate private/ public key pair (FCS\_CKM.1),
- Key entry (FCS\_CKM.2/Other\_keys),
- Destruction of private and public keys (FCS\_CKM.4),
- Cryptographic operation including sign DTBS representation (FCS\_COP.1/all iterations).

### 6.1.3.2 FDP\_ACC.1 (CONFIG) Subset access control

**FDP\_ACC.1.1 (CONFIG)** The TSF shall enforce the **Config-SFP** on **User configuration of cryptographic parameters**.

### 6.1.3.3 FDP\_ACC.1 (AUDIT) Subset access control

**FDP\_ACC.1.1 (AUDIT)** The TSF shall enforce the **Audit-SFP** on **User Audit data; export and delete** operations.

### 6.1.3.4 FDP\_ACC.1 (BACKUP) Subset access control

**FDP\_ACC.1.1 (BACKUP)** The TSF shall enforce the **Backup SFP** on **User; R.USER\_DATA and other cryptographic keys, backup key(s), backup data; backup (FDP\_BKP.1), restore (FDP\_BKP.1), backup key entry (FCS\_CKM.2/backup\_keys)**.

### 6.1.3.5 FDP\_ACC.1 (LOAD) Subset access control

**FDP\_ACC.1.1 (LOAD)** The TSF shall enforce the **load SFP** on **User; software code; load software update**.

### 6.1.3.6 FDP\_ACC.1 (DEPERSONALIZATION) Subset access control

**FDP\_ACC.1.1 (DEPERSONALIZATION)** The TSF shall enforce the **Depersonalization SFP** on **User; depersonalization**.

### 6.1.3.7 FDP\_ACF.1 (CRYPTO) Security attribute based access control

**FDP\_ACF.1.1 (CRYPTO)** The TSF shall enforce the **Crypto-SFP** to objects based on **Identity and Role**.

**FDP\_ACF.1.2 (CRYPTO)** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:



1. User with security attribute Role Crypto-officer is allowed to generate (FCS\_CKM.1) the objects R.USER\_DATA, R.USER\_PUB\_KEYS and other cryptographic keys under, at least, dual person control.
2. User with security attribute Role Crypto-officer or Crypto-user is allowed to distribute (FCS\_CKM.2/Other\_keys) the objects R.USER\_DATA, R.USER\_PUB\_KEYS and other cryptographic keys.
3. User with security attribute Role Crypto-officer is allowed to destruct (FCS\_CKM.4) the objects R.USER\_DATA, R.USER\_PUB\_KEYS and other cryptographic keys.
4. User with security attribute Role Crypto-officer is allowed to export R.BACKUP\_KEY.
5. User with security attribute Role Crypto-user is allowed to perform *cryptographic operations* and create signature of the DTBS-representation with R.USER\_DATA (FCS\_COP.1/all iterations).

**FDP\_ACF.1.3 (CRYPTO)** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

**FDP\_ACF.1.4 (CRYPTO)** The TSF shall explicitly deny access of subjects to objects based on the following rules: **User with security attribute Role Crypto-user is not allowed:**

1. to generate (FCS\_CKM.1) the objects R.USER\_DATA, R.USER\_PUB\_KEYS and other cryptographic keys.
2. to destroy (FCS\_CKM.4) the objects R.USER\_DATA, R.USER\_PUB\_KEYS and other cryptographic keys.

### 6.1.3.8 FDP\_ACF.1 (CONFIG) Security attribute based access control

**FDP\_ACF.1.1 (CONFIG)** The TSF shall enforce the **Config-SFP** to objects based on **Identity and Role**.

**FDP\_ACF.1.2 (CONFIG)** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **User with security attribute Role Crypto-officer is allowed to configure the cryptographic parameters.**

**FDP\_ACF.1.3 (CONFIG)** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

**FDP\_ACF.1.4 (CONFIG)** The TSF shall explicitly deny access of subjects to objects based on the **User with security attribute Role Crypto-user is not allowed to configure the cryptographic parameters.**

### 6.1.3.9 FDP\_ACF.1 (AUDIT) Security attribute based access control

**FDP\_ACF.1.1 (AUDIT)** The TSF shall enforce the **Audit-SFP** to objects based on **Identity and Role**.

**FDP\_ACF.1.2 (AUDIT)** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

1. Users with security attribute Role Master Auditor are allowed
  - (a) to export Audit data.



- (b)to clear (zeroisation) Audit data.
- 2. Users with security attribute Role HSM Auditor are allowed (a)to export Audit data generated by the virtual HSM.
- 3. Users with security attribute Role Crypto-officer are allowed to export Audit data.

**FDP\_ACF.1.3 (AUDIT)** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

**FDP\_ACF.1.4 (AUDIT)** The TSF shall explicitly deny access of subjects to objects based on the **following rules**:

- 1. Users with security attribute Role Crypto-officer are not allowed to delete Audit data.
- 2. Users with security attribute Role Crypto-user are not allowed to export or to delete Audit data.
- 3. Users with security attribute Role HSM Crypto- officer are not allowed to export or to delete Audit data

The audit data can only be erased by the role Auditor, by zeroisation method.

### 6.1.3.10 FDP\_ACF.1 (BACKUP) Security attribute based access control

**FDP\_ACF.1.1 (BACKUP)** The TSF shall enforce the **Backup SFP** to objects based on **Identity and-Role**.

**FDP\_ACF.1.2 (BACKUP)** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **User with security attribute Role Crypto-officer is allowed under, at least, dual person control**

- 1. to backup all keys including R.USER\_DATA and R.USER\_PUB\_KEYS (FDP\_BKP.1),
- 2. to restore all keys including R.USER\_DATA and R.USER\_PUB\_KEYS (FDP\_BKP.1).
- 3. to enter back-up keys (FCS\_CKM.2)

**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 **User with security attribute Role Crypto user is not allowed :**

- 1. to backup R.USER\_DATA, R.USER\_PUB\_KEYS and other cryptographic keys (FDP\_BKP.1),
- 2. to enter a back-up key (FCS\_CKM.2)

### 6.1.3.11 FDP\_ACF.1 (LOAD) Security attribute based access control

**FDP\_ACF.1.1 (LOAD)** The TSF shall enforce the **Load-SFP** to objects based on **Identity and-Role**.

**FDP\_ACF.1.2 (LOAD)** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- 1. Users with security attribute Role Master Security Officer are allowed to perform software update

**FDP\_ACF.1.3 (LOAD)** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

**FDP\_ACF.1.4 (LOAD)** The TSF shall explicitly deny access of subjects to objects based on the **following rules**:

- 1. Users with security attribute Role Auditor and HSM Auditor are not allowed to perform software update.**
- 2. Users with security attribute Role Crypto-user are not allowed to perform software update.**

### 6.1.3.12 **FDP\_ACF.1 (DEPERSONALIZATION) Security attribute based access control**

**FDP\_ACF.1.1 (DEPERSONALIZATION)** The TSF shall enforce the **Depersonalization-SFP** to objects based on **Identity and-Role**.

**FDP\_ACF.1.2 (DEPERSONALIZATION)** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- 1. Users with security attribute Role Crypto-officer are allowed to perform virtual HSM depersonalization**

**FDP\_ACF.1.3 (DEPERSONALIZATION)** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

**FDP\_ACF.1.4 (DEPERSONALIZATION)** The TSF shall explicitly deny access of subjects to objects based on the **following rules**:

- 1. Users with security attribute Role Master Security officer are not allowed to perform depersonalization.**
- 2. Users with security attribute Role Auditor and HSM Auditor are not allowed to perform depersonalization.**
- 3. Users with security attribute Role Crypto-user are not allowed to perform depersonalization.**

### 6.1.3.13 **FDP\_BKP.1 Backup and recovery**

**FDP\_BKP.1.1** The TSF shall be capable of invoking the backup function on demand.

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

1. a copy of the same version of the TOE as was used to create the backup data;
2. a stored copy of the backup data;
3. the cryptographic key(s) needed to decrypt the cryptographic keys (R.USER\_DATA) and any other encrypted critical security parameters;
4. the cryptographic key(s) needed to verify the cryptographic checksum of the backup data.

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

**FDP\_BKP.1.4** The cryptographic keys, other critical security parameters and other confidential information shall be exported in encrypted form only.

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

- 6.1.3.14 FDP\_ETC.1 Export of user data without security attributes**
- FDP\_ETC.1.1** The TSF shall enforce the **Crypto-SFP** when exporting user data, controlled under the SFP(s), outside of the TOE.
- FDP\_ETC.1.2** The TSF shall export the user data without the user data's associated security attributes.
- 6.1.3.15 FDP\_RIP.1 Subset residual information protection**
- FDP\_RIP.1.1** The TSF shall ensure that any previous information content of a resource is made unavailable upon the **depersonalization of the resource** from the following objects: **R.USER\_DATA, R.BACKUP\_KEY and RAD.**
- 6.1.3.16 FDP\_SDI.2 Stored data integrity monitoring and action**
- FDP\_SDI.2.1** The TSF shall monitor user data stored within the TSC for **integrity errors** on all objects, based on the following attributes: **error-detecting code.**
- FDP\_SDI.2.2** Upon detection of a data integrity error, the TSF shall **enter the secure blocking state.**
- Refined by adding:**
- The TSF are not required to monitor the DTBS representation for integrity errors.
- Application note:**
- In this context, secure blocking state means a state where the only data returned to the user is an error code and the TOE does not continue to produce a signature value.
- 6.1.4 Identification and authentication (FIA)**
- The Crypto-user role may be associated with only one user – the client application. The client application in the TOE environment may act as agent for more than one user demanding signing of DTBS by the HSM.
- 6.1.4.1 FIA\_AFL.1 Authentication failure handling**
- FIA\_AFL.1.1** The TSF shall detect when **five (5)** unsuccessful authentication attempts occur related to **the Master Security Officer, Master Auditor, HSM Security Officer and HSM Auditor authentication**
- FIA\_AFL.1.2** When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **block the identity for authentication.**
- Application note:**
- The number of authentication failures handling (5) applies to each authentication smart card. The user (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.
- 6.1.4.2 FIA\_ATD.1 User attribute definition**
- FIA\_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users: **Identity and-Role.**

### 6.1.4.3 FIA\_SOS.1 Verification of secrets

**FIA\_SOS.1.1** The TSF shall provide a mechanism to verify that secrets meet **secure proprietary mechanism based on the reconstruction of a key split into several fragments by reading M out of N cards.**

### 6.1.4.4 FIA\_UAU.1 Timing of authentication

**FIA\_UAU.1.1** The TSF shall allow **start-up, self-test (FPT\_TST.1), detection of the secure blocking state (FPT\_FLS.1), detection of violation of physical integrity (FPT\_PHP.2), identification (FIA\_UID.1)** on behalf of the user to be performed before the user is authenticated.

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

### 6.1.4.5 FIA\_UID.1 Timing of identification

**FIA\_UID.1.1** The TSF shall allow **start-up, self-test (FPT\_TST.1), detection of the secure blocking state (FPT\_FLS.1), detection of violation of physical integrity (FPT\_PHP.2)** on behalf of the user to be performed before the user is identified.

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

## 6.1.5 Security management (FMT)

### 6.1.5.1 FMT\_MOF.1 Management of security functions behaviour

**FMT\_MOF.1.1** The TSF shall restrict the ability to **disable, enable** the functions:

- **SF.CO;**  
to **HSM Security officer.**

### 6.1.5.2 FMT\_MSA.1 (ROLE\_CRYPTO) Management of security attributes

**FMT\_MSA.1.1 (ROLE\_CRYPTO)** The TSF shall enforce the **Backup-SFP and Crypto-SFP** to restrict the ability to **query, modify and delete** the security attributes **Role Crypto-user and Role HSM Security-officer** to **HSM Security-officer and Master Security Officer** respectively.

### 6.1.5.3 FMT\_MSA.1 (ROLE\_AUDIT) Management of security attributes

**FMT\_MSA.1.1 (ROLE\_AUDIT)** The TSF shall enforce the **Audit-SFP** to restrict the ability to **query, modify and delete** the security attributes **Role Master Auditor and HSM Auditor** to **Master Auditor**:

**The query, modify and delete operations are performed by Auditor, but under control of the Master Security Officer or HSM Security Officer.**

### 6.1.5.4 FMT\_MSA.2 Secure security attributes

**FMT\_MSA.2.1** The TSF shall ensure that only secure values are accepted for security attributes.

### 6.1.5.5 FMT\_MSA.3 Static attribute initialisation

**FMT\_MSA.3.1** The TSF shall enforce the **Config-SFP, Audit-SFP, Backup SFP, Load-SFP, Depersonalization-SFP and Crypto-SFP**, to provide **restrictive** default values for security attributes that are used to enforce the SFP.

**FMT\_MSA.3.2** The TSF shall allow the **Master Auditor, HSM Security officer and Master Security officer** to specify alternative initial values to override the default values when an object or information is created.

**Operations are performed under control of the Master Security Officer or HSM Security Officer.**

### 6.1.5.6 FMT\_MTD.1 (ACCESS\_CONTROL) Management of TSF data

**FMT\_MTD.1.1 (ACCESS\_CONTROL)** The TSF shall restrict the ability to **query and modify** the **access control lists** to crypto officer authenticated as **Master Security Officer or HSM security officer**.

### 6.1.5.7 FMT\_MTD.1 (USER\_CRYPTO) Management of TSF data

**FMT\_MTD.1.1 (USER\_CRYPTO)** The TSF shall restrict the ability to **change default and delete** the **Identity and RAD for user with role Crypto-officer and Crypto-user to Crypto-officer**. FMT\_MTD.1 (USER\_AUDIT) Management of TSF data

**FMT\_MTD.1.1 (USER\_AUDIT)** The TSF shall restrict the ability to **change default and delete** the **Identity and RAD for user with role attribute Master Auditor to Master Auditor**

**Operations are performed under control of the Master Security Officer or HSM Security Officer.**

### 6.1.5.8 FMT\_MTD.1 (RAD) Management of TSF data

**FMT\_MTD.1.1 (RAD)** The TSF shall restrict the ability to **modify** the RAD to its owner (i.e. **Crypto officer, Crypto user or Auditor**)

**Operations are performed under control of the Master Security Officer or HSM Security Officer.**

### 6.1.5.9 FMT\_MTD.1 (AUDIT) Management of TSF data

**FMT\_MTD.1.1 (AUDIT)** The TSF shall restrict the ability to **query** the **audit data of the TSF required by FAU\_GEN.1 to Auditor and HSM Auditor**.

### 6.1.5.10 FMT\_SMF.1 Specification of Management Functions

**FMT\_SMF.1.1** The TSF shall be capable of performing the following security management functions:

1. **User management (FMT\_MSA.1/ROLE\_CRYPTO, FMT\_MSA.1/ROLE\_AUDIT, FMT\_MTD.1/RAD, FMT\_MTD.1/USER\_CRYPTO and FMT\_MTD.1/USER\_AUDIT),**
2. **Management of audit data (FMT\_MSA.3, FMT\_MTD.1/AUDIT),**
3. **Management of TSF data (FMT\_MTD.1/ACCESS\_CONTROL),**
4. **Management of functions (FMT\_MOF.1).**

### 6.1.5.11 FMT\_SMR.1 Security roles

**FMT\_SMR.1.1** The TSF shall maintain the roles **Master Security Officer, HSM security officer, User , Master Auditor and HSM Auditor.**

**FMT\_SMR.1.2** The TSF shall be able to associate users with roles

**Application note:**

The User role may be associated with only one user – the client application. The client application in the TOE environment may act as agent for more than one user demanding signing of DTBS by the HSM.

## 6.1.6 Privacy (FPR)

### 6.1.6.1 FPR\_UNO.1 (CRYPTO) Unobservability

The TSF shall ensure that **Anybody** are unable to observe the operation

1. **Key generation (FCS\_CKM.1),**
2. **Cryptographic operations, including signature creation (FCS\_COP.1),**
3. **Key destruction (FCS\_CKM.4)**

on **cryptographic keys** by **User, Crypto-officer or Auditor.**

**Application note:**

The TSF requires the TOE to prevent side-channel attacks against the R.USER\_DATA and other secret data where the attack is based on external observable physical phenomena of the TOE.

The set of measurable physical phenomena is influenced by the technology employed to implement the TOE. Examples of measurable phenomena are variations in the timing of transitions of internal states, the power consumption and the electromagnetic radiation. Such phenomena may be caused by normal internal operation of the TOE or may be forced by an attacker who varies the physical environment under which the TOE operates (e.g. power supply, temperature, radio emission or emission of light). Due to the heterogeneous nature of the technologies that may cause such emanations, evaluation is assumed against state-of-the-art attacks applicable to the technologies employed by the TOE. Examples of such attacks are, but are not limited to, evaluation of the TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc.

The TSF requires the TOE to prevent side-channel attacks against the R.USER\_DATA through the intended output data of the TOE e.g. the random padding bits in the signature may contain information about the R.USER\_DATA if both are generated by the same pseudo-random number generator.

### 6.1.6.2 FPR\_UNO.1 (BACKUP) Unobservability

The TSF shall ensure that **anybody** is unable to observe the operation

1. **Key entry (FCS\_CKM.2)**
2. **Key destruction (FCS\_CKM.4)**
3. **Backup (FDP\_BKP.1, FCS\_COP.1/BACKUP\_ENC, FCS\_COP.1/BACKUP\_INT),**
4. **Restore (FDP\_BKP.1, FCS\_COP.1/BACKUP\_ENC, FCS\_COP.1/BACKUP\_INT),**

on **backup keys** by **User, Crypto-officer or Auditor.**

**Application note:**

The TSF requires the TOE to prevent side-channel attacks against the R.USER\_DATA and other secret data, where the attack is based on external observable physical phenomena of the TOE.

The set of measurable physical phenomena is influenced by the technology employed to implement the TOE. Examples of measurable phenomena are variations in the timing of transitions of internal states, the power consumption and the electromagnetic radiation. Such phenomena may be caused by normal internal operation of the TOE or may be forced by an attacker who varies the physical environment under which the TOE operates (e.g. power supply, temperature, radio emission or emission of light). Due to the heterogeneous nature of the technologies that may cause such emanations, evaluation is assumed against state-of-the-art attacks applicable to the technologies employed by the TOE. Examples of such attacks are, but are not limited to, evaluation of the TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc.

## 6.1.7 Protection of the TOE Security Functions (FPT)

### 6.1.7.1 FPT\_FLS.1 Failure with preservation of secure state

**FPT\_FLS.1.1** The TSF shall preserve a secure state when the following types of failures occur: **failures detected by the TSF FPT\_TST.1**.

**Refined by adding:**

The TSF shall destroy the plaintext R.SCP-SCD and other confidential secret and private keys if failures occur.

### 6.1.7.2 FPT\_ITC.1 Inter-TSF confidentiality during transmission

**FPT\_ITC.1.1** The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorised disclosure during transmission.


**Application note:**

The SFR FPT\_ITC.1 addresses the confidentiality protection of the TSF data if they are exported as part of the backup data.

### 6.1.7.3 FPT\_ITI.1 Inter-TSF detection of modification

**FPT\_ITI.1.1** The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric: **cryptographic checksum according to the list of approved algorithms and parameters**.

**FPT\_ITI.1.2** The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform **error indication to the crypto-officer** if modifications are detected.

 **Note:** *the Security Officer is in charge of the creation of the user account using the installation smartcards. He is present during the operation. The Security Officer is therefore able to detect a transmission error. Furthermore the smartcard content is protected by a cryptographic checksum, any modification of this content will be detected during injection in the TSF.*

**Application note:**

The SFR FPT\_ITI.1 addresses the integrity protection of the TSF data if they are imported as part of the backup data.



#### 6.1.7.4 **FPT\_ITT.1 Basic internal TSF data transfer protection**

**FPT\_ITT.1.1** The TSF shall protect TSF data from **modification and disclosure** when it is transmitted between separate parts of the TOE.

**Application note:**

The SFR FPT\_ITT.1 addresses the confidentiality and integrity protection of all cryptographic keys and PKCS#11 data objects.

#### 6.1.7.5 **FPT\_PHP.2 Notification of physical attack**

**FPT\_PHP.2.1** The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

**FPT\_PHP.2.2** The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

**FPT\_PHP.2.3** For **TOE**, the TSF shall monitor the devices and elements and notify anybody when physical tampering with the TSF's devices or TSF's elements has occurred.

**Refined by adding:**

The TSF shall detect physical tampering performed by opening the device, removal or penetration of a cover.

**Application Note:**

The notification about detected physical attacks may be given e.g. through functional interfaces (stopping any other services but alarm signalisation), acoustic or optic signals. The TOE non-IT environment should ensure that notification about physical tampering attempts given by the TOE shall be noticed by the security personnel of the organisation using the TOE services in order to alert privileged users (i.e. HSM security officer or Auditor).

#### 6.1.7.6 **FPT\_PHP.3 Resistance to physical attack**

**FPT\_PHP.3.1** The TSF shall resist **physical tampering by opening the device or removal of a cover** to the **components which:**

- **generate keys (FCS\_CKM.1)**
- **create the signature with R.USER\_DATA (FCS\_COP.1)**
- **perform any other cryptographic operation**
- **store R.USER\_DATA**
- **creates the cryptographic checksum of backup data and encrypts backup data with R.BACKUP\_KEY (FCS\_COP.1.1/BACKUP\_ENC, FCS\_COP.1.1/BACKUP\_INT)**
- **store other secret or private keys**

by responding automatically such that the SFRs are always enforced.

**Refined by adding:**

The TSF shall resist the tampering by destruction of plaintext R.SCP-SCD and other confidential secret and private keys if physical tampering performed by opening the device, or removal of a cover is detected.

**Application Note:**



The TOE protects the confidentiality of the secret and private keys in case of physical maintenance or physical tampering. The TOE will destroy the R.USER\_DATA in case of loss of power. The TOE will invoke the TSF required by FCS\_CKM.4 to destroy the R.USER\_DATA and all other plaintext secret and private keys. The destruction of the R.USER\_DATA will prevent the use of an attacked TOE for signing until restoring the operational state.

#### 6.1.7.7 **FPT\_RCV.1 Manual recovery**

**FPT\_RCV.1.1** After a **failure or service discontinuity**, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

#### 6.1.7.8 **FPT\_STM.1 Time stamps**

**FPT\_STM.1.1** The TSF shall be able to provide reliable time stamps for its own use.

#### 6.1.7.9 **FPT\_TST.1 TSF testing**

**FPT\_TST.1.1** The TSF shall run a suite of self-tests **during initial start-up, at the request of the authorised user, during installation and maintenance** to demonstrate the correct operation of **the TSF**,

by rebooting the TSF on the following cases:

- On request of the authorised user,
- During installation,
- During maintenance.

**FPT\_TST.1.2** The TSF shall provide authorised users with the capability to verify the integrity of **TSF data**.

**FPT\_TST.1.3** The TSF shall provide authorised users with the capability to verify the integrity of **stored TSF executable code**.

##### **Refined by adding:**

The TSF shall perform self-tests:

##### **Initialisation**

- Extended software/firmware integrity and authenticity tests

##### **Power-Up Tests**

- Software/firmware integrity and authenticity tests
- Internal TSF data integrity test
- Cryptographic algorithm tests
- Random number generator tests
- Critical functions tests

##### **Conditional Tests**

- Pair-wise consistency test (for public and private keys)
- Manual key entry test (if manual key entry is implemented)
- Continuous random number generator test

**Application note:**

The TSF performs self-tests according to FPT\_TST.1 to ensure that the TOE is functioning properly. The extended software/firmware integrity test might verify error-detecting codes, cryptographic checksums or digital signatures generated by the software/firmware developer or by other authorities. A digital signature might prove that the firmware or software is part of the evaluated product. The power-up software/firmware integrity test and internal TSF data integrity test may detect modification of these data if the device was switched off. The tests may be implemented by internally generated error detecting codes, cryptographic checksums or digital signatures. The cryptographic algorithm test may detect errors in hardware, firmware or software implementing critical cryptographic mechanisms (see FCS\_CKM.1, FCS\_COP.1). The test might be a known-answer-test (e.g. for encryption) or a pair-wise consistency test (e.g. verifying a generated signature before the signature is exported).

Supplementary tests shall detect error of the random number generator used for the generation of R.USER\_DATA (see FCS\_CKM.1 and FCS\_RND.1). If any critical function is not covered by these tests the TSF should implement additional self-tests.

The pair-wise consistency test for public and private keys may detect errors in the key generation process. Other consistency tests may check the correctness of the signing process and other cryptographic processes to prevent e.g. differential fault attacks. Manual key entry test may detect errors to prevent use of incorrect keys if manual key entry is implemented.

Continuous random number generator test may detect failure in operation of the generator to prevent use of wrong random number.

The TOE shall verify the integrity and authenticity of the TSF executable code at installation, maintenance and initialisation to prevent malicious software running on the TOE.

## 6.1.8 Trusted path (FTP)

### 6.1.8.1 FTP\_TRP.1 (TOE) Trusted path

**FTP\_TRP.1.1 (TOE)** The TSF shall provide a communication path between itself and **local** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **modification and disclosure**.

**FTP\_TRP.1.2 (TOE)** The TSF shall permit **local users** to initiate communication via the trusted path.

**FTP\_TRP.1.3 (TOE)** The TSF shall require the use of the trusted path for **initial user authentication (FIA\_UID.1, FIA\_UAU.1) and TSF management (FMT\_MOF.1, FMT\_MSA.1/ROLE, FMT\_MTD.1/USER\_CRYPT, FMT\_MTD.1/USER\_AUDIT, FMT\_MTD.1/RAD, FMT\_MSA.2, FMT\_MSA.3, FMT\_MTD.1/ACCESS, FMT\_MTD.1/AUDIT, FMT\_SMR.1)**.

**Application Note:**

Local users are those that interact with the TOE using the local interface (see Figure 6: TOE and environment general overview).

## 6.2 TOE Security Assurance Requirements

The assurance level is EAL4 augmented with the following components:

- 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).

Assurance Class	Assurance Components
ADV	ADV_ARC.1; ADV_FSP.4; ADV_IMP.2; ADV_TDS.3.
AGD	AGD_OPE.1; AGD_PRE.1
ALC	ALC_CMC.5; ALC_CMS.4; ALC_DEL.1; ALC_DVS.2; ALC_FLR.3; ALC_LCD.1; ALC_TAT.1.
ATE	ATE_COV.2; ATE_DPT.1; ATE_FUN.1; ATE_IND.2.
AVA	AVA_VAN.5.

**Tableau 6-1 - Assurance Requirements: EAL 4 augmented.**

## Chapter 7. TOE summary specification

### 7.1 TOE Security functions

#### 7.1.1 Audit Data Generation (SF.AUDIT)

##### 7.1.1.1 SF.AUDIT.EVENTS

The TOE generates an audit record of all events related to the TOE security (unsuccessful self-tests, authentication failure, embedded software update, ...).

There are two types of audit files:

- The audit file of security events related to the appliance
  - The events are independent of user roles
- The audit file of security events related to the security module (one audit file per virtual HSM)
  - Certain events relate to the operation of the appliance and are therefore not associated with a role;
  - Other events are explicitly associated to a role (PIN code verification);
  - Other events are implicitly associated to a role ( virtual HSM creation (Security officer), cryptographic configuration modification (Crypto-officer), ...

##### 7.1.1.2 SF.AUDIT.FILE

All security events are recorded in flash memory. There is a general audit file, a security audit file and a virtual HSM specific audit file. The general audit file and the security audit file can be read via an administration command under Auditor control, and the virtual HSM specific audit file can be read via an administration command under HSM Auditor control. When storage exhaustion occurs in the general audit file or in the virtual HSM specific audit file, the new audit data overwrite the oldest audit data to guaranty service continuity. The security audit file of virtual HSMs can be erased under HSM auditor control, the general file cannot be erased.

### 7.1.2 Authentication (SF.AUTHENTICATION)

#### 7.1.2.1 SF. AUTHENTICATION.ROLES

The TOE supports the following user categories Security Officer (SO), Auditor, Crypto-officer, HSM Auditor, Crypto-user. Access rights are described in §3.5.6 Roles.

The TOE allows for the creation of multiple users in the Crypto-officer and Crypto-user roles. Each user is created within a cryptographically separated partition in Bull HSM and each partition must have one and only one user in the Crypto-officer role. A partition may also have one and only one user in the Crypto-user role. It is possible to have up to eight (8) partitions defined within Bull HSM

### 7.1.2.2 **SF.AUTHENTICATION.TRUSTED\_PATH**

The authentication of the Security Officer, Auditor, Crypto-officer or HSM Auditor, takes place via the smart card reader housed into the appliance, which is linked by a serial connection (trusted path) to the TOE.

The protection of the communicated data is then mainly achieved by the secure IT environment through this local serial connection.

The user authentication is executed by password.

### 7.1.2.3 **SF.AUTHENTICATION.POLICY**

When authentication is required, the smart card reader asks for the user's smart card and the TOE is blocked in this state.

Authentication consists in verifying that the smart card can answer to a challenge based on a proprietary algorithm implementing encryption and hash functions.

FIA\_AFL.1 requires the TOE to detect and respond to failed authentication attempts.

#### **Failed authentication attempt due to wrong PIN code**

The authentication of the PIN code is handled entirely by the smart card. The user can have up to 5 tries.

The HSM generates warning messages that are sent to the smart card reader:

Bad code Try again            (upon first, second or third failure)

Bad code Last try            (fourth failure)

Card blocked                (fifth and last failure)

The authentication card then becomes unusable.

#### **Failed authentication due to bad authentication card**

The TOE allows 5 consecutive failed authentication attempts with a wrong authentication card. At the end of 5 consecutive failed authentication attempts, the authentication card becomes unusable.

The number of authentication failures handling (5) applies to each authentication smart card. The user (Security Officer, Auditor, Crypto-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.

Authentication of Crypto-user is performed with a password.

The password is configured by the Crypto-officer.

If the Crypto-user enters two consecutive wrong passwords, the TOE imposes a 3,5 seconds waiting time between each new retry.

## 7.1.3 Access control (SF.ACCESS\_CONTROL)

The TOE protects the sensitive data from unauthorised access (user administrative data, keys ...).

The TOE is configured by an administration function with authentication by the Crypto-officer. This configuration is used to:

- Forbid usage of cryptographic functions
- Restrict key sizes

The authentication will be requested following the user and the function to be executed. For example, to be able to execute a software update, the Security officer will have to previously authenticate himself. Likewise, the Crypto-officer will have to authenticate himself to be able to modify the cryptographic configuration.

## 7.1.4 HSM management

### 7.1.4.1 Secure installation (SF.SI)

The TOE must be installed before it can be used in any way (use of the PKCS#11, authentication, TOE update services, etc.).

This installation process takes place once the TOE has been physically installed and comprises the following steps:

- At the first power on, the roles Security officer and Auditor are created, as well as their authentication smart cards.
- Create the installation cards (creation of N smart cards, from which M can be used to install the virtual HSM).
- Create the virtual HSMs with their installation cards (creation of Crypto-officer and virtual HSM auditor roles).
- Personalize the virtual HSMs and edit their cryptographic configuration. Personalize a virtual HSM means:
  - Create the role Crypto-user for this virtual HSM and choose its associated password.
  - Configure the virtual HSM start mode (automatic mode or smart card mode).
- Start the virtual HSMs.

## 7.1.5 Cryptographic operations (SF.CO)

### 7.1.5.1 SF.CO.KEY\_GENERATION

All the symmetrical and asymmetrical keys as well as the random codes used by the dual-key generation functions are generated according to the process described here after:

- RSA 1024 to 4096 bits (step 128) key pairs (in accordance with FIPS PUB 186-4 for key sizes of 2048bits and 3072bits).
- DES 64 bits in accordance with FIPS PUB 46-3.
- DES2 128 bits in accordance with FIPS PUB 46-3.
- TDES 192 bits in accordance with FIPS PUB 46-3.
- AES 128, 192, 256 bits in accordance with FIPS PUB 197.

- ECC 192 to 521 bits in accordance with FIPS PUB 186-4 and ANSI X9.62.
- Generic Secret 32 to 512 bits in accordance with PKCS#11 v2.11.

The token uses a hardware based random number generator that meets ANSSI cryptographic referential for "Enhanced" strength level [2] and FIPS 140-2 tests criteria.

Key-pair consistency test is performed according to Miller-Rabin algorithm.

### 7.1.5.2 SF.CO.KEY\_DESTRUCTION

The destruction of the keys complies with FPS140-2, Section 4.7.6, Key zeroisation.

The destruction of a particular (encrypted) key stored in the secure memory takes place by setting the relevant memory location to zero.

The red (non-encrypted) keys stored in the cryptographic module are destroyed (zeroisation) following the activation of certain alarms (intrusion detection while the appliance is powered on, low battery, ...).

### 7.1.5.3 SF.CO.CRYPTOGRAPHIC\_FUNCTIONS

The TOE implements the cryptographic algorithms described in § 3.5.2.

The TOE implements the cryptographic operations described in § 3.5.1.

Trustway Proteccio implements specific PKCS#11 functions, such as C\_CreateObject, allowing enter secret, public or private keys.

## 7.1.6 Secure loading (SF.SL)

The executable code is loaded into the TOE in two cases:

- When pre-personalizing the TOE;
- When updating the TOE embedded software.

The TOE pre-personalization is an operation performed into BULL environment. It allows the update of:

- The cryptographic module (FPGA) software which comprises the FPGA bitstream and the software of NIOS processors;
- The COMExpress module software.

The update operation involves replacing the binary code in the card by new binary code. This operation is carried out by the Security Officer, once he has been authenticated. The operation is recorded in the general events log.

### 7.1.6.1 General mechanism

The principle of the secure loading procedure involves verifying an ECKCDSA signature against a hash of the code to be loaded.

Upon completion of the production phase, the TOE contains the loader code and a public key that provides a means of checking the signature of the binary code to be loaded.

Upon completion of the pre-personalization phase, the TOE's flash memory contains all the binary code that it needs in order to operate.

The signature of both codes (cryptographic module and Linux system), are verified at each system start-up.

## 7.1.7 Security mechanisms (SF.SM)

### 7.1.7.1 SF.SM.HARDWARE

The hardware security mechanisms ensure that the card operates properly and protect the integrity of its sensitive data (keys and algorithms) by monitoring the temperature and the various voltages used by the module. Additional alarms originating from the system and from the outside world (via the flat band connector) are also taken into account (intruder detection, panic button), and cause a security alert to be activated. The implementation of the various mechanisms is described below.

Hardware security mechanisms are implemented at two levels:

- Level1 (opening of the housing, for example), available with the power on and off.
- Level2 (temperature monitoring, voltage monitoring, emergency erase), available only with the power on.

All components of the security module are embedded in a hard opaque potting material (resin).

Level 1 alarms provoke the zeroisation of BULL keys.

Level 2 alarms lead to a blocking state and provoke the zeroisation of internal FPGA key memories.

### 7.1.7.2 SF.SM.KEYS

All the cryptographic keys and DATA objects of the TOE are protected in confidentiality and in integrity.

The cryptographic keys must not appear in plaintext out of the internal FPGA memories.

Keys are encrypted when they are created and decrypted in a secure memory before being used by an automaton.

### 7.1.7.3 SF.SM.TESTS

#### 7.1.7.3.1 Start-up tests

At start-up, the BISTs test all security elements of the TOE. If an error is detected the test stops and an error message appears on the LCD. The appliance is restarted 5 seconds later.

In case of voltage or temperature alarm, the appliance is stopped.

#### 7.1.7.3.2 Periodic tests and fault management

Software security mechanisms involve a series of periodic tests that continuously monitor the functioning and integrity of sensitive functions:

- Cryptographic operations;
- The random number generator.

#### 7.1.7.3.3 Code integrity tests

The FPGA software integrity is verified:

- FPGA bitstream ;



- Software executed by the processors.

The integrity checking of the software running on the ComExpress module (Linux) is provided by the PCA4 BIOS every time the appliance starts:

- Detect the active code partition (initial, flip or flop);
- Verify the active code partition signature;
- Boot the active partition.

The integrity of sensitive files of the software running on the ComExpress module is periodically checked by AIDE. It operates by regularly testing the integrity of certain sensitive files aiming to counter attacks which may modify or alter these files.

#### **7.1.7.3.4 Secure memory integrity tests**

The cryptographic keys must not appear in clear mode outside the FPGA internal memories.

A key is encrypted (black key) when it is generated and decrypted (red key) in a secure memory at the moment it must be used by an automaton.

A malfunction in the decryption mechanism can be immediately detected for each operation independently of continuous testing.

#### **7.1.7.4 SF.SM.ALARMS**

##### **7.1.7.4.1 Error report**

The presence of an error or anomaly is reported via an event code recorded into the security log file. (in flash memory).

##### **7.1.7.4.2 Dealing with an error**

If an error is detected during power-on tests, a message is recorded into the security log file, the appliance restarts or stops.

##### **7.1.7.4.3 LEDs meaning**

The TOE provides a visual indication of the status of its operations and internal security.

The visual indication is realized by means of a status two-coloured LED (Ready/Error/Alarm), connected to the FPGA.

#### **7.1.7.5 SF.SM.DEPERSONALIZATION**

A virtual HSM can be depersonalized by its security officer (crypto-officer). The secret elements introduced during the personalization phase, together with the user keys are erased.

The TOE has an emergency pushbutton (only effective in power-up mode) provoking the HSM depersonalization.

### **7.1.8 Backup and Recovery (SF.BACKUP)**

#### **7.1.8.1 SF.BACKUP .COMMAND**

The TOE provides the capability to securely backup the user's private and secret keys.

**7.1.8.2 SF.BACKUP.AUDIT**

The failures of key storage and reloading operations are all recorded into the general events logs.

**7.1.8.3 SF.BACKUP.DATA\_PROTECTION**

Each key is protected by the encryption of the secret elements of the key and by a MAC which ensures the identification, authenticity and integrity of the keys.

The wrapping key is generated by the token at initialisation time and cannot be extracted from the token by the application.

Conversely, the reloading of one or more keys into the TOE involves a transfer to the TOE of key structures that were generated and output by the TOE in the first place with for each key a control of the MAC and a decrypting of the secret elements.

## Chapter 8. PP claims

### 8.1 Reference PP

See § 2.4 PP conformance.

### 8.2 PP addition

The TOE is intended to be used as a general purpose cryptographic card. Thus, threats, security objectives and security requirements defined in the PP have been generalised to all cryptographic keys and all cryptographic operations.

For each threat, assumption, security objective and security objective for the environment it will be explicitly indicated if it has been added, expanded or iterated regarding to the PP:

A : Addition

E : Expansion

I : Iteration

#### 8.2.1 Threats

- T.KEYS\_DERIVE (E)
  - T.KEYS\_DISCLOSE (E)
  - T.KEYS\_ALTERATION (E)
  - T.MISUSE\_OPERATION (E)
  - T.CRYPTO\_FORGERY (E)
  - T.TRUSTED\_PATH (A)
  - T.INSECURE\_CHANNEL (A)

#### 8.2.2 Assumptions

- A.ADMIN (A)
  - A.PROTECTION\_HOST (A)

#### 8.2.3 Security objectives

- O.KEYS\_SECURE (E)
  - O.CRYPTO\_SECURE (E)
  - O.SECURE\_LOADING (A)
  - O.TRUSTED\_PATH (A)
  - O.DEPERSONALIZATION (A)

#### 8.2.4 Security objectives for the environment

- O.ENV\_ADMIN (A)
  - O.ENV\_PROTECTION\_HOST (A)
  - O.ENV\_SECURE\_CHANNEL (E)

#### 8.2.5 Security Functional Requirements

- FCS\_COP.1 /SIGN/VERIFY (E)

- FCS\_COP.1/MESSAGE AUTHENTICATION VERIFY (I)
- FCS\_COP.1 /ECDH (I)
- FCS\_COP.1/ENCRYPT/DECRYPT (I)
- FCS\_COP.1/DIGEST (I)
- FCS\_COP.1/WRAP/ UNWRAP (I)
- FDP\_ACC.1/CONFIG (I)
- FDP\_ACC.1/LOAD (I)
- FDP\_ACC.1 /DEPERSONALIZATION (I)
- FDP\_ACF.1/CONFIG (I)
- FDP\_ACF.1/LOAD (I)
- FDP\_ACF.1 /DEPERSONALIZATION (I)
- FMT\_MOF.1 (A)
- FPT\_ITI.1/Basic Internal TSF data transfer protection (A)
- FPT\_STM.1 (A)

ADV\_IMP.2 (implementation of the TSF), ALC\_CMC.5 (Advanced Support), ALC\_DVS.2 (Development Security) and ALC.FLR.3 (Systematic flaw remediation) security assurance requirements have been added to the PP.

419221-2 Protection Profile considers a TOE without any trusted path.

The TOE related to this security target has a trusted path physically independent from application path. This trusted path ensures secure transmission of the authentication data. Therefore T.TRUSTED\_PATH threat has been added to impose a trusted path for all authentication issues for the following roles: security officer, auditor, crypto-officer, HSM auditor. This threat is mapped by a new security objective for the TOE (O.TRUSTED\_PATH). The objective is covered by FTP\_TRP.1/TOE SFR.

The TOE provides a communication path between the applications (client applications and administration application) and the TOE used to transfer authentication (crypto-user) and management data. This security objective for the environment maps T.INSECURE\_CHANNEL threat imposing a secure communication between the applications and the TOE (O. ENV\_SECURE\_CHANNEL).

# Chapter 9. Rationale

## 9.1 Introduction

The TOE that has been defined covers cryptographic modules that implement—partly or completely—the functionality necessary for devices involved in generating the advanced electronic signatures of qualified certificates. The tables in sub-sections 9.2.1 “Security Objectives Coverage” and “9.3.1 “Security Requirement Coverage” provide the mapping of the security objectives and security requirements for these TOE types.

## 9.2 Security Objectives Rationale

### 9.2.1 Security Objectives Coverage

Policy/Threat/Assumptions	Objectives
<b>Policies</b>	
P.ALGORITHMS	O.KEYS_SECURE, O.CRYPTO_SECURE, O.PROTECT_EXPORTED_DATA, O.USER_AUTHENTICATION
<b>Threats</b>	
T.BACKUP	O.AUDIT, O.CHECK_OPERATION, O.RBAC, O.SECURE_STATE, O.PROTECT_EXPORTED_DATA, O.USER_AUTHENTICATION, O.ENV_APPLICATION, O.ENV_AUDIT, O.ENV_PERSONNEL, O.ENV_RECOVERY; O.BACKUP_SECURE
T.BACKUP_KEY_Alteration T.BACKUP_KEY_Derive T.BACKUP_KEY_Disclose	O.CHECK_OPERATION, , O.ATTACK_RESPONSE, O.SECURE_STATE, O.BACKUP_SECURE ; O.BACKUP_KEY_SECURE , O.ENV_PROTECT_ACCESS, O.ENV_SECURE_INIT
T.BAD_SW	O.AUDIT, O.SECURE_LOADING, O.CHECK_OPERATION, O.RBAC, O.SECURE_STATE, O.USER_AUTHENTICATION, O.ENV_APPLICATION, O.ENV_AUDIT, O.ENV_SECURE_CHANNEL, O.ENV_PERSONNEL
T.KEYS_DERIVE	O.KEYS_SECURE, O.CRYPTO_SECURE
T.KEYS_DISCLOSE	O.KEYS_SECURE, O.PROTECT_EXPORTED_DATA, O.CRYPTO_SECURE, O.ENV_SECURE_INIT
T.KEYS_ALTERATION	O.KEYS_SECURE, O.CHECK_OPERATION, O.ATTACK_RESPONSE, O.SECURE_STATE, O.ENV_PROTECT_ACCESS, O.ENV_SECURE_INIT, O.DEPERSONALIZATION
T.CSP_SVD_ALTERATION	O.PROTECT_EXPORTED_DATA, O.ENV_APPLICATION
T.DATA_MANIPUL	O.ENV_APPLICATION, O.ENV_SECURE_CHANNEL
T.MALFUNCTION	O.AUDIT, O.CHECK_OPERATION, O.SECURE_STATE, O.ENV_AUDIT, O.ENV_PERSONNEL, O.ENV_PROTECT_ACCESS, O.ENV_RECOVERY
T.INSECURE_INIT	O.AUDIT, O.RBAC, O.SECURE_STATE, O.USER_AUTHENTICATION, O.ENV_APPLICATION, O.ENV_AUDIT, O.ENV_SECURE_CHANNEL, O.ENV_PERSONNEL, O.ENV_SECURE_INIT, O.DEPERSONALIZATION
T.MISUSE_OF_TOE	O.AUDIT, O.RBAC, O.USER_AUTHENTICATION, O.ENV_AUDIT, O.ENV_PERSONNEL, O.ENV_SECURE_OPER, O.DEPERSONALIZATION

Policy/Threat/Assumptions	Objectives
T.MISUSE_OPERATION	O.KEYS_SECURE, O.RBAC, O.CRYPTO_SECURE, O.USER_AUTHENTICATION, O.ENV_SECURE_OPER
T.PHYS_MANIPUL	O.AUDIT, O.CHECK_OPERATION, O.ATTACK_RESPONSE, O.SECURE_STATE, O.ENV_AUDIT, O.ENV_PROTECT_ACCESS
T.INSECURE_CHANNEL	O.ENV_SECURE_CHANNEL
T.TRUSTED_PATH	O.TRUSTED_PATH
T.CRYPTO_FORGERY	O.CRYPTO_SECURE
T.KeyGeneration_Misuse	O.AUDIT, O.RBAC, O.USER_AUTHENTICATION, O.ENV_APPLICATION, O.ENV_AUDIT, O.ENV_PERSONNEL, O.ENV_PROTECT_ACCESS, O.ENV_SECURE_OPER
Assumptions	
A.AUDIT_SUPPORT	O.ENV_APPLICATION, O.ENV_AUDIT, O.ENV_PERSONNEL
A.CORRECT_DTBS	O.ENV_APPLICATION, O.ENV_SECURE_OPER
A.DATA_STORE	O.ENV_PERSONNEL, O.ENV_RECOVERY, O.ENV_SECURE_OPER
A.CRYPTOUSER_AGENT	O.ENV_APPLICATION
A.TRUSTED_ENVIRONMENT	O.ENV_PROTECT_ACCESS, O.ENV_SECURE_OPER
A.ADMIN	O.ENV_ADMIN
A.PROTECTION_HOST	O.ENV_PROTECTION_HOST

**Tableau 9-1 - Security Environment to Security Objectives Mapping**

Objectives	Policy/Threat/Assumptions
Security Objectives for the TOE	
O.AUDIT	T.BACKUP, T.BAD_SW, T.MALFUNCTION, T.INSECURE_INIT, T.MALFUNCTION, T.MISUSE_OF_TOE, T.PHYS_MANIPUL, T.KeyGenertion_Misuse
O.KEYS_SECURE	T.KEYS_DERIVE, T.KEYS_DISCLOSE, T.KEYS_ALTERATION, T.MISUSE_OPERATION, P.ALGORITHMS
O.CHECK_OPERATION	T.BACKUP, T.BAD_SW, T.KEYS_ALTERATION, T.MALFUNCTION, T.PHYS_MANIPUL, T.BACKUP_KEY_Manipulation
O.RBAC	T.BACKUP, T.BAD_SW, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.MISUSE_OPERATION; T.KeyGenertion_Misuse
O.ATTACK_RESPONSE	T.KEYS_ALTERATION, T.PHYS_MANIPUL, T.BACKUP_KEY_Manipulation
O.SECURE_STATE	T.BACKUP, T.BAD_SW, T.KEYS_ALTERATION, T.MALFUNCTION, T.INSECURE_INIT, T.PHYS_MANIPUL, T.BACKUP_KEY_Manipulation
O.PROTECT_EXPORTED_DATA	T.BACKUP, T.KEYS_DISCLOSE, T.CSP_SVD_ALTERATION, P.ALGORITHMS
O.CRYPTO_SECURE	T.KEYS_DERIVE, T.KEYS_DISCLOSE, T.MISUSE_OPERATION, T.CRYPTO_FORGERY, P.ALGORITHMS
O.USER_AUTHENTICATION	T.BACKUP, T.BAD_SW, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.MISUSE_OPERATION, P.ALGORITHMS
O.TRUSTED_PATH	T.TRUSTED_PATH
O.SECURE_LOADING	T.BAD_SW

Objectives	Policy/Threat/Assumptions
O.DEPERSONALIZATION	T.KEYS_ALTERATION, T.INSECURE_INIT, T.MISUSE_OF_TOE
O.BACKUP_SECURE	T.BACKUP , T.BACKUP_KEY_Manipulation
O.BACKUP_KEY_SECURE	T.BACKUP_KEY_Manipulation

Objectives	Policy/Threat/Assumptions
<b>Security Objectives for the Environment</b>	
O.ENV_APPLICATION	T.BACKUP, T.BAD_SW, T.CSP_SVD_ALTERATION, T.DATA_MANIPUL, T.INSECURE_INIT, A.AUDIT_SUPPORT, A.CORRECT_DTBS, A.CRYPTOUSER_AGENT, T.KeyGenertion_Misuse
O.ENV_AUDIT	T.BACKUP, T.BAD_SW, T.MALFUNCTION, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.PHYS_MANIPUL , A.AUDIT_SUPPORT, T.KeyGenertion_Misuse
O.ENV_PERSONNEL	T.BACKUP, T.BAD_SW, T.MALFUNCTION, T.INSECURE_INIT, T.MISUSE_OF_TOE, A.AUDIT_SUPPORT, A.DATA_STORE, T.KeyGenertion_Misuse
O.ENV_PROTECT_ACCESS	T.KEYS_ALTERATION, T.MALFUNCTION, T.PHYS_MANIPUL, A.TRUSTED_ENVIRONMENT, T.KeyGenertion_Misuse, T.BACKUP_KEY_Manipulation
O.ENV_RECOVERY	T.BACKUP_RESTORE, T.MALFUNCTION, A.DATA_STORE
O.ENV_SECURE_INIT	T.KEYS_DISCLOSE, T.KEYS_ALTERATION, T.INSECURE_INIT, T.BACKUP_KEY_Manipulation
O.ENV_SECURE_OPER	T.MISUSE_OF_TOE, T.MISUSE_OPERATION, A.TRUSTED_ENVIRONMENT, A.CORRECT_DTBS, A.DATA_STORE, T.KeyGenertion_Misuse
O.ENV_ADMIN	A.ADMIN
O.ENV_SECURE_CHANNEL	T.BAD_SW, T.DATA_MANIPUL, T.INSECURE_INIT, T.INSECURE_CHANNEL
O.ENV_PROTECTION_HOST	A.PROTECTION_HOST

**Tableau 9-2 - Tracing of Security Objectives to the TOE Security Environment**

## 9.2.2 Security Objectives Sufficiency

The following paragraphs provide the rational between Security Objectives versus Threats, OSP and Assumptions.

### 9.2.2.1 Menaces

#### T.BACKUP

Backup and Restore operations shall be auditable events, recorded in a general events log file (O.AUDIT) and protected in integrity (O.CHECK\_OPERATION). Only authenticated users shall perform them (O.USER\_AUTHENTICATION) thanks to a reliable client application (O.RBAC, O.ENV\_APPLICATION). Auditor has access to restore and backup logs by analysis of stored audit logs (O.ENV\_AUDIT).

The data export mechanism shall use adequate confidentiality and integrity cryptographic mechanism to prevent and detect any tampering over backup data (O.PROTECT\_EXPORTED\_DATA). Recovery plans and procedures shall explain the correct usage of Backup data, and describe backup and restore operations (O.CHECK\_OPERATION, O.ENV\_RECOVERY) and personnel shall be trained to perform these tasks using a reliable application (O.ENV\_PERSONNEL, O.ENV\_APPLICATION). In case of error detection during restore operation, the TOE shall enter a secure state (O.SECURE\_STATE).

#### T.BACKUP\_KEY\_Manipulation

The TOE shall only allow authenticated users to perform backup and restore operations (O.CHECK\_OPERATION, O.BACKUP\_SECURE, O.BACKUP\_KEY\_SECURE).

The TOE Shall provide a recovery plan that allows for swift and sure recovery in case of major TOE issue (O.CHECK\_OPERATION, O.BACKUP\_SECURE, O.BACKUP\_KEY\_SECURE).

Personnel shall be trained to perform these tasks using a reliable application that performs the required verifications on data (O.ENV\_PROTECT\_ACCESS, O.ENV\_SECURE\_INIT).

The TOE shall detect any change to R.BACKUP\_KEY in its environment and log any restore errors in the secure log (O.AUDIT).

In case of error detection during restore operation, the TOE shall enter a secure state (O.SECURE\_STATE).

#### T.BAD\_SW

Only Security Officer role can perform firmware update (O.RBAC). Therefore a reliable authentication shall be done to ensure that user's identity (O.USER\_AUTHENTICATION) is associated with the Security Officer role. This association is done by the administration application (O.ENV\_APPLICATION). The Security Officer shall be aware of the consequences of his acts and trained (O.ENV\_PERSONNEL). This kind of operation can have an important security impact on the TOE and its lifecycle. This is the reason why it shall be logged for future audit (O.AUDIT). The operational environment of the TOE shall provide technical solutions for audit storage and edition (O.ENV\_AUDIT).



The data uploaded in the TOE shall be authenticated (O.CHECK\_OPERATION) and verified in integrity (O.CHECK\_OPERATION) prior to be installed in the TOE. Revision number of the data set to be uploaded shall be verified in order to counter any downgrading attempt (O.CHECK\_OPERATION). These data shall be uploaded through a secure channel (O.ENV\_SECURE\_CHANNEL) to lower the risk of distant software attack via the communication port of the TOE. O.SECURE\_LOADING ensures that the TOE provides a secure loading process. In case of error during the update, the TOE shall return to a secure state, i.e. not applying the patch or step to a secure blocked state (O.SECURE\_STATE)

### **T.KEYS\_DERIVE**

The electronic signature algorithm and process that are used for the signature operation shall not leak information that might help to derive R.USER\_DATA and other cryptographic keys (O.CRYPTO\_SECURE). This means that every data involved in the electronic signature (R.DTBS, R.USER\_DATA and other cryptographic keys, or even signed data) shall not embed information about the secret key. This is also covered by the security objective of confidentiality over R.USER\_DATA and other cryptographic keys (O.KEYS\_SECURE).

### **T.KEYS\_DISCLOSE**

The TOE shall ensure integrity and confidentiality of R.USER\_DATA and other cryptographic keys (O.KEYS\_SECURE). Moreover, the electronic signature operation itself shall not leak information about R.USER\_DATA and other cryptographic keys (O.CRYPTO\_SECURE). Of course, the TOE shall not export R.USER\_DATA and other cryptographic keys (needed for backup) without protecting its confidentiality (O.PROTECT\_EXPORTED\_DATA).

In order to proceed to a secure electronic signature operation, procedures and controls in the TOE environment shall be defined and applied that allow secure key generation (O.ENV\_SECURE\_INIT).

### **T.KEYS\_ALTERATION**

The TOE shall ensure integrity of R.USER\_DATA and other cryptographic keys (O.KEYS\_SECURE). This is partially achieved with a nominal initialization of R.TSF\_DATA (O.ENV\_SECURE\_INIT). During normal operation, integrity of cryptographic material shall be checked by the TOE (O.CHECK\_OPERATION). Alteration of R.USER\_DATA and other cryptographic keys might come from a physical attack. Therefore, if such a data alteration arises, the TOE shall detect the attack, respond (O.ATTACK\_RESPONSE) and jump to a secure state (O.SECURE\_STATE) requiring a new personalisation (O.DEPERSONALIZATION) to prevent loss of confidentiality from secret elements. To lower the risk of physical attack, the TOE shall be used in a secure place (O.ENV\_PROTECT\_ACCESS).

The TOE shall prevent the export of clear text R.USER\_DATA.

### **T.CSP\_SVD\_ALTERATION**

Applications that use the TOE shall not alter the exported data (O.ENV\_APPLICATION). Moreover, the TOE shall apply integrity and confidentiality protection measures to all assets listed in the asset list requiring integrity or confidentiality protection when they are exported from the TOE (O.PROTECT\_EXPORTED\_DATA).

### **T.DATA\_MANIPUL**

Applications that use the TOE shall perform the necessary security checks on the data passed to the TOE and user authentication (O.ENV\_APPLICATION, O.USER\_AUTHENTICATION). Security check in the TOE environment shall also prevent any unauthorised manipulation of the data transmitted to the TOE. Nevertheless, the threat can also come from the outside world and therefore, a secure channel has to be set between the client application and the TOE (O.ENV\_SECURE\_CHANNEL).

### T.MALFUNCTION

- The TOE shall regularly verify R.USER\_DATA (O.CHECK\_OPERATION). Malfunction shall be detected by monitoring operation of the TOE (O.CHECK\_OPERATION). In case a malfunction arises, it shall be recorded (O.AUDIT) for future exploitation by maintenance services, and eventually compared with previous log files (O.ENV\_AUDIT). In case of malfunction, the TOE shall jump to a secure state (O.SECURE\_STATE). If malfunctions arise, personnel shall behave adequately (O.ENV\_PERSONNEL) and manage to set up a recovery solution (O.ENV\_RECOVERY) to avoid service discontinuity.
- To lower the risk of tampering with the TOE, the TOE shall be used in a secure place, protected by organisational and logical means (O.ENV\_PROTECT\_ACCESS).

### T.INSECURE\_INIT

Applications used for TOE initialisation shall perform the necessary security checks on the data passed to the TOE and user authentication (O.ENV\_APPLICATION, O.USER\_AUTHENTICATION). TOE must restrict access to its services depending on the role, to the services explicitly assigned to this role, even during initialisation (O.RBAC).

Initialisation data have to be uploaded securely into the TOE (O.ENV\_APPLICATION + O.ENV\_SECURE\_CHANNEL) and verified by the TOE itself before being validated inside the TOE in authenticity and integrity (O.CHECK\_OPERATION)

Personnel that operate the TOE shall be aware of civil, financial and legal responsibilities, and trained (O.ENV\_PERSONNEL), and they should apply the procedures and controls that allow to securely set-up and initialise the TOE for the generation of signatures for qualified certificates or certificate status information. This includes the secure key generation / key import as well as the initial configuration of other TSF data like roles, users and user authentication information (O.ENV\_SECURE\_INIT).

TOE shall ensure that all secrets are erased after depersonalisation (O.DEPERSONALIZATION).

A TOE may also be initialised to be the copy of another TOE that became unusable e. g. because of a hardware failure. In this case the TOE needs to be initialised with TSF data that has been previously exported from the other TOE. O.PROTECT\_EXPORTED\_DATA addresses the issue that this data has been manipulated after it has been exported. This allows the new TOE to get securely initialised with the data of the old TOE.

Critical operation shall be logged (O.AUDIT + O.ENV\_AUDIT)

TOE environment shall ensure the availability of the audit record (O.ENV\_AUDIT).

If a problem appears during the initialisation process, the TOE shall jump or remain in a secure state (O.SECURE\_STATE).

### T.MISUSE\_OF\_TOE

TOE must restrict access to its cryptographic services to authorised users only (O.RBAC).

TOE must be able to identify and authenticate users with regards to their role before granting them access to TOE assets (O.USER\_AUTHENTICATION).

TOE shall extend its protection to the end-users of the client application via user authentication and access control (O.USER\_AUTHENTICATION)

Personnel using the cryptographic services of the TOE must be aware on the implication of their activity and be formed to correctly accomplish their tasks (O.ENV\_PERSONNEL).

Procedures and controls in the TOE environment must be defined, that allow the TOE to operate within a CA system in compliance with the requirements of the EU directive (O.ENV\_SECURE\_OPER)

TOE shall allow the verification of the fact that an unauthorised user have tried to access the TOE or an authorised user have tried to force the TOE by using functions which are forbidden to him (O.AUDIT).

TOE environment shall ensure the availability of the audit record (O.ENV\_AUDIT).

TOE shall destroy R.USER\_DATA and enter a state requiring a new personalisation to be operational (O.DEPERSONALIZATION).

### **T.MISUSE\_OPERATION**

TOE shall restrict the usage of cryptographic functions to authorised users (O.RBAC).

TOE must be able to identify and authenticate users with regards to their role before granting them access to TOE assets (O.USER\_AUTHENTICATION).

Personnel using the cryptographic services of the TOE must be aware on the implication of their activity and be formed to correctly accomplish their tasks (O.ENV\_PERSONNEL).

TOE shall ensure that no information related to R.USER\_DATA is directly transmitted to any entity outside of the TOE (O.KEYS\_SECURE).

Toe shall ensure that the algorithms and their implementation do not allow the disclosure of R.USER\_DATA (O.CRYPTO\_SECURE).

TOE environment will not facilitate the wrongful usage of the TOE (O.ENV\_SECURE\_OPER).

All physical manipulation shall be logged in the audit file (O.AUDIT).

TOE environment shall ensure the availability of the audit record (O.ENV\_AUDIT).

### **T.PHYS\_MANIPUL**

TOE shall detect any intrusion attempt and securely destroy R.USER\_DATA in such a case (O.ATTACK\_RESPONSE).

Physical manipulation can be also detected by a loss of integrity of critical data, therefore they need to be regularly checked (O.CHECK\_OPERATION)

The TOE shall enter a secure state upon detection of an error that could have been caused by a physical intrusion (O.SECURE\_STATE).

To prevent modification of the TOE and disclosure of assets, the TOE shall be protected by physical, logical and organisational measures (O.ENV\_PROTECT\_ACCESS)

### **T.INSECURE\_CHANNEL**

An attacker could manipulate sensitive data from applications sent by an authorised user to the TOE during the network transmission, and thus affect the TOE initialisation or configuration.

The environment must ensure the confidentiality and integrity of the data transferred between the applications (client applications and administration application) and the TOE by providing a secure communication channel (O.ENV\_SECURE\_CHANNEL).

### **T.TRUSTED\_PATH**

An attacker could manipulate sensitive authentication data sent by an unauthorised personnel to the TOE, and thus affect the TOE initialisation or configuration.

The TOE will provide a secure communication path for the users, independent of the one with the client application. This trusted path ensures that the user's authentication and identification data are correctly transmitted to the TOE (O.TRUSTED\_PATH).

### **T.CRYPTO\_FORGERY**

This threat describes the fact that an attacker is able to generate a forged signature with the result that either a forged qualified signature or forged certificate status information is generated. While the threat of disclosing information about R.USER\_DATA is covered elsewhere, this threat deals with the problem that it might be able for someone to forge a signature without knowledge of the R.USER\_DATA. O.CRYPTO\_SECURE counters this threat by stating that it should not be possible to generate a valid signature without knowledge of the R.USER\_DATA.

### **T.KeyGeneration\_Misuse**

TOE shall restrict the usage of cryptographic functions to authorised users (O.RBAC).

TOE must be able to identify and authenticate users with regards to their role before granting them access to key generation functions (O.USER\_AUTHENTICATION).

TOE shall ensure that the algorithms and their implementation do not allow the disclosure of R.USER\_DATA (O.CRYPTO\_SECURE).

All physical manipulation shall be logged in the audit file (O.AUDIT).

TOE environment shall ensure the availability of the audit record (O.ENV\_AUDIT).

## 9.2.2.2 Organisational Security Policies (OSP)

### P.ALGORITHMS

Cryptographic algorithms used in O.CRYPTO\_SECURE have to comply with the referential for "Enhanced" strength level edited by ANSSI ([1], [2], [3]).

O.KEYS\_SECURE, O.CRYPTO\_SECURE, O.PROTECT\_EXPORTED\_DATA and O.USER\_AUTHENTICATION shall also use adequate cryptographic algorithms, to comply with the same referential.

## 9.2.2.3 Assumptions

### A.AUDIT\_SUPPORT

This assumption assumes that audit capabilities of the TOE will be exploited usefully by Auditors that are trained and aware of their responsibilities (O.ENV\_PERSONNEL) thanks to a trusted Client Application (O.ENV\_APPLICATION) and the whole operational system that make the TOE audit trails available (O.ENV\_AUDIT).

### A.CORRECT\_DTBS

This assumption assumes that the operational environment of the TOE provides correct organisational procedures. This assumption relies on the Client application (O.ENV\_APPLICATION) and its capability to establish a secure communication channel with the TOE. Finally, a set of operational procedures must be in place for the organisation operating the TOE as part of their certification system (O.ENV\_SECURE\_OPER).

### A.DATA\_STORE

This assumption is supported by O.ENV\_SECURE\_INIT and O.ENV\_SECURE\_OPER witch deals with the security of sensitive data required for initialisation, start-up and exploitation of the TOE when they are stored outside of the TOE. It is also supported by O.ENV\_PERSONNEL that ensures the availability of data stored inside the TOE.

### A.CRYPTOUSER\_AGENT

This assumption assumes that the only crypto-user is the client application and that it performs efficiently the user authentication operations for the Crypto-User role (O.ENV\_APPLICATION).

### A.TRUSTED\_ENVIRONMENT

This assumption assumes that the operational environment of the TOE is secure (O.ENV\_PROTECT\_ACCESS) because the TOE by itself cannot verify this property. Finally, a set of operational procedures must be in place for the organisation operating the TOE as part of their certification system (O.ENV\_SECURE\_OPER).

### A.ADMIN

This assumption is met by the objective O.ENV\_ADMIN, which ensures that the administrators are non-hostile, appropriately trained and have the means to correctly perform their tasks.

### A.PROTECTION\_HOST

This assumption is met by O.ENV\_PROTECTION\_HOST, which assumes that the operating system is a rugged and secure system that ensures the integrity of sensitive files and allows access only to authorised remote applications.

## 9.3 Security Requirements Rationale

### 9.3.1 Security Requirement Coverage

Objectives	Requirements
<b>Security Objectives for the TOE</b>	
O.AUDIT	FAU_GEN.1, FAU_GEN.2, FAU_STG.2/TOE, FDP_ACC.1/AUDIT, FDP_ACF.1/AUDIT, FMT_MTD.1/AUDIT, FMT_SMF.1, FPT_ITI.1, FPT_STM.1
O.PROTECT_EXPORTED_DATA	FAU_GEN.1, FAU_GEN.2, FCS_CKM.1 (backup), FCS_CKM.2/backup_keys, FCS_COP.1/BACKUP_ENC, FCS_COP.1/BACKUP_INT, FDP_ACC.1/BACKUP, FDP_ACF.1/BACKUP, FDP_BKP.1, FDP_ETC.1, FMT_MSA.1/ROLE_CRYPT, FMT_MSA.3, FPR_UNO.1/FPT_ITC.1, FPT_ITI.1, FMT_MOF.1
O.KEYS_SECURE	FCS_CKM.1, FCS_CKM.2/Other_Keys, FCS_CKM.4, FCS_COP.1/all iterations, FCS_RND.1, FDP_ACC.1/CRYPTO, FDP_ACF.1/CRYPTO, FDP_BKP.1, FDP_RIP.1, FDP_SDI.2, FPR_UNO.1, FPT_ITT.1
O.CHECK_OPERATION	FAU_GEN.1, FAU_GEN.2, FPT_TST.1
O.RBAC	FDP_ACC.1/AUDIT, FDP_ACC.1/BACKUP, FDP_ACC.1/CRYPTO, FDP_ACC.1/LOAD, FDP_ACC.1/CONFIG, FDP_ACF.1/AUDIT, FDP_ACF.1/BACKUP, FDP_ACF.1/CRYPTO, FDP_ACF.1/LOAD, FDP_ACF.1/CONFIG, FMT_MSA.1/ ROLE_CRYPT, FMT_MSA.1/ROLE_AUDIT, FMT_MOF.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1/ACCESS_CONTROL, FMT_MTD.1/USER_CRYPT, FMT_MTD.1/USER_AUDIT, FMT_MTD.1/RAD, FMT_MTD.1/AUDIT, FPT_TST.1, FMT_SMR.1
O.ATTACK_RESPONSE	FPT_PHP.2, FPT_PHP.3
O.SECURE_STATE	FPT_FLS.1, FPT_RCV.1, FPT_TST.1
O.CRYPTO_SECURE	FCS_COP.1/all iterations, FPR_UNO.1
O.USER_AUTHENTICATION	FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.1, FIA_UID.1, FMT_MTD.1/USER_CRYPT, FMT_MTD.1/USER_AUDIT, FMT_MTD.1/RAD, FMT_MTD.1/AUDIT, FMT_SMF.1, FTP_TRP.1/TOE
O.TRUSTED_PATH	FTP_TRP.1/TOE
O.SECURE_LOADING	FAU_GEN.1, FAU_GEN.2, FCS_COP.1/VERIFY, FCS_COP.1/DECRYPT, FDP_ACC.1/LOAD, FDP_ACF.1/LOAD
O. DEPERSONALIZATION	FDP_ACC.1/DEPERSONALIZATION, FDP_ACF.1/DEPERSONALIZATION, FDP_RIP.1
O.BACKUP_SECURE	FCS_COP.1 / BACKUP_ENC, FCS_COP.1 / BACKUP_INT, FDP_ACC.1 / BACKUP, FDP_ACF.1 / BACKUP
O.BACKUP_KEY_SECURE	FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1 / BACKUP_ENC, FCS_COP.1 / BACKUP_INT, FDP_ACC.1 / BACKUP, FDP_ACF.1 / BACKUP, FDP_RIP.1, FDP_SDI.2.

**Tableau 9-3 - Functional and Assurance Requirement to Security Objective Mapping**

## 9.3.2 Security Requirements Sufficiency

### 9.3.2.1 TOE Security Requirements Sufficiency

#### O.AUDIT

This objective addresses the generation and protection of audit data by the TOE. The audit generation is implemented by the SFR FAU\_GEN.1 with the audit events matching the list in O.AUDIT.

FAU\_GEN.2, ensures that the events are associated to the identity of the users. The TOE stores the audit data according to the SFR FAU\_STG.2/TOE until the audit trail is exported upon request of the Auditor or Crypto-officer under control of the SFR FDP\_ACC.1/AUDIT, FDP\_ACF.1/AUDIT and FMT\_MTD.1/AUDIT.

– FMT\_SMF.1 and FMT\_MTD.1/AUDIT require management function for the audit. These management functions are provided to the Auditor only. The integrity of the audit data will be ensured by the SFR FAU\_STG.2/TOE inside the TOE. FPT\_STM.1 guarantees a reliable time stamp.

#### O.PROTECT\_EXPORTED\_DATA

This objective addresses the integrity and confidentiality protection measures to all assets listed in the asset list requiring integrity or confidentiality protection when they are exported from the TOE. The SFR FDP\_ETC.1 implements the Crypto-SFP for all exported data. The TOE backup and restore functions require the SFR FDP\_BKP.1 the confidentiality and integrity protection of backup data. The backup and restore of R.USER\_DATA, other user data and TSF data is described in the SFR FDP\_BKP.1. The confidentiality and integrity protection of the TSF data as part of the backup data is implemented by the SFR FPT\_ITC.1 and SFR FPT\_ITI.1.

The FDP\_BKP.1 needs the cryptographic functions implemented by the following SFR:

- Generation of backup keys FCS\_CKM.1/backup
- Import the backup keys by FCS\_CKM.2/backup,
- Encryption of backup data by FCS\_COP.1/BACKUP\_ENC,
- Data integrity protection by FCS\_COP.1/BACKUP\_INT.

The SFR FDP\_BKP.1 requires encrypting the CSP\_SCD and electronically exported keys if they are exported. The backup and restore TSF will be under access control required by the SFR FDP\_ACF.1/BACKUP according to FDP\_ACC.1/BACKUP. The SFR FMT\_MSA.1/ROLE\_BACKUP and FMT\_MSA.3 extend the management functions of security attributes to the Backup SFP. The SFR FAU\_GEN.1 require audit data specific for the use of the backup and restore function associated with the identity of the users (FAU\_GEN.2). Because FDP\_BKP.1 handles and exports the CSP\_SCD outside the TOE, the TOE shall protect against side-channels to prevent any illicit information flow. The SFR FPR\_UNO.1 implements this protection.



## O.KEYS\_SECURE

This objective addresses the confidentiality and integrity of the R.USER\_DATA which shall be ensured during their whole life time. The SFR ensure the cryptographic secure R.USER\_DATA generation by FCS\_CKM.1 and FCS\_RND.1 as well as operation by FCS\_COP.1/all iterations according to the list of approved algorithms and parameters for "Enhanced" level [2]. The confidentiality and integrity of the R.USER\_DATA will be protected by SFR FDP\_RIP.1, FDP\_SDI.2 and FPT\_ITT.1 while internal processing. The SFR FCS\_CKM.2/Other\_keys ensures the distribution of cryptographic keys. The SFR FCS\_CKM.4 requires secure key destruction to prevent any misuse of R.USER\_DATA after operational lifetime. All R.USER\_DATA management and operation is under access control of the SFR FDP\_ACC.1/CRYPTO and FDP\_ACF.1/CRYPTO. The TOE shall protect R.USER\_DATA against side-channels by the SFR FDP\_UNO.1.

Note that the special protection of the R.USER\_DATA needed if the R.USER\_DATA is exported by backup function. This is addressed by O.PROTECT\_EXPORTED\_DATA and implemented by appropriate SFR. The SFR FDP\_BKP.1 will protect the confidentiality if the R.USER\_DATA (or any other cryptographic key) is exported.

## O.CHECK\_OPERATION

This objective address regular checks to verify that its components operate correctly. This security objective is implemented in the TOE by the SFR FPT\_TST.1 (TSF Testing). If these tests detect an error the TOE will transit into a secure state (see O.SECURE\_STATE) and prevent the normal operation. FAU\_GEN.1 generates audit records about the test results of FPT\_TST.1 to inform the user (Auditor) about the performed self-tests and their results. FAU\_GEN.2 ensures the association between user identity and audit event. The FPT\_TST.1 includes checks of the executable code. It also covers security of the firmware update operation (integrity, authenticity and anti-replay mechanism over uploaded data).

## O.RBAC

This objective addresses the access control to TOE services and its management. The access control is implemented in the TOE by :

- a) FDP\_ACC.1/CRYPTO and FDP\_ACF.1/CRYPTO for the cryptographic functions (Crypto-SFP),
- b) FDP\_ACC.1/AUDIT and FDP\_ACF.1/AUDIT for the audit function (Audit-SFP),
- c) FDP\_ACC.1/LOAD and FDP\_ACF.1/LOAD for the software update function (Load-SFP),
- d) FDP\_ACC.1/CONFIG and FDP\_ACF.1/CONFIG for the cryptographic configuration function (Config-SFP),
- e) FDP\_ACC.1/BACKUP and FDP\_ACF.1/BACKUP for the backup function (Backup-SFP)

with the roles Auditor, HSM Auditor, Security Officer, Crypto-officer and Crypto-user as defined by the SFR FMT\_SMR.1.

The SFR FMT\_MSA.1/ROLE\_CRYPTO, FMT\_MSA.1/ROLE\_AUDIT, FMT\_MSA.2, FMT\_MSA.3, FMT\_MTD.1/ACCESS\_CONTROL, FMT\_MTD.1/AUDIT and FMT\_SMF.1 assign the management functions for the cryptographic to the Crypto-officer and audit functions to the Auditor and HSM Auditor. The SFR FMT\_MSA.1/ROLE\_CRYPTO extends the Crypto-user's cryptographic functions to backup and restore.

The SFR requires the TSF to enforce the Audit-SFP, Backup-SFP and Crypto-SFP to provide restrictive default values for security attributes that may be changed by the Auditor and the Crypto-officer.

The user management is addressed by O.USER\_AUTHENTICATION.

## O.ATTACK\_RESPONSE

This objective addresses the detection of physical tampering attempts and the secure destruction of the R.USER\_DATA if such attempts are detected. The SFR FPT\_PHP.2 implements notification of and FPT\_PHP.3 resistance to physical attack. The refinements limit the tamper scenarios to opening the device or removal of a cover. This limitation is reasonable because O.ENV\_PROTECT\_ACCESS requires security measures for physical protection of the TOE by the organisation.

## O.SECURE\_STATE

This objective addresses a secure state and protection of R.USER\_DATA confidentiality whenever the TOE detects a defect or an integrity error.

The SFR FPT\_TST.1 requires tests for error detection and the SFR FPT\_FLS.1 requires preservation of a secure state when errors are detected.

The SFR FPT\_RCV.1 requires a maintenance mode where the ability to return the TOE to a secure state is provided.

O.ENV\_RECOVERY describes the related security measures in the TOE environment.

## O.CRYPTO\_SECURE

This objective addresses the security of all the cryptographic operations, including the signatures, i.e. the signature does not reveal the R.USER\_DATA and cannot be forged without knowledge of the R.USER\_DATA.

The cryptographic security of cryptographic operations is implemented by the SFR FCS\_COP.1/all iterations in compliance with the "Enhanced" level ANSSI referential [2]. T

he SFR FPR\_UNO.1 requires TSF to prevent illicit information flow about the R.USER\_DATA through side-channels in the signatures.

## O.USER\_AUTHENTICATION

This objective addresses the identification and authentication the users before having any access to TOE protected assets.

The SFR FIA\_AFL.1 protects the VAD against guessing.

The SFR FIA\_ATD.1 defines the security attributes for identity based authentication.

The SFR FIA\_SOS.1 ensures the verification of the quality of the secret used for authentication.

The SFR require timing identification by FIA\_UID.1 and timing authentication by FIA\_UAU.1.

The SFR FMT\_MTD.1/USER\_CRYPT, FMT\_MTD.1/USER\_AUDIT, FMT\_MTD.1/RAD and FMT\_SMF.1 provide management functions for identification.

The SFR FTP\_TRP.1/TOE provides a trusted path for identification and authentication of users with the TOE, ensuring their correct transmission.

The following actions are allowed on behalf of the user to be performed before the user is identified respectively authenticated: start-up, identification (FIA\_UID.1), self-test (FPT\_TST.1), detection of the secure blocking state (FPT\_FLS.1) and detection of violation of physical integrity (FPT\_PHP.2).

## O.TRUSTED\_PATH

This objective addresses a trusted communication path with human users which must be physically independent from application path. This trusted path will ensure that the identification and authentication data of TOE users are transmitted correctly and in a confidential way to the TOE. The objective is covered by FTP\_TRP.1/TOE.

### **O.SECURE\_LOADING**

This objective addresses a secure loading process to update the TOE embedded software.

The loading operation must be performed by applying integrity and confidentiality protection measures to protect any loading from malicious software.

FAU\_GEN.1 generates audit records about the software update results.

FAU\_GEN.2, ensures that the events are associated to the identity of the users.

The software is decrypted according to FCS\_COP.1/DECRYPT and verified according to FCS\_COP.1/VERIFY.

The load TSF will be under access control required by FDP\_ACF.1/LOAD according to FDP\_ACC.1/LOAD.

### **O. DEPERSONALIZATION**

This objective addresses the depersonalization of a virtual HSM. FDP\_ACC.1/DEPERSONALIZATION et FDP\_ACF.1/ DEPERSONALIZATION, assure that the depersonalization of a virtual HSM is under access control.

The SFR FDP\_RIP.1 assure that the user keys and the authentication data will not be accessible once the virtual HSM has been depersonalized.

### **O.BACKUP\_SECURE**

This objective addresses the security of cryptographic operations (checksum and encryption) of the TOE associated with backup. For instance, the checksum must not disclose R.BACKUP\_KEY and cannot be forged without the knowledge of R.BACKUP\_KEY. Idem with the encryption.

FCS\_COP.1/BACKUP\_INT implements the cryptographic security of the checksum to ensure integrity.

FCS\_COP.1/BACKUP\_ENC implements the cryptographic security of the encryption.

FDP\_ACC.1/BACKUP and FDP\_ACF.1/BACKUP implement access control for backup functions.

FPR\_UNO.1/BACKUP prevents information leakage of R.BACKUP\_KEY via side-channels.

### **O.BACKUP\_KEY\_SECURE**

This objective addresses the confidentiality and integrity of R.BACKUP\_KEY during their whole life cycle.

FCS\_CKM.1 and FCS\_RND.1 as well as FCS\_COP.1/BACKUP\_ENC and FCS\_COP.1/BACKUP\_INT ensures correct and secure generation of R.BACKUP\_KEY in compliance with the approved algorithm list and their parameters.

FDP\_RIP.1 and FDP\_SDI.2 implements internal confidentiality and integrity of R.BACKUP\_KEY

FCS\_CKM.2 ensures the secure management of R.BACKUP\_KEY during initialisation.

FCS\_CKM.4 requires secure destruction in order to prevent a wrongful usage of R.BACKUP\_KEY once it is no longer operational.

Management and usage of R.BACKUP\_KEY are under the control of FDP\_ACC.1/CRYPTO and FDP\_ACF.1/CRYPTO for the hardware and FDP\_ACC.1/BACKUP and FDP\_ACF.1/BACKUP for backup operations.

FPR\_UNO.1/BACKUP prevents information leakage of R.BACKUP\_KEY via side-channels.

FPT\_PHP.3 requires physical protection of components that use R.BACKUP\_KEY.

## 9.4 TOE Summary Specification Rationale

### 9.4.1 TOE Security functions Coverage

Security requirements	TOE security functions
FAU_GEN.1	SF.AUDIT.EVENTS, SF.AUDIT.FILE, SF.BACKUP.AUDIT, SF.SL
FAU_GEN.2	SF.AUDIT.EVENTS
FAU_STG.2/TOE	SF.AUDIT.FILE
FCS_CKM.1	SF.CO.KEY_GENERATION
FCS_CKM.1/backup	SF.CO.KEY_GENERATION, SF.SI
FCS_CKM.2/backup_keys	SF.BACKUP.COMMAND
FCS_CKM.2/Other_keys	SF.CO.CRYPTOGRAPHIC_FUNCTIONS
FCS_CKM.4	SF.CO.KEY_DESTRUCTION
FCS_COP.1/Key_Derivation	SF.CO.CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/SIGN/ VERIFY	SF.CO.CRYPTOGRAPHIC_FUNCTIONS, SF.SL
FCS_COP.1/ECDH	SF.CO.CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/ENCRYPT/DECRYPT	SF.CO. CRYPTOGRAPHIC_FUNCTIONS, SF.SL,
FCS_COP.1/MESSAGE AUTHENTICATION/MESSAGE AUTHENTICATION VERIFY	SF.CO. CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/DIGEST	SF.CO. CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/WRAP/UNWRAP	SF.CO. CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/BACKUP_ENC	SF.BACKUP.DATA_PROTECTION
FCS_COP.1/BACKUP_INT	SF.BACKUP.DATA_PROTECTION
FCS_RND.1	SF.CO.KEY_GENERATION
FDP_ACC.1/CRYPTO	SF.ACCESS_CONTROL
FDP_ACC.1/AUDIT	SF.ACCESS_CONTROL
FDP_ACC.1/BACKUP	SF.ACCESS_CONTROL
FDP_ACC.1/LOAD	SF.ACCESS_CONTROL
FDP_ACC.1/CONFIG	SF.ACCESS_CONTROL
FDP_ACC.1/DEPERSONALIZATION	SF.ACCESS_CONTROL, SF.SM. DEPERSONALIZATION
FDP_ACF.1/CRYPTO	SF.AUTHENTICATION.ROLES
FDP_ACF.1/AUDIT	SF.AUTHENTICATION.ROLES
FDP_ACF.1/BACKUP	SF.AUTHENTICATION.ROLES
FDP_ACF.1/LOAD	SF.AUTHENTICATION.ROLES
FDP_ACF.1/CONFIG	SF.AUTHENTICATION.ROLES
FDP_ACF.1/DEPERSONALIZATION	SF.AUTHENTICATION.ROLES, SF.SM. DEPERSONALIZATION
FDP_BKP.1	SF.BACKUP.COMMAND, SF.BACKUP.DATA_PROTECTION
FDP_ETC.1	SF.BACKUP.DATA_PROTECTION
FDP_RIP.1	SF.SI, SF.SM. DEPERSONALIZATION
FDP_SDI.2	SF.SM.TESTS

Security requirements	TOE security functions
FIA_AFL.1	SF.AUTHENTICATION.POLICY
FIA_ATD.1	SF.AUTHENTICATION.ROLES
FIA_SOS.1	SF.AUTHENTICATION.POLICY
FIA_UAU.1	SF.SM.TESTS, SF.SM.ALARMS, SF.AUTHENTICATION.POLICY
FIA_UID.1	SF.SM.TESTS, SF.SM.ALARMS
FMT_MOF.1	SF.ACCESS_CONTROL, SF.AUTHENTICATION.ROLES
FMT_MSA.1/ROLE_CRYPT0	SF.AUTHENTICATION.ROLES
FMT_MSA.1/ROLE_AUDIT	SF.AUTHENTICATION.ROLES
FMT_MSA.2	SF.AUTHENTICATION.ROLES
FMT_MSA.3	SF.AUTHENTICATION.ROLES
FMT_MTD.1/ACCESS_CONTROL	SF.ACCESS_CONTROL, SF.AUTHENTICATION.ROLES
FMT_MTD.1/USER_CRYPT0	SF.ACCESS_CONTROL, SF.AUTHENTICATION.ROLES
FMT_MTD.1/USER_AUDIT	SF.ACCESS_CONTROL, SF.AUTHENTICATION.ROLES
FMT_MTD.1/RAD	SF.ACCESS_CONTROL, SF.AUTHENTICATION.ROLES
FMT_MTD.1/AUDIT	SF.ACCESS_CONTROL, SF.AUTHENTICATION.ROLES
FMT_SMF.1	SF.AUTHENTICATION.ROLES, SF.ACCESS_CONTROL
FMT_SMR.1	SF.AUTHENTICATION.ROLES
FPR_UNO.1	SF.SM.HARDWARE
FPT_FLS.1	SF.SM.ALARMS
FPT_ITC.1	SF.BACKUP.DATA_PROTECTION
FPT_ITI.1	SF.BACKUP.DATA_PROTECTION
FPT_ITT.1	SF.SM.KEYS
FPT_PHP.2	SF.SM.ALARMS
FPT_PHP.3	SF.SM.HARDWARE, SF.SM.ALARMS
FPT_RCV.1	SF.SI
FPT_STM.1	SF.SI
FPT_TST.1	SF.SM.TESTS, SF.CO.KEY_GENERATION
FTP_TRP.1/TOE	SF.AUTHENTICATION.TRUSTED_PATH

**Tableau 9-4 - TOE security functions to Security Requirements Mapping**

## 9.4.2 TOE Security functions Sufficiency

### FAU\_GEN.1 (Audit Data Generation)

Outlines the data that must be included in audit records and the events that must be audited. This component is met by SF.AUDIT.EVENTS (events handling), SF.AUDIT.FILE (audit file handling), SF.BACKUP.AUDIT (backup audit function), SF.SL (software update audit function).

### FAU\_GEN.2 (Audit Data Generation)

Ensures that each event is associated to a user identity. Event record described in SF.AUDIT.EVENTS indicates the user association.

### FAU\_STG.2/TOE (Guarantees of audit data availability)

Guaranties audit data availability. SF.AUDIT.FILE ensures audit file protection in TOE flash memory and defines policy when storage exhaustion occurs.

### FCS\_CKM.1 (Cryptographic Key Generation)

Ensures that the keys generated are of adequate strength. SF.CO.KEY\_GENERATION performs generic secret, RSA (including Key-pair consistency test), AES, ECC key generation.

### FCS\_CKM.1/backup (Cryptographic Key Generation)

Ensures that the backup keys generated are of adequate strength. SF.CO.KEY\_GENERATION performs backup key generation. The backup keys are created at initialisation time as described by SF.SI.

### FCS\_CKM.2/backup\_keys (Cryptographic Key Distribution)

Ensures that the backup keys are distributed securely to provide confidentiality and integrity of backup data including backup data transmitted between peer TOEs. SF.SI ensures backup key distribution on smart cards at re-initialisation time (SF.SI also ensures sharing of backup keys between different TOEs).

### FCS\_CKM.2/Other\_keys (Cryptographic Key Distribution)

Ensures that the backup keys are distributed securely. SF.CO.CRYPTOGRAPHIC\_FUNCTIONS assures that key distribution is performed through the PKCS#11 API.

### FCS\_CKM.4 (Cryptographic Key Destruction)

Ensures that the keys are correctly destroyed. SF.CO.KEY\_DESTRUCTION defines the key memory erasing method.

**FCS\_COP.1/Key Derivation (Cryptographic Operation)**

Ensures that key derivations are done in compliance with the approved standards.

SF.CO.CRYPTOGRAPHIC\_FUNCTIONS, implements all key derivation methods.

**FCS\_COP.1/ECDH (Cryptographic Operation)**

Ensures that key derivations ECDH algorithm is done according to ANSI X9-63-2001/RFC5903 standards

**FCS\_COP.1/(SIGN /VERIFY) (Cryptographic Operation)**

Ensures that all data are signed and verified according to approved standards. SF.CO.CRYPTOGRAPHIC\_FUNCTIONS implements RSA, SHA512-RSA, ECDSA, ECDSA-SHA1 algorithms. SF.SL uses FCS\_COP.1/VERIF ECKDSA-sha256 to check executable code integrity during software update (ECC 256 bits key pair).

**FCS\_COP.1/(MESSAGE AUTHENTICATION /VERIFY) (Cryptographic Operation)**

Ensures that all the messages are authenticated and verified according to approved standards. SF.CO.CRYPTOGRAPHIC\_FUNCTIONS implements HMAC MD5, HMAC SHA-1, SHA256, SHA384, SHA512, DES MAC, DES3 MAC, AES MAC, RSA, MD5-RSA, SHA1-RSA, SHA256-RSA, SHA384 algorithms.

**FCS\_COP.1/(ENCRYPT /DECRYPT) (Cryptographic Operation)**

Ensures that all data are encrypt and decrypt according approved standards. SF.CO. CRYPTOGRAPHIC\_FUNCTIONS implements RSA and AES algorithms. SF.SL uses FCS\_COP.1/DECRYPT to decrypt executable code during software update.

**FCS\_COP.1/DIGEST (Cryptographic Operation)**

Ensures that all data are hashed according approved standards. SF.CO. CRYPTOGRAPHIC\_FUNCTIONS implements MD5, SHA-1and SHA-2 algorithms.

**FCS\_COP.1/(WRAP /UNWRAP) (Cryptographic Operation)**

Ensures that all keys are wrap and unwrap according approved standards. SF.CO. CRYPTOGRAPHIC\_FUNCTIONS implements AES and RSA algorithms.

**FCS\_COP.1/BACKUP\_ENC and FCS\_COP.1/BACKUP\_INT (Cryptographic Operation)**

Establishes confidentiality and integrity of backup data. SF.BACKUP.DATA\_PROTECTION implements standard algorithms with approved key sizes.

**FCS\_RND.1 (Quality metrics for random numbers)**

Ensures that keys are generated according a quality random number generator. The hardware based random number generator described in SF.CO.KEY\_GENERATION meets the requirement.

**FDP\_ACC.1/CRYPTO, FDP\_ACC.1/AUDIT, FDP\_ACC.1/BACKUP, FDP\_ACC.1/LOAD, FDP\_ACC.1/CONFIG et FDP\_ACC.1/DEPERSONALIZATION**

Guarantees the application of access control SFPs. This component is met with SF.ACCESS\_CONTROL.



**FDP\_ACF.1/CRYPTO, FDP\_ACF.1/AUDIT, FDP\_ACF.1/BACKUP,  
FDP\_ACF.1/LOAD, FDP\_ACF.1/CONFIG et  
FDP\_ACF.1/DEPERSONALIZATION**

Describes the rules of the access control policy. SF.AUTHENTICATION.ROLES defines the roles to access cryptographic functions, backup functions, secure loading, configuration, depersonalization and audit functions.

**FDP\_BKP.1 (Backup and recovery)**

Ensures that a backup function is available and that the recovery function can restore the initial state of the TOE. SF.BACKUP.COMMAND defines the backup command (global mode and unique mode) and SF.BACKUP.DATA\_PROTECTION ensures protection of sensitive data for further recovery.

**FDP\_ETC.1 (Export of user data without security attributes)**

Defines function to backup data without security attributes. SF.BACKUP.DATA\_PROTECTION defines the backup policy to meet this component.

**FDP\_RIP.1 (Subset residual information protection)**

Ensures that keys and authentication data are no longer available after TOE de allocation data. SF.SI and SF.SM.DEPERSONALIZATION ensure that the TOE uninstallation clears the secure memory and prohibits any access to authentication process.

**FDP\_SDI.2 (Stored data integrity monitoring and action)**

Ensures that stored user data are protected from disclosure. SF.SM.TESTS performs tests to control stored keys prior to any utilisation and embedded code integrity at start-up.

**FIA\_AFL.1 (Authentication Failure Handling)**

Ensures that human users who are not Authorised Administrators cannot endlessly attempt to authenticate. SF.AUTHENTICATION.POLICY imposes that after 5 failures the smartcard becomes unusable and that after user is unable from that point on to authenticate.

**FIA\_ATD.1 (User Attribute Definition)**

Exists to provide attributes to distinguish Authorised Administrators from one another. SF.AUTHENTICATION.ROLES defines roles and identities for crypto-officer security officer, auditor and HSM auditor.

**FIA\_SOS.1 (Verification of secrets)**

Ensures high strength verification of authentication secrets. SF.AUTHENTICATION.POLICY defines the secure mechanism implemented to meet this component.

**FIA\_UAU.1 (Timing of Authentication)**

Ensures that the user is authenticated before any action is allowed by the TSF. SF.SM.TESTS guaranties that, at power on, all TOE security elements are tested before allowing any other action. SF.SM.ALARMS guaranties that the TOE is unavailable after failure detection. SF.AUTHENTICATION.POLICY guaranties strong authentication before performing any other action.

**FIA\_UID.1 (Timing of Identification)**

Ensures that the user's identity is identified to the TOE before anything occurs on behalf of the Authorised Administrator. SF.SM.TESTS guaranties that, at power on, all TOE security elements are tested before allowing any other action. SF.SM.ALARMS guaranties that the TOE is unavailable after failure detection.

### **FMT\_MOF.1 (Management of Security Functions Behaviour)**

Ensures that the TSF restricts the ability to modify the behaviour of loading and backup functions to an Authorised Administrator. SF.ACCESS\_CONTROL implements such a control with several control options and SF.AUTHENTICATION\_ROLES defines administrator (crypto officer and security officer) role.

### **FMT\_MSA.1/ROLE\_CRYPT0 and FMT\_MSA.1/ROLE\_AUDIT (Management of Security Attributes)**

Ensures that the TSF restricts the ability to query, delete, and modify the security attributes. SF. AUTHENTICATION\_ROLES imposes the policy to manage the different security attributes relative to roles user, crypto officer, security officer, auditor and HSM auditor.

### **FMT\_MSA.2 (Secure Security Attributes)**

Guaranties valid values for security attributes. SF. AUTHENTICATION\_ROLES affects secure values to the different security attributes relative to roles user, crypto officer, security officer, auditor and HSM auditor.

### **FMP\_MSA.3 (Static Attribute initialisation)**

Guaranties valid default values for security attributes. SF. AUTHENTICATION\_ROLES imposes the default value policy to manage the different security attributes relative to roles user (crypto-user), security officer, auditor, crypto officer and HSM auditor.

### **FMT\_MTD.1/ACCESS\_CONTROL, FMT\_MTD.1/USER\_CRYPT0, FMT\_MTD.1/USER\_AUDIT, FMT\_MTD.1/RAD and FMT\_MTD.1/AUDIT (Management of TSF Data)**

Ensures that the TSF restricts the ability to handle TSF data to Authorised users. This component is met with SF.ACCESS\_CONTROL (access control) and SF. AUTHENTICATION\_ROLES (role definition).

### **FMT\_SMF.1 (Specification of Management Functions)**

Defines the security management functions performing by the TSF. SF. AUTHENTICATION\_ROLES performs the user management function including audit data management, SF.ACCESS\_CONTROL performs management of functions and TSF data.

### **FMT\_SMR.1 (Security Roles)**

Ensures that each of the FMT components depends on the assignment of a user to the Authorised role. SF. AUTHENTICATION\_ROLES lists and defines the TOE roles.

### **FPR\_UNO.1 (Unobservability)**

Guaranties the protections needed to avoid information flow. This component is covered by SF.SM.HARDWARE.

### **FPT\_FLS.1 (Failure with preservation of secure state)**

Ensures that all sensitive data are not available after test failure detection. After failure detection, SF.SM.ALARMS halts all processors.

### **FPT\_ITC.1 (Inter-TSF confidentiality during transmission)**

Defines the rules to protect confidentiality of backup data during transmission outside the TOE. SF.BACKUP.DATA\_PROTECTION ensures confidentiality of all backup data during transmission towards another IT product.

### **FPT\_ITI.1 (Inter-TSF detection of modification)**

Defines the rules to protect integrity of backup data during transmission outside the TOE. SF.BACKUP.DATA\_PROTECTION ensures integrity protection of all backup data during transmission towards another IT product. Any modification during key restore process causes an event in the audit file and an abort of backup command.

### **FPT\_ITT.1 (Basic internal data transfer protection)**

Ensures that the cryptographic keys are protected outside the cryptographic module. This component is met with SF.SM.KEYS.

### **FPT\_PHP.2 (Notification of physical attack)**

Ensures that any physical tampering is detectable. After tampering detection SF.SM.ALARMS halts all processors and clears the whole secure memory.

### **FPT\_PHP.3 (Resistance of physical attack)**

Ensures that all the sensitive data are protected from physical attacks. SF.SM.HARDWARE guarantees various hardware protection including power voltage monitoring, temperature monitoring, TOE embedded in a hard opaque potting material and intrusion detection. SF.SM.ALARMS defined alarm response to physical attacks.

### **FPT\_RCV.1 (Manual recovery)**

Ensures human intervention after failure. SF.SI ensures that the TOE must be returned to Bull logistic centre to be personalized in order to assure service continuity.

### **FPT\_STM.1 (Reliable Time Stamps)**

Was included because FAU\_GEN.1 depends on having the date and time accurately recorded in the audit records. SF.SI ensures correct date and time setting at installation phase.

### **FPT\_TST.1 (TSF Testing)**

Ensure the correct functioning and the integrity of all TSF code and data.

SF.SM.TESTS implements software integrity tests, keys integrity tests, cryptographic algorithms tests, random number tests. SF.CO.KEY\_GENERATION performs Pair-wise consistency tests for public and private keys.

### **FTP\_TRP.1/TOE (Trusted Path)**

Ensures that authentication process is performed through a secure path logically and physically independent from user data path.

SF.AUTHENTICATION.TRUSTED\_PATH guarantees that any authentication takes place via a smart card reader housed into the appliance, which is directly linked (trusted path) to the TSF.

## 9.5 Dependency Rationale

### 9.5.1 Functional and Assurance Requirements Dependencies

Requirement	CC-required Dependencies	Remark
<b>Functional Requirements for the TOE</b>		
FAU_GEN.1	FPT_STM.1	dependency is not satisfied by the PP prTS 419221-2 (see justification in section au § 9.5.2)
FAU_GEN.2	FAU_GEN.1, FIA_UID.1	
FAU_STG.2/TOE	FAU_GEN.1	
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1], FCS_CKM.4	
FCS_CKM.2/backup_keys	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Back-up key material is provided by the TOE (FCS_CKM.1/backup)
FCS_CKM.2/Other_keys	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1 dependency is not satisfied (the keys are entered through the PKCS#11AP)
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	
FCS_COP.1/BACKUP_ENC	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Back-up key material is provided by the TOE (FCS_CKM.1/backup)
FCS_COP.1/BACKUP_INT	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	Back-up key material is provided by the TOE (FCS_CKM.1/backup)
FCS_COP.1/all iterations	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	
FDP_ACC.1/BACKUP	FDP_ACF.1/BACKUP	
FDP_ACC.1/AUDIT	FDP_ACF.1/AUDIT	
FDP_ACC.1/CRYPTO	FDP_ACF.1/CRYPTO	
FDP_ACC.1/LOAD	FDP_ACF.1/LOAD	
FDP_ACC.1/CONFIG	FDP_ACF.1/CONFIG	
FDP_ACC.1/DEPERSONALIZATION	FDP_ACF.1/DEPERSONALIZATION	
FDP_ACF.1/BACKUP	FDP_ACC.1/BACKUP, FMT_MSA.3	
FDP_ACF.1/AUDIT	FDP_ACC.1/AUDIT, FMT_MSA.3	
FDP_ACF.1/CRYPTO	FDP_ACC.1/CRYPTO, FMT_MSA.3	
FDP_ACF.1/LOAD	FDP_ACC.1/LOAD, FMT_MSA.3	
FDP_ACF.1/CONFIG	FDP_ACC.1/CONFIG, FMT_MSA.3	
FDP_ACF.1/DEPERSONALIZATION	FDP_ACC.1/DEPERSONALIZATION	

Requirement	CC-required Dependencies	Remark
FDP_BKP.1	[FCS_CKM.1/backup or FCS_CKM.2/backup_keys or FDP_ITC.1], FCS_COP.1/BACKUP_ENC, FCS_COP.1/BACKUP_INT	
FDP_ETC.1	[FDP_ACC.1/CRYPTO, FDP_ACC.1/BACKUP, FDP_ACC.1/AUDIT, FDP_IFC.1/CRYPTO or FDP_IFC.1/ BACKUP]	
FIA_AFL.1	FIA_UAU.1	
FIA_UAU.1	FIA_UID.1	
FIA_UID.1	(no dependency)	
FMT_MOF.1	FMT_SMR.1, FMT_SMF.1	
FMT_MSA.1/ ROLE_CRYPTO	[FDP_ACC.1/BACKUP, FDP_ACC.1/CRYPTO, FDP_ACC.1/LOAD or FDP_IFC.1], FMT_SMR.1, FMT_SMF.1	
FMT_MSA.1/ ROLE_AUDIT	[FDP_ACC.1/AUDIT or FDP_IFC.1], FMT_SMR.1, FMT_SMF.1	
FMT_MSA.2	[FDP_ACC.1/AUDIT, FDP_ACC.1/BACKUP, FDP_ACC.1/CRYPTO, FDP_ACC.1/LOAD or FDP_IFC.1], FMT_MSA.1/ROLE_AUDIT, FMT_MSA.1/ROLE_CRYPTO, FMT_SMR.1	
FMT_MSA.3	FMT_MSA.1/ROLE_AUDIT, FMT_MSA.1/ROLE_CRYPTO, FMT_SMR.1	
FMT_MTD.1/AUDIT	FMT_SMR.1, FMT_SMF.1	
FMT_MTD.1/ ACCESS_CONTROL	FMT_SMR.1, FMT_SMF.1	
FMT_MTD.1/ USER_CRYPTO	FMT_SMR.1, FMT_SMF.1	
FMT_MTD.1/ USER_AUDIT	FMT_SMR.1, FMT_SMF.1	
FMT_MTD.1/RAD	FMT_SMR.1, FMT_SMF.1	
FMT_SMF.1	(no dependency)	
FMT_SMR.1	FIA_UID.1	
FPR_UNO.1	(no dependency)	
FPT_FLS.1	(no dependency)	
FPT_ITI.1	(no dependency)	
FPT_ITT.1	(no dependency)	
FPT_PHP.2	FMT_MOF.1	dependency is not satisfied by the PP prTS 419221-2 (see justification in section au § 9.5.2)

Requirement	CC-required Dependencies	Remark
FPT_PHP.3	(no dependency)	
FPT_RCV.1	AGD_OPE.1	
FPT_STM.1	(no dependency)	
FPT_TST.1	(no dependency)	
FTP_TRP.1	(no dependency)	

**Tableau 9-5 – Functional and Assurance Requirements Dependencies.**

Assurance Requirements		
ADV_ARC.1	ADV_FSP.1, ADV_TDS.1	ADV_FSP.4 is hierarchical to ADV_FSP.1 ADV_TDS.3 is hierarchical to ADV_TDS.1
ADV_FSP.4	ADV_TDS.1	ADV_TDS.3 is hierarchical to ADV_TDS.1
ADV_IMP.2	ADV_TDS.3, ALC_TAT.1, ALC_CMC.5	
ADV_TDS.3	ADV_FSP.4	
AGD_OPE.1	ADV_FSP.1	ADV_FSP.2 is hierarchical to ADV_FSP.1
AGD_PRE.1	(no dependency)	
ALC_CMC.5	ALC_CMS.1, ALC_DVS.2, ALC_LCD.1	ALC_CMS.4 is hierarchical to ALC_CMS.1
ALC_CMS.4	(no dependency)	
ALC_DEL.1	(no dependency)	
ALC_DVS.2	(no dependency)	
ALC_FLR.3	(no dependency)	
ALC_LCD.1	(no dependency)	
ALC_TAT.1	ADV_IMP.1	ADV_IMP.2 is hierarchical to ADV_IMP.1
ATE_COV.2	ADV_FSP.2, ATE_FUN.1	ADV_FSP.4 is hierarchical to ADV_FSP.1
ATE_DPT.2	ADV_ARC.1, ADV_TDS.3, ATE_FUN.1	
ATE_FUN.1	ATE_COV.1	ATE_COV.2 is hierarchical to ATE_COV.1
ATE_IND.2	ADV_FSP.2, AGD_OPE.1, AGD_PRE.1, ATE_COV.1, ATE_FUN.1	ADV_FSP.4 is hierarchical to ADV_FSP.1 ATE_COV.2 is hierarchical to ATE_COV.1
AVA_VAN.5	ADV_ARC.1, ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1	ADV_IMP.2 is hierarchical to ADV_IMP.1

**Tableau 9-6 - Functional and Assurance Requirements Dependencies**

## 9.5.2 Justification of unsupported Dependencies

Component	Justification for not including	Remark
<b>Security Functional Requirements for the TOE</b>		
FAU_GEN.1	FPT_STM.1	FPT_STM.1 must be included if FAU_GEN.1 uses a reliable time stamp.
FCS_COP.1/X	FCS_CKM.1	The backup key material will be provided by the TOE environment (as required by O.ENV_RECOVERY)
FPT_PHP.2	FMT_MOF.1	FPT_PHP.2 informs the local user about detected tampering attempt.

**Tableau 9-7 - Justification of Unsupported Dependencies**



## 9.6 Rationale for Assurance Level 4 Augmented

The assurance level for this protection profile is **EAL4 augmented**.

The TOE described in this security target is just such a product. Augmentation results from the selection of:

- ADV\_IMP.2 (Implementation of the TSF),
- ALC\_CMC.5 (Advanced Support),
- ALC\_DVS.2 (Development Security),
- ALC\_FLR.3 (Systematic Flaw Remediation),
- AVA\_VAN.5 (Vulnerability Analysis).

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialised processes and practices. It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions.

The TOE described in this security target is just such a product. Augmentation results from the selection of **AVA\_VAN.5**.

The TOE generates uses and manages the most sensitive data of the organisation – the R.USER\_DATA. Any loss of confidentiality or integrity of the R.USER\_DATA threatens the security of the certificates signed with this R.USER\_DATA and therefore the security of all signatures created with the SCD which correspond to the certificates.

The cryptographic security of the R.USER\_DATA / R.USER\_PUB\_KEYS pair generation and the signing with the R.USER\_DATA can be ensured only by the TOE itself. The TOE shall be free of any covert channel which might compromise the R.USER\_DATA. The TOE environment shall support the TOE in R.USER\_DATA protection against physical and some other attacks but cannot make up for TOE security

### 9.6.1 AVA\_VAN.5 Advanced methodical vulnerability analysis

The TOE environment must assist the TOE in the protection of R.USER\_DATA against physical or other types of attacks. It cannot however ensure the security of the TOE.

The complex protection of the R.USER\_DATA requires a systematic and complete vulnerability analysis by SAR AVA\_VAN.5. The TOE protecting the R.USER\_DATA as most valuable asset shall be shown to be highly resistant to penetration attacks.

### 9.6.2 ADV\_IMP.2 Complete mapping of the implementation representation of the TSF

The TOE implementation shall be entirely under control and well documented. This will help the evaluators to attain a high level of confidence on the security and reliability of the implementation.

### **9.6.3 ALC\_DVS.2 Sufficiency of security measures**

The development, integration and validation phases shall be performed in a highly secure environment. This security includes the personnel, premises, and procedures providing integrity and confidentiality to the TOE.

### **9.6.4 ALC\_FLR.3 Systematic Flaw Remediation**

The flaw management shall be efficient. Users must be informed as soon as possible in case of vulnerability discovery and a mitigation or correction shall be deployed quickly. Users must also have the possibility to submit questions or problems easily.

**END OF DOCUMENT**