TRUSTWAY

TrustWay PROTECCIO

Security Target

Nom du Document	PCA4_0003_CIB_Security Target_EN
Version	1.5
Statut	Final version
Date	October 14th 2015
Niveau de classification	NP
Préparé par	Liliana CABALANTTI
Validé par	René MARTIN
Approuvé par	ANSSI

Table of contents

Chapitre 1. Introduction 9

1.1 Document presentation 9

1.2 References 9

1.3 Glossary 11

Chapitre 2. ST introduction 13

2.1 ST identification 13

2.2 ST overview 13

2.3 CC conformance 14

2.4 PP conformance 14

2.5 Qualification conformance 14

Chapitre 3. TOE description 15

3.1 Product type 15

3.2 Architecture 17

3.3 Life cycle 20

3.4 TOE boundary 21

3.5 TOE functionalities 22

3.5.1 Cryptographic operations 22

3.5.2 Cryptographic algorithms 22

3.5.3 Key sizes supported by the TOE 22

3.5.4 Key management 23

3.5.5 TOE roles 23

3.5.5.1 Security Officer 23

3.5.5.2 Auditor 24

3.5.5.3 HSM Security Officer 24

3.5.5.4 HSM Auditor 24

3.5.5.5 User 24

3.5.6 Correspondence between PP and TrustWay Proteccio roles 24

3.5.7 Administration 26

3.5.8 TOE installation 26

3.5.9 TOE personalization 27

3.5.10 CIK activation 27

3.5.11 Test of critical fonctions 27

3.5.11.1 Black key decryption 27

3.5.11.2 Periodic tests and fault management 27

3.6 Protect the network link between applications and TOE 28

3.7 TOE usage 28

Chapitre 4. TOE Security Environment 31

4.1 Assets to protect 31
4.1.1 TOE services 31
4.1.2 TOE internal data 31
4.1.3 Data shared between the TOE and its environment 32
4.2 Threats 32
4.3 Organisational Security Policies 36
4.4 Assumptions 36

Chapitre 5. Security Objectives 38

5.1 Security Objectives for the TOE 38
5.2 Security Objectives for the Environment 41

Chapitre 6. Security Requirements 44

6.1 Security Functional Requirements 44
6.1.1 Security audit (FAU) 44
6.1.1.1 FAU_GEN.1 Audit data generation 44
6.1.1.2 FAU_GEN.2 User identity association 45
6.1.1.3 FAU_STG.2 (TOE) Guarantees of audit data availability 45
6.1.2 Cryptographic support (FCS) 46
6.1.2.1 FCS_CKM.1 Cryptographic key generation 46
6.1.2.2 FCS_CKM.1 (backup) Cryptographic key generation 46
6.1.2.3 FCS_CKM.2 (backup_keys) Cryptographic key distribution 46
6.1.2.4 FCS_CKM.2 (Other_keys) Cryptographic key distribution 47
6.1.2.5 FCS_CKM.4 Cryptographic key destruction 47
6.1.2.6 FCS_COP.1 (SIGN/VERIFY) Cryptographic operation 48
6.1.2.7 FCS_COP.1 (MESSAGE AUTHENTICATION/VERIFY) Cryptographic operation 48
6.1.2.8 FCS_COP.1 (DIGEST) Cryptographic operation 49
6.1.2.9 FCS_COP.1 (WRAP/UNWRAP) Cryptographic operation 50
6.1.2.10 FCS_COP.1 (BACKUP_ENC) Cryptographic operation 50
6.1.2.11 FCS_COP.1 (BACKUP_INT) Cryptographic operation 50
6.1.2.12 FCS_RND.1 Quality metrics for random numbers 50
6.1.3 User data protection (FDP) 50
6.1.3.1 FDP_ACC.1 (CRYPTO) Subset access control 50
6.1.3.2 FDP_ACC.1 (CONFIG) Subset access control 50
6.1.3.3 FDP_ACC.1 (AUDIT) Subset access control 51
6.1.3.4 FDP_ACC.1 (BACKUP) Subset access control 51
6.1.3.5 FDP_ACC.1 (LOAD) Subset access control 51
6.1.3.6 FDP_ACC.1 (DEPERSONALIZATION) Subset access control 51
6.1.3.7 FDP_ACF.1 (CRYPTO) Security attribute based access control 51
6.1.3.8 FDP_ACF.1 (CONFIG) Security attribute based access control 52
6.1.3.9 FDP_ACF.1 (AUDIT) Security attribute based access control 52
6.1.3.10 FDP_ACF.1 (BACKUP) Security attribute based access control 53
6.1.3.11 FDP_ACF.1 (LOAD) Security attribute based access control 53
6.1.3.12 FDP_ACF.1 (DEPERSONALIZATION) Security attribute based access control 54

6.1.3.13 FDP_BKP.1 Backup and recovery 54
6.1.3.14 FDP_ETC.1 Export of user data without security attributes 55
6.1.3.15 FDP_RIP.1 Subset residual information protection 55
6.1.3.16 FDP_SDI.2 Stored data integrity monitoring and action 55
6.1.4 Identification and authentication (FIA) 56
6.1.4.1 FIA_AFL.1 Authentication failure handling 56
6.1.4.2 FIA_ATD.1 User attribute definition 56
6.1.4.3 FIA_SOS.1 Verification of secrets 56
6.1.4.4 FIA_UAU.1 Timing of authentication 56
6.1.4.5 FIA_UID.1 Timing of identification 56
6.1.5 Security management (FMT) 57
6.1.5.1 FMT_MOF.1 Management of security functions behaviour 57
6.1.5.2 FMT_MSA.1 (ROLE_CRYPTO) Management of security attributes 57
6.1.5.3 FMT_MSA.1 (ROLE_AUDIT) Management of security attributes 57
6.1.5.4 FMT_MSA.2 Secure security attributes 57
6.1.5.5 FMT_MSA.3 Static attribute initialisation 57
6.1.5.6 FMT_MTD.1 (ACCESS_CONTROL) Management of TSF data 57
6.1.5.7 FMT_MTD.1 (USER_CRYPTO) Management of TSF data 58
6.1.5.8 FMT_MTD.1 (USER_AUDIT) Management of TSF data 58
6.1.5.9 FMT_MTD.1 (RAD) Management of TSF data 58
6.1.5.10 FMT_MTD.1 (AUDIT) Management of TSF data 58
6.1.5.11 FMT_SMF.1 Specification of Management Functions 58
6.1.5.12 FMT_SMR.1 Security roles 58
6.1.6 Privacy (FPR) 59
6.1.6.1 FPR_UNO.1 (CRYPTO) Unobservability 59
6.1.6.2 FPR_UNO.1 (BACKUP) Unobservability 59
6.1.7 Protection of the TOE Security Functions (FPT) 60
6.1.7.1 FPT_FLS.1 Failure with preservation of secure state 60
6.1.7.2 FPT_ITC.1 Inter-TSF confidentiality during transmission 60
6.1.7.3 FPT_ITI.1 Inter-TSF detection of modification 61
6.1.7.4 FPT_ITT.1 Basic internal TSF data transfer protection 61
6.1.7.5 FPT_PHP.2 Notification of physical attack 61
6.1.7.6 FPT_PHP.3 Resistance to physical attack 62
6.1.7.7 FPT_RCV.1 Manual recovery 62
6.1.7.8 FPT_STM.1 Time stamps 62
6.1.7.9 FPT_TST.1 TSF testing 62
6.1.8 Trusted path (FTP) 64
6.1.8.1 FTP_TRP.1 (TOE) Trusted path 64

6.2 TOE Security Assurance Requirements 65

Chapitre 7. TOE summary specification 66

7.1 TOE security functions 66
7.1.1 Audit Data Generation (SF.AUDIT) 66
7.1.1.1 SF.AUDIT.EVENTS 66
7.1.1.2 SF.AUDIT.FILE 66
7.1.2 Authentication (SF.AUTHENTICATION) 66
7.1.2.1 SF.AUTHENTICATION.ROLES 66
7.1.2.2 SF.AUTHENTICATION.TRUSTED_PATH 67
7.1.2.3 SF.AUTHENTICATION.POLICY 68

7.1.3 Access control (SF.ACCESS_CONTROL) ... 69
7.1.4 HSM management ... 69
7.1.4.1 Secure installation (SF.SI) ... 69
7.1.5 Cryptographic operations (SF.CO) ... 69
7.1.5.1 SF.CO.KEY_GENERATION ... 69
7.1.5.2 SF.CO.KEY_DESTRUCTION ... 70
7.1.5.3 SF.CO.CRYPTOGRAPHIC_FUNCTIONS ... 70
7.1.6 Secure loading (SF.SL) ... 71
7.1.6.1 General mechanism ... 71
7.1.7 Security mechanisms (SF.SM) ... 71
7.1.7.1 SF.SM.HARDWARE ... 71
7.1.7.2 SF.SM.KEYS ... 72
7.1.7.3 SF.SM.TESTS ... 72
7.1.7.4 SF.SM.ALARMS ... 73
7.1.7.5 SF.SM.DEPERSONALIZATION ... 73
7.1.8 Backup and Recovery (SF.BACKUP) ... 74
7.1.8.1 SF.BACKUP.COMMAND ... 74
7.1.8.2 SF.BACKUP.AUDIT ... 74
7.1.8.3 SF.BACKUP.DATA_PROTECTION ... 74

Chapitre 8. PP claims ... 75

8.1 PP référence ... 75

8.2 PP addition ... 75
8.2.1 Threats ... 75
8.2.2 Assumptions ... 75
8.2.3 Security objectives ... 76
8.2.4 Security objectives for the environment ... 76
8.2.5 Security Functional Requirements ... 76

8.3 Configuration recommandations ... 77
8.3.1 Key generation configuration (KGC virtual HSM) ... 78
8.3.1.1 Configuration ... 78
8.3.1.2 Key generation ... 78
8.3.2 Key usage configuration (User virtual HSM) ... 79
8.3.2.1 Configuration ... 79

Chapitre 9. Rationale ... 80

9.1 Introduction ... 80

9.2 Security Objectives Rationale ... 80
9.2.1 Security Objectives Coverage ... 80
9.2.2 Security Objectives Sufficiency ... 82
9.2.2.1 Threats ... 82
9.2.2.2 Organisational Security Policies ... 86
9.2.2.3 Assumptions ... 86

9.3 Security Requirements Rationale ... 88
9.3.1 Security Requirement Coverage ... 88
9.3.2 Security Requirements Sufficiency ... 89
9.3.2.1 TOE Security Requirements Sufficiency ... 89

9.4 TOE Summary Specification Rationale 93
9.4.1 TOE Security functions Coverage 93
9.4.2 TOE Security functions Sufficiency 95

9.5 Dependency Rationale 100
9.5.1 Functional and Assurance Requirements Dependencies 100
9.5.2 Justification of Unsupported Dependencies 104

9.6 Rationale for Assurance Level 4 Augmented 104
9.6.1 AVA_VAN.5 Advanced methodical vulnerability analysis 105

Chapitre 10. Appendix A – Acronyms 106

Chapitre 1. Introduction

1.1 Document presentation

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. http://www.ssi.gouv.fr/IMG/pdf/RGS_qualif_renforcee_version_1-0.pdf

2. **Mécanismes cryptographiques : Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques de niveau de robustesse standard ou durci** Version 2.00, June 18th 2012 n° 1635/ANSSI

3. **Gestion des clés cryptographiques : Règles et recommendations concernant la gestion des clés utilisées dans des mécanismes cryptographiques de niveau de robustesse standard ou durci** Version 2.00, June 18th 2012, n° 1636/ANSSI

4. **Authentification : Règles et recommandations concernant les mécanismes d'authentification de niveau de robustesse renforcé** Version 0.13, April 3rd 2007, n° 730/SGDN/DCSSI/SDS

5. **Recommandation pour l'intégration des fonctions cryptographiques dans les systèmes d'information** n°501/SGDN/DCSSI/DR du 20 décembre 2002

6. **Common Criteria for Information Technology Security Evaluation, Part 1**: Introduction and General Model; Version 3.1, Revision 3, July 2009, CCMB-2006-09-001

7. **Common Criteria for Information Technology Security Evaluation, Part 2**: Security Functional Requirements; Version 3.1, Revision 3, July 2009, CCMB-2006-09-002

8. **Common Criteria for Information Technology Security Evaluation, Part 3**: Security Assurance Requirements; Version 3.1, Revision 3, July 2009, CCMB-2006-09-003

9. **CEN/ISSS WS/E-Sign; Area D1, CWA 14167-1**: Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures

10. **prEN 14167-2:2012** (Cryptographic Module for CSP Signing Operations with Backup – Protection Profile).

11. **prTS 14167-3 :2011** (Cryptographic Module for CSP key generation services), en ce qui concerne certaines des exigences de sécurité.

12. **DIRECTIVE 1999/93/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL** of 13 December 1999 on a Community framework for electronic signatures

13. **ETSI SR 002 176 - Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures** V1.1.1 (2003-03)

14. **European Telecommunications Standards Institute Technical Specification, ETSI TS 101462 Policy requirements for certification authorities issuing qualified certificates**, V1.1.1, 2000

15. **FIPS 46-3 Data Encryption Standard (DES)**
    October 25, 1999

16. **Recommendation for Random Number Generation Using Deterministic random bit Generators**
    Référence : NIST Special Publication 800-90 - March 2007

17. **FIPS PUB140-2 Security requirements for cryptographic modules**
    May 25, 2001

18. **FIPS 180-1Secure hash standard**
    April 17, 1995

19. **FIPS PUB 186-2 Digital Signature Standard**
    January 27,2000

20. **RFC 1321The MD5 Message-Digest algorithm**
    April 1992

21. **RFC 2104 HMAC: Keyed-Hashing for message Authentication**
    February 1997

22. **ISO 9797-1Message Authentication Codes (MACs) part 1 - Mechanisms using a block cipher**
    First edition 1999-12-15

23. **PKCS#1 RSA Cryptography Standard V2.1**
    June 14, 2002.

24. **PKCS#8 Private-Key information syntax standard**
    V1.2 November 1, 1993.

25. **PKCS#11Cryptographic Token interface standard**
    V2.11 November 2001

1.3 Glossary

Acronyme	Definition
ACP	Auxiliary Cryptographic Processor
AES	Advanced Encryption Standard
API	Application Programming Interface
Backup	Secure export and external storage of the CSP_SCD, the TSF data and the system data (backup data) sufficient to recreate the state of the TOE at the time the backup was created
CA	Certificate Authority
CC	Common Criteria
Certificate	Electronic attestation which links the SVD to a person and confirms the identity of that person
CIK	Crypto Ignition Key
CM	Cryptographic Module
CSP	Certification Service Provider
CSP_SCD	SCD which is used by the CSP, e.g. for the creation of advanced electronic signatures in qualified certificates or for signing certificate status information
CSP_SVD	SVD which corresponds to the CSP_SCD and which is used to verify the advanced electronic signature in the qualified certificate
DSA	Digital Signature Algorithm
DTBS	Data to be Signed
DTBS-representation	The data sent to the TOE for signing
Digital signature	Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery
EAL	Evaluation Assurance Level
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
GCP	General Cryptographic Processor
HMAC	Keyed-Hash Message Authentication Code
HSM	Hardware Security Module
IT	Information Technology

PKI	Public Key Infrastructure
PP	Protection Profile
Qualified certificate	Certificate which meets the requirements laid down in Annex I of the Directive and is provided by a CSP who fulfils the requirements laid down in Annex II of the Directive
RAD	Reference Authentication Data
RNG	Random Number Generator
RSA	Asymmetric algorithm developed by Rivest, Shamir and Adleman
SAR	Security Assurance Requirements
SCD	Signature Creation Data
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirementd
SHA	Secure HASH Algorithm
SCA	Application de création de signature
SCD	Signature-creation data
SSCD	Secure Signature Creation Device
ST	Security Target
SVD	Signature Verification Data
TDM	TrustWay Domain Management
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
user	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE
User data	Data created by and for the user that does not affect the operation of the TSF
VAD	Verification Authentication Data

Table 1-1. Glossary

Chapitre 2. ST introduction

2.1 ST identification

Title: Bull TrustWay HSM – Security Target

Author: Liliana CABALANTTI

ST version: 1.5, October 14th 2015

TOE version:

- TrustWay Proteccio EL : 1.05.03 - 76681604-004D/76681604-005 (hardware) X130 (Host system) V128 (Security module)
- TrustWay Proteccio HR : 1.05.03 - 76681610-004D/76681610-005 (hardware) X130 (Host system) V128 (Security module)

TOE commercial name: TrustWay Proteccio

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 [7] extended and Part 3 [8] augmented of Common Criteria v3.1.

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

- 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.

2.4 PP conformance

The ST is compliant with the Protection Profile prEN 14167-2:2012 (Cryptographic Module for CSP Signing Operations with Backup – Protection Profile).

The ST also includes most of the security requirements of Protection Profile prTS 14167-3:2011 (Cryptographic Module for CSP key generation services) for some of the security requirements.

2.5 Qualification conformance

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

- Cryptographic referential [2];
- Key management architecture referential [3];
- Authentication referential [4].

Chapitre 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 zeroization of plaintext key material and security parameters in the event a tamper signal is received.

The product is 100% developed and manufactured in Europe (including the cryptographic components).

There are two 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 I5 processor and an ARRIA2GX260 FPGA.

Bull HSM provides cryptographic functions for key generation, key storage, encryption and decryption, digital signature and verification used by application systems that provide cryptographic support functions.

The operating system supported into the appliance is Linux and the operating systems supported on the client side are Linux and Windows.

Bull HSM supports the PKCS#11 API which can be accessed from the client side through a TCP/IP network environment.

Figure 1: Bull TrustWay Proteccio

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) that complies with 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).

Bull HSM can be partitioned in virtual HSM while guaranteeing the 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

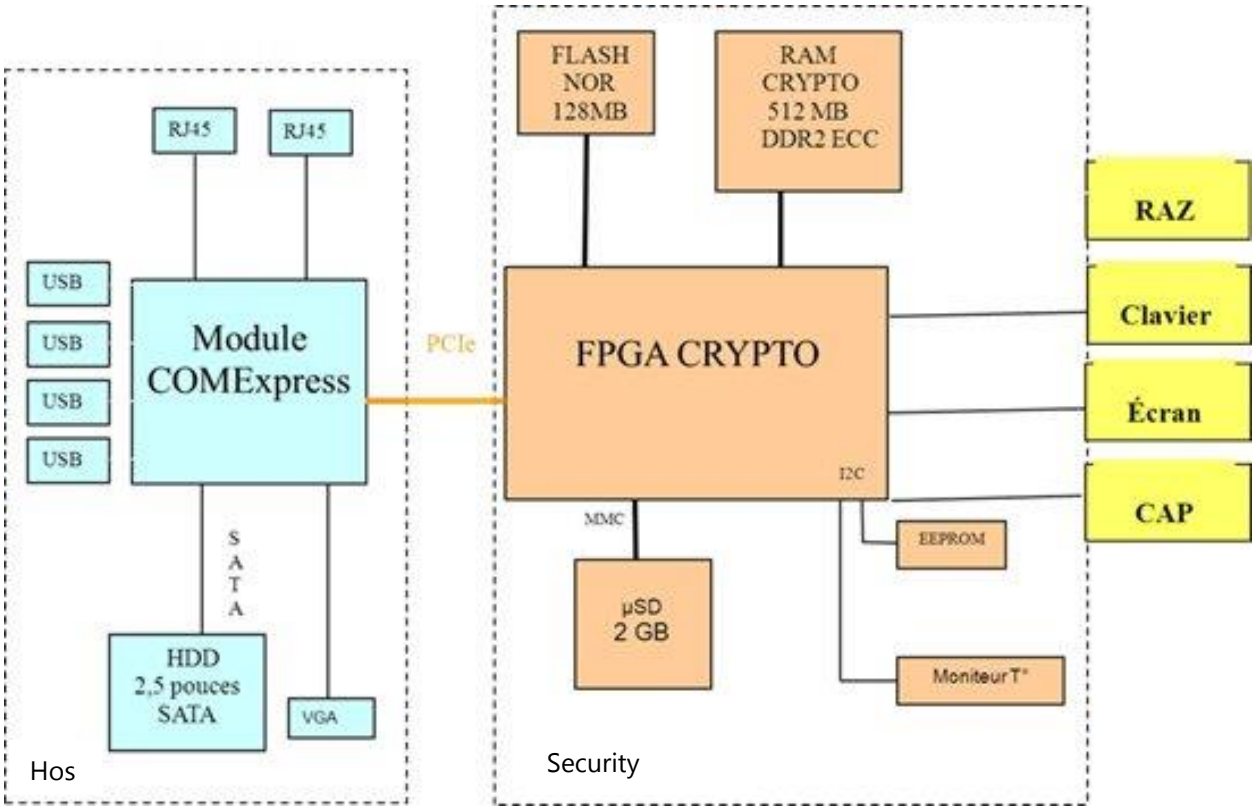

Figure 2: TrustWay Proteccio architecture

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 ETHERNET Gigabit interface 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
  - 512 MB of DDR2 RAM

  - 128 MB of NOR FLASH which contains the FPGA bitstream and the cryptographic processor code
  - 2 GB of 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 3 : TrustWay Proteccio front view

Rear:

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

Figure 4 : TrustWay Proteccio rear view

The Smart card used is the IDeal Citiz from Morpho (certified EAL5+, Certification Report ANSSI-CC-2010/56, Maintenance Report ANSSI-CC-2011/63-M01).

3.3 Life cycle

TrustWay Proteccio life cycle can be divided into 9 phases:

Phase		Phase Responsibility	Phase Environment
Phase 1	Development	The development team (Bull) is in charge of the hardware design and the embedded software development and signing Bull Les Clayes-sous-Bois	This phase is executed into the development environment (under the responsibility of the developer)
Phase 2	Software signing	All the signatures use elliptic curves according to the asymmetric mechanism ECKCDSA. Bull Les Clayes-sous-Bois	These phases are executed in BULL site (under the responsibility of the developer)
Phase 3	Manufacturing	The HSM manufacturing process is performed by subcontractors: • Manufacture of printed circuit motherboard (SOMACIS), • Enclosure manufacturing (ATOS), • Assembly, programming (manufacturing specific version) and test of components and integration into the enclosure (ASTEEL), • Repeat the tests into the enclosure (ASTEEL).	Production
Phase 4	Test and preparation by Bull Angers	Bull Angers executes complementary robustness tests and prepares the HSM for the next phase, which will take place on order receipt	These phases are executed in BULL Angers site (under the responsibility of Bull Angers)

Phase 5	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.	
Phase 6	Delivery to the client	The TrustWay Proteccio is sent to the client.	
Phase 7	Personalization	Personalization by the client.	These phases are executed into the end user environment (under the responsibility of the end user)
Phase 8	Embedded software update	If needed, the end user (security officer) can update the HSM embedded software.	
Phase 9	TOE use	This last phase is executed by the end user.	

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

Upon detection of un intrusion attempt, the TOE must be returned to Bull logistic centre to be repersonalized (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:

- sign and verify functions
- encrypt and decrypt functions
- hash function
- wrap and unwrap 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:

- Symmetric encryption/decryption : AES, DES, 3DES, modes ECB and CBC ;
- Asymmetric encryption/decryption: RSA (RSA-PKCS, RSA-PKCS-PSS, RSA-PKCS-OAEP);
- Sign/Verify : RSA, MD5-RSA, SHA1-RSA, SHA256-RSA, SHA384-RSA, SHA512-RSA, ECDSA, ECDSA-SHA1 ;
- Message authentication/Vérification : HMAC MD5, HMAC SHA-1, HMAC SHA256, HMAC SHA384, HMAC SHA512, DES MAC, DES3 MAC, AES MAC ;
- Hash : SHA256, SHA384, SHA512, SHA-1, MD5.

The cryptographic operations are performed by the TOE user, in its “Key usage” configuration.

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)
- ECC : 192 à 521 bits

3.5.4 Key management

Key generation is performed by the TOE user, in its "Key generation" configuration.

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 zeroization.

3.5.5 TOE roles

The TOE supports the user categories (roles) described below.

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

User authentication is performed with a password. The login operation is executed by means of C_Login PKCS#11 fonction.

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

Similarly, the term auditor will be used for the auditor and the HSM auditor.

3.5.5.1 Security Officer

The Security Officer is authorized to execute the following functions:

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

3.5.5.2 Auditor

The Auditor is authorized 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.

3.5.5.3 HSM Security Officer

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

The HSM security officer is authorized to execute the following operations:

- Personalize the virtual HSM;
- Depersonalize the virtual HSM ;
- Create the 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.

3.5.5.4 HSM Auditor

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

The HSM Auditor is authorized to execute the following operations:

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

3.5.5.5 User

A virtual HSM must have one and only one user.

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

The user is authorized to perform cryptographic operations.

3.5.6 Correspondence between PP and TrustWay Proteccio roles

The TOE has several phases of operation, with roles clearly identified as to their privileges in each of these phases. The table below details the three phases of operation of the TOE (Installation, Key generation, Key usage), identifying the respective roles.

The use of the TOE in accordance with these phases assures the conformance to the security requirements of the Protection Profile.

Roles	Installation	Key generation	Key usage
Crypto-Officer in its role Security officer	• Create internal virtual HSM • Create the role HSM Security officer • Create the role HSM Auditor	• Create virtual HSMs	• Create virtual HSMs
Crypto-Officer in its role HSM Security officer		• Personalize the virtual HSM • Create the role Crypto-Officer • Choose the cryptographic configuration (« Key generation » configuration)	• Personalize the virtual HSM • Create the role Crypto-User • Choose the cryptographic configuration (« Key usage » configuration)
Crypto-Officer in its role User		• Cryptographic operations: generation and backup of keys	
Crypto-User in its role User			• Cryptographic operations: restore keys, use keys (cryptographic calculations),
Auditor in its role Auditor	• Read audit data generated (events log and security log)	• Read audit data generated (events log and security log) • Get the token status	• Read audit data generated (events log and security log) • Get the token status
Auditor in its role HSM Auditor		• Read audit data generated by the virtual HSM	• Read audit data generated by the virtual HSM

Table 3-1. Correspondence between PP and TrustWay Proteccio roles

Notes:

- The PP defines a « Crypto-officer » role. This role can be split as follows: « Master Security officer », « HSM Security officer » and « User » (in the specific configuration « Key generation »).
- The PP defines a « Crypto-user » role. This role is declined like « User » (in the specific configuration « Key usage »).

- The PP defines an « Auditor » role. This one can be split as follows « Auditor » and « HSM Auditor ».
- The virtual HSM configuration « Key generation » is the configuration of a virtual HSM allowing only key generation and backup (see §8.3).
- The virtual HSM configuration « Key usage » is the configuration of a virtual HSM allowing only key restoration and cryptographic operations using those keys (see §8.3).
- Both virtual HSMs (the one configured for key generation and the one configured for key usage) should not be used simultaneously and must be installed with the same set of installation smart cards.

3.5.7 Administration

Product administration is performed by a Java application that uses a smart card based authentication.

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 authorized to the 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 implemented either locally or remotely 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 using a dedicated local application, 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 uninstall imposes the Crypyo-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 fonctions

3.5.11.1 Black key decryption

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

- Key cryptographic integrity 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 Protect the network link between applications and TOE

The TOE uses version 1.0.1 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:

- Certificate generation using (RSA 2048 bits);
- Certificate signature: SHA256WithRSAEncryption.

The ciphersuite used by TLS is: DHE-RSA-AES256-SHA256.

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 unauthorized modification and for ensuring that the TOE services are only used in an authorized way.

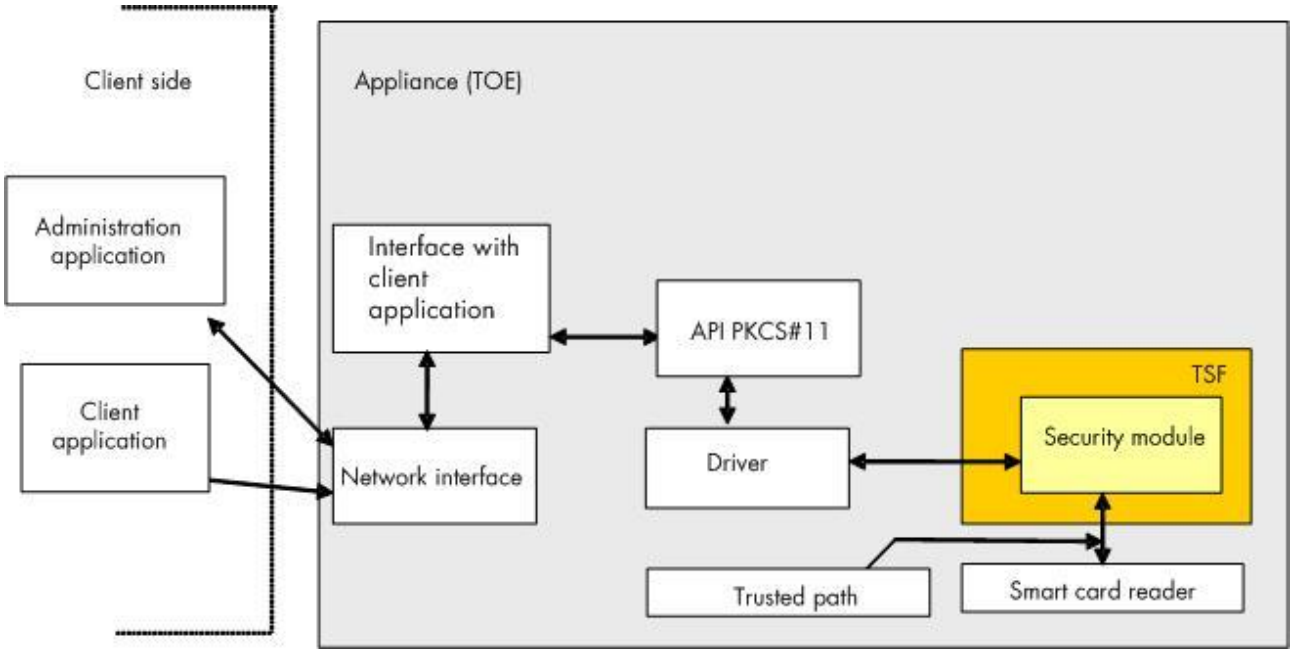

Figure 5: TOE and environment general overview

The TOE provides a physically and logically protected component for the performance of cryptographic functions for key generation, key storage, encryption and decryption, digital signature and verification used by application systems that provide cryptographic support functions such as a Certificate Authority/Certification Service Provider (CA/CSP).

Figure 5 shows the TOE in its appliance deployment configuration – as part of the Bull network-attached appliance.

The figure illustrates the relationship between the TOE and the appliance and also shows the division between the TSF and non-TSF portions of the TOE, specifically the FPGA crypto and the supporting smart card reader and library and driver software.

As shown in Figure 5, end-users will communicate with the client application, which in turn will call TOE services on behalf of the end-user. 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. It is 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 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 users in a secure way to the TOE.

The TOE supports backup and restoration of cryptographic objects, such as the CSP_SCD, 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 normally used as the cryptographic module for Bull appliance. As such, it 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 Kernel;
    - A specific driver and the PKCS #11 Cryptographic API software (provided as a Linux shared-object 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.
- A Java application for administration of the TOE.

The client application, provided as part of the TOE as a Windows dll or Linux-type shared-object library depending on the host platform configuration, runs within the appliance environment and provides the programming interface to the host software application, which normally acts as the user 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 VGA interface) allowing its local administration:

- By the Security Officer, after authentication:
    - Network configuration;
    - SSL configuration;
    - Embedded cryptographic software update;
    - System (Linux) software update.
- By the Auditor:
    - Allow PKCS#11 traces;
    - Execute diagnostic tests.

Chapitre 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 of CSP SCD and other cryptographic keys;
- Usage of cryptographic keys for cryptographic operations;
- Backup and export of TSF data;
- Secure update of the embedded software that implements the TOE services;
- 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.DTBS_REPRESENTATION (I)

Data To Be Signed Representation (DTBS_REPRESENTATION) means the data sent to the TOE for signing, and has to be protected in Integrity.

R.DTBSR_DS (I)

The result of the electronic signature over the R.DTBS_REPRESENTATION, produced by the cryptographic module, has to be protected in integrity.

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)

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

4.1.3 Data shared between the TOE and its environment

R.BACKUP (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.

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*".

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

T.INSECURE_CHANNEL – Sensitive data from applications sent by an authorized user to the TOE could be manipulated during the network transmission.

T.TRUSTED_PATH – Sensitive authentication data sent by unauthorized personnel to the TOE could be manipulated.

T.KEYS_DERIVE – Expands the thread T.CSP_SCD_DERIVE to include all key material handled by the TOE.

T.KEYS_DISCLOSE – Expands the thread T.CSP_SCD_DISCLOSE to include all key material handled by the TOE.

T.KEYS_ALTERATION – Expands the thread T.CSP_SCD_ALTERATION to include all key material handled 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_RESTORE

The attacker might manipulate R.BACKUP in the TOE environment in order to restore altered R.BACKUP that will alter R.USERMGMT_DATA or R.TSF_DATA.

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 or R.SERVICES.

For example, an attacker, using the TOE remote interface, may inject a malicious code (malware) into the TOE. Later on, this malware may compromise the confidentiality of the R.USER_DATA by exfiltrating its value from the TOE boundaries.

T.KEYS_ALTERATION

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

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

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.

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.

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.

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 HSM security officer may also result in a compromise of the security of the TOE. The backup of user data and TSF data might be lost.

T.INSECURE_INIT

The attacker, (e.g. unauthorised CSP personnel, authorised CSP personnel without using adequate organisational controls) may initialise the TOE with insecure R.TSF_DATA or R.USERMGMT_DATA.

T.MALFUNCTION

There is no active agent for this threat.

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

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, CSP personnel may misuse the TOE R.SERVICES to forge a CSP_SCD/SVD pair, resulting in R.USER_DATA and R. R.USER_PUB_KEYS used without proper authorisation.

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 with the R.USER_DATA forged qualified certificates or forged certificate status information).

T.PHYS_MANIPUL

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

- derive all or part of the R.USER_DATA (by side channel for example) or,
- manipulate the R.DTBS_REPRESENTATION 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 authorized 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 authorized 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.DTBSR_DS CSP digital signature in a way that is not detectable by the verifier of the signature).

4.3 Organisational Security Policies

P.ALGORITHMS

Only algorithms and algorithm parameter (e. g. key length) approved for being used for signature-creation by trustworthy systems shall be used to e.g. generate qualified certificates or to sign certificate status information.

The referentials for "renforcé" strength level edited by ANSSI ([2], [3], [4]) 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 CWA14167-2 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 authorized 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 CSP 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 CSP (Role Auditor/HSM Auditor) according to the audit procedure of the CSP.

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, the same as the administration application, are assumed as users of the TOE in the User or HSM Security officer role. Other users authorized for the TOE user services may be not be known to the TOE itself. The TOE environment performs identification and authentication for theses individual users and allows successfully authenticated users to use the client application as their agent for the 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

Authorized administrators are non-hostile, appropriately trained and follow all administrator guidance. In particular, authorized 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 authorized remote applications.

Chapitre 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
- Unsuccessful authentication (R.TSF_DATA)
- Modification of TOE management data (R.MNGT_DATA)
- Unsuccessful self test operations (R.TSF_DATA)
- Unsuccessful restore attempt
- Creation/suppression of virtual HSMs
- Personalization/depersonalization of virtual HSMs
- TOE software update
- Tamper detection event

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.

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.

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

Threats countered: T.BACKUP_RESTORE, T.BAD_SOFTWARE, T.MALFUNCTION, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.PHYS_MANIPUL

O. KEYS_SECURE

The confidentiality and integrity of the R.USER_DATA and other cryptographic keys shall be ensured during their whole lifetime.

The TOE shall ensure cryptographic secure R.USER_DATA and other cryptographic keys 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 “renforcé” 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 the conditions installation and maintenance.

Threats countered: T.BACKUP_RESTORE, T.BAD_SW, T.KEYS_ALTERATION, T.MALFUNCTION, T.PHYS_MANIPUL

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 HSM security officer or by default. Roles may also be predefined in the production or initialisation phase.

Threats countered: T.BACKUP_RESTORE, T.BAD_SW, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.MISUSE_OPERATION

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_RESTORE, 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 “renforcé” strength level [2].

Threats countered: T.BACKUP_RESTORE, 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 R.USER_DATA and
- Can not 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 “renforcé” strength level [2].

Threats countered: T.KEYS_DERIVE, T.KEYS_DISCLOSE, T.MISUSE_OPERATION, T.SIGNATURE_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.

Threats countered: T.BACKUP_RESTORE, T.BAD_SW, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.MISUSE_OPERATION

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.

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 crypto 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

5.2 Security Objectives for the Environment

The following security objectives relate to the TOE environment. This includes the client 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 – Authorized 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 authorized remote applications.

O.ENV_APPLICATION

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

The applications shall also perform the required user authentication and access control functions that cannot be performed within the TOE.

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

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 will ensure the confidentiality and integrity of the 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 authorized 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

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

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 authorized remote applications.

Chapitre 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 [7]. Some security functional requirements represent extensions to [CC2], with a reasoning given in section 6.5. Operations for assignment, selection and refinement have been made. Operations not performed in this PP are identified in order to enable instantiation of the PP to a Security Target (ST).

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 [8].

6.1 Security Functional 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) Software update of the TOE,

3) Authentication failure handling (FIA_AFL.1): the reaching of the threshold for the unsuccessful authentication attempts and the actions,

4) Timing of authentication (FIA_UAU.1): all unsuccessful use of the authentication mechanism,

5) **Management of security attributes (FMT_MSA.1) /(all instantiations): all modifications of the values of security attributes,**

6) **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,**

7) **Management of TSF data (FMT_MTD.1/ACCESS_CONTROL): All modifications to the values of TSF data,**

8) **Failure with preservation of secure state (FPT_FLS.1): Failure detection of the TSF and secure state,**

9) **Inter-TSF detection of modification (FPT_ITI.1): The detection of modification of imported backuped TSF data,**

10) **Notification of physical attack (FPT_PHP.2): Detection of intrusion (security audit file).**

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 chapter 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.

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 512 to 4096 bits (step 128) key pairs in accordance with FIPS PUB 186-3.**
2. **DES 64 bits in accordance with FIPS PUB 46-3.**
3. **DES2 128 bits in accordance with FIPS PUB 46-3.**
4. **TDES 168 bits in accordance with FIPS PUB 46-3.**
5. **AES 128, 256 bits in accordance with FIPS PUB 197.**
6. **ECC 192 to 521 bits in accordance with FIPS PUB 186-3 and ANSI X9.62.**
7. **Generic Secret 32 to 512 bits in accordance with PKCS#11 v2.11.**

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 **zeroization** 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 (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 512 to 4096 bits (PKCS #1 v1.5),**
2. **RSA PSS, RSA key pairs 512 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 186-3 and ANSI X9.62),**
6. **ECDSA with SHA-1, ECC key pairs 192 to 521bits (FIPS PUB 186-3 and ANSI X9.62),**
7. **EC-KCDSA-SHA256, EC-KCDSA key pairs 256 bits (FIPS PUB 186-3 and ANSI X9.62). Only for internal use.**

Note:

Les paramètres de la courbe sont passés via l'API PKCS#11.

6.1.2.7 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 168 bits (FIPS PUB 113),**
3. **AES MAC, AES MAC-GENERAL, AES keys 128, 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 192 bits (FIPS PUB 198).**

6.1.2.7.1 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 **512 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 **168 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 and CBC mode)** and cryptographic key sizes **128 bits and 256 bits** that meet the following: **FIPS PUB 197**.

FCS_COP.1.1 (AES Mode CTR DECRYPT) The TSF shall perform **symmetric decryption** in accordance with a specified cryptographic algorithm **AES (CTR mode)** and cryptographic key sizes **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**.

6.1.2.8 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).**

6.1.2.9 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 **512 to 4096 bits (step 128)** that meet the following: **PKCS#1 v1.5**.

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**.

6.1.2.10 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.**

6.1.2.11 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**.

6.1.2.12 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 **ANSSI cryptographic referential for "renforcé" strength level [2] 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, 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.**

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 or Crypto-user is allowed to export R.USER_PUB_KEYS.**

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).**

The TOE is used in compliance with its two usage configurations identified in §3.5.7 (Key generation, Key usage). The roles detailed above are those of one of these two configurations: 1, 2, 3, 4 are relevant to configuration "Key generation", while 2,4 and 5 are relevant to configuration "Key usage".

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.**

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 Auditor are allowed**

   (a) to export 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-user are not allowed to export or to delete Audit data.**

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**

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 restore 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 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 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

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

This component is necessary to specify a unique requirement of certificate issuing and management components that is not addressed by the Common Criteria. The specific requirements address the protection of R.USER_DATA and TSF data for 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 CSP_SCD and other 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 TSC.

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, 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 Security Officer, 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 (Security Officer, 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 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 and delete** the security attributes **Role Auditor and HSM Auditor** to **Security Officer and Crypto- officer**, respectively.

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 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 **HSM security officer** to specify alternative initial values to override the default values when an object or information is created.

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 **Security Officer and 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 attribute HSM security officer and User** to **Security Officer and HSM security officer**, respectively.

6.1.5.8 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 Auditor and HSM Auditor** to **Security Officer and HSM security officer**, respectively.

6.1.5.9 FMT_MTD.1 (RAD) Management of TSF data

FMT_MTD.1.1 (RAD) The TSF shall restrict the ability to **modify** the RAD to **User (Security Officer, Auditor, HSM security officer, User and HSM Auditor) for its own RAD**.

6.1.5.10 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**.

6.1.5.11 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).**

6.1.5.12 FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles **Security Officer, HSM security officer, User, 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 **CSP_SCD and other cryptographic keys** by **User**.

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 maximum capacity of the side channels should be defined by the ST, allowing the organisation using the TOE services to prevent any remaining side channels by appropriate security measures in the TOE environment.

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.**

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 a remote 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 a remote 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 a remote trusted IT product and perform **error indication** if modifications are detected.

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. Crypto-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**
- **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 R.SCP-CSD and other secret and private keys in case of physical maintenance or physical tampering. If the detection of opening the device or removal of a cover might not be effective for the switched off device 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.SCP-SCD 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 conditions installation and maintenance, and during operational phase** to demonstrate the correct operation of **the TSF**.

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

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of **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_CRYPTO, 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 5: TOE and environment general overview).

6.2 TOE Security Assurance Requirements

Le niveau des exigences d'assurance de sécurité est EAL4 augmenté de :

- 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).

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

Table 6-1. Assurance Requirements: EAL 4 augmented

Chapitre 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 (HSM security 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 cannot be erased.

7.1.2 Authentication (SF.AUTHENTICATION)

7.1.2.1 SF.AUTHENTICATION.ROLES

The TOE supports the following user categories:

- The Security Officer (SO) is authorized 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 vitual HSM;
  - Update the HSM embedded software;
  - Update the system software;
  - Introduce the software license keys (virtual HSM, …);
  - Modify the network configuration.
- The Auditor is authorized to execute the following functions:
  - Create its own smart card for further authentication;
  - Read general audit file and security audit file generated by the TOE and exported for audit review in the TOE environment;
  - Read the PKCS#11 log file;
  - Get the token status
- The HSM security officer is authorized to execute the following operations:
  - Personalize the virtual HSM;
  - Depersonalize the virtual HSM ;
  - Create the 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.
- The HSM Auditor is authorized to execute the following operations:
  - Read audit data generated by the virtual HSM and exported for audit review in the TOE environment.
- The User is authorized to perform authorized cryptographic operations. The User role is associated with only one user: the client application.

The TOE allows for the creation of multiple users in the HSM security officer and 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 HSM security officer role. A partition may also have one and only one user in the 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, HSM security 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, 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.

User authentication is performed with a password.

The password is configured by the HSM security officer.

If the 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 unauthorized access (user administrative data, keys …).

The TOE is configured via an administration function with authentication by the HSM security officer.

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 HSM security 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 HSM security officer and virtual HSM auditor roles).
- Personalize the virtual HSMs and edit their cryptographic configuration. Personalize a virtual HSM means:
  - Create the role 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 512 to 4096 bits (step 128) key pairs in accordance with FIPS PUB 186-3.
- DES 64 bits in accordance with FIPS PUB 46-3.

- DES2 128 bits in accordance with FIPS PUB 46-3.
- TDES 168 bits in accordance with FIPS PUB 46-3.
- AES 128, 256 bits in accordance with FIPS PUB 197.
- ECC 192 to 521 bits in accordance with FIPS PUB 186-2 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 "renforcé" 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 zeroization.

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 (zeroization) 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 following cryptographic algorithms:

- Symmetric encryption/decryption: AES, DES, 3DES, modes ECB and CBC ;
- Asymmetric encryption/decryption : RSA (RSA-PKCS, RSA-PKCS-PSS, RSA-PKCS-OAEP) ;
- Sign/Verify : 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;
- Hash: SHA256, SHA384, SHA512, SHA-1, MD5.

The TOE implements the following cryptographic operations:

- sign and verify functions
- encrypt and decrypt functions
- hash function
- wrap and unwrap functions
- key management function

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 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 startup.

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 zeroization of BULL keys.

Level 2 alarms lead to a blocking state and provoke the zeroization 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 Startup tests

At startup, 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 integriry of sensitive functions:

- Cryptographic operations;
- The random number generator.

7.1.7.3.3 Code integrity tests

The FPGA software integriry is verified:

- FPGA bitstream ;
- Software executed by the NIOS.

The integrity checking of the software running on the ComExpress module (Linux) is provided by the BIOS PCA4 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 Tripwire, an open source software available under GPL license for Linux. 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 log.

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.

Chapitre 8.PP claims

8.1 PP référence

The ST is compliant with the Protection Profile prEN 14167-2:2012 (Cryptographic Module for CSP Signing Operations with Backup – Protection Profile – CMCSOB PP).

The ST also includes most of the security requirements of Protection Profile prTS 14167-3:2011 (Cryptographic Module for CSP key generation services – CMCKG PP).

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

The following threats have been added to the PP:

- 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

The following assumption has been added to the PP:

- A.ADMIN (A)

- A.PROTECTION_HOST (A)

8.2.3 Security objectives

The following security objectives have been added to the PP:

- 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

The following security objective for the environment has been added to the PP:

- O.ENV_ADMIN (A)
- O.ENV_PROTECTION_HOST (A)
- O.ENV_SECURE_CHANNEL (E)

8.2.5 Security Functional Requirements

The following IT security requirements have been added to the PP:

- FCS_COP.1/MESSAGE AUTHENTICATION (I)
- FCS_COP.1/MESSAGE AUTHENTICATION VERIFY (I)
- FCS_COP.1/ENCRYPT (I)
- FCS_COP.1/DECRYPT (I)
- FCS_COP.1/DIGEST (I)
- FCS_COP.1/WRAP (I)
- FCS_COP.1/UNWRAP (I)
- FDP_ACC.1/CONFIG (I)
- FDP_ACF.1/CONFIG (I)
- FDP_ACC.1/LOAD (I)
- FDP_ACF.1/LOAD (I)
- FMT_MOF.1 (A)
- FPT_ITI.1 (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.

prEN 14167-2 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 (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).

8.3 Configuration recommandations

To comply with the Protection Profile, it is necessary to strongly partition both operation phases: key generation and use of generated keys for cryptographic operations such as signing, verifying, ...).

One recommendation is to define two virtual HSMs which must not be used simultaneously, namely:

- A KGC (Key Generation Center) virtual HSM on which are performed key generation and secure backup (« Key generation » configuration)
- A « User » virtual HSM on which are performed key restoration and cryptographic operations using the restored keys (« Key usage » configuration)

Both virtual HSMs must be installed with the same set of installation smart cards, so that key restoration can be achieved.

8.3.1 Key generation configuration (KGC virtual HSM)

All the keys will be generated on the KGC virtual HSM.

8.3.1.1 Configuration

Recommendations concerning the KGC virtual HSM configuration using the administration application:

Cryptographic configuration

Options

Memory size

Configure only the types of keys to be generated

Keys

Configure only the types of keys to be generated

Countermeasures

Activate the protection against « timing attack »
Activate the protection against « fault attack »

Operations

Encrypt

Uncheck all the options to forbid the operation

Sign

Uncheck all the options to forbid the operation

Wrap

Uncheck all the options to forbid the operation

Generate

Check only the types of keys to be generated
Uncheck CKA_WRAP+CKA_DECRYPT

Create

Uncheck all the options to forbid the operation

8.3.1.2 Key generation

Recommendations concerning the attributes of generated keys:

Public keys (only one cryptographic operation):

CKA_MODIFIABLE=FALSE
CKA_VERIFY

Private keys:

CKA_PRIVATE=TRUE, CKA_MODIFIABLE=FALSE,
CKA_SIGN

Symmetric keys (only one type of cryptographic operation):

CKA_PRIVATE=TRUE, CKA_MODIFIABLE=FALSE,
CKA_SIGN/VERFY

8.3.2 Key usage configuration (User virtual HSM)

8.3.2.1 Configuration

Recommendations concerning the User virtual HSM configuration using the administration application:

Cryptographic configuration

Options

Memory size

Configure only the types of keys to be used.

Keys

Configure only the types of keys to be used

Countermeasures

Activate the protection against « timing attack »
Activate the protection against « fault attack »

Operations

Encrypt

Uncheck all the options to forbid the operation

Sign

Check only the signing mechanism to be used

Wrap

Uncheck all the options to forbid the operation

Generate

Uncheck all the options to forbid the operation

Create

Uncheck all the options to forbid the operation

Chapitre 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
Policies	
P.ALGORITHMS	O.KEYS_SECURE, O.CRYPTO_SECURE, O.PROTECT_EXPORTED_DATA, O.USER_AUTHENTICATION
Threats	
T.BACKUP_RESTORE	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
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

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
Policy/Threat/Assumptions	**Objectives**
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

Table 9-1. Security Environment to Security Objectives Mapping

Objectives	**Policy/Threat/Assumptions**
Security Objectives for the TOE	
O.AUDIT	T.BACKUP_RESTORE, T.BAD_SW, T.MALFUNCTION, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.PHYS_MANIPUL
O.KEYS_SECURE	T.KEYS_DERIVE, T.KEYS_DISCLOSE, T.KEYS_ALTERATION, T.MISUSE_OPERATION, P.ALGORITHMS
O.CHECK_OPERATION	T.BACKUP_RESTORE, T.BAD_SW, T.KEYS_ALTERATION, T.MALFUNCTION, T.PHYS_MANIPUL
O.RBAC	T.BACKUP_RESTORE, T.BAD_SW, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.MISUSE_OPERATION
O.ATTACK_RESPONSE	T.KEYS_ALTERATION, T.PHYS_MANIPUL
O.SECURE_STATE	T.BACKUP_RESTORE, T.BAD_SW, T.KEYS_ALTERATION, T.MALFUNCTION, T.INSECURE_INIT, T.PHYS_MANIPUL
O.PROTECT_EXPORTED_DATA	T.BACKUP_RESTORE, 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_RESTORE, 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
O.DEPERSONALIZATION	T.KEYS_ALTERATION, T.INSECURE_INIT, T.MISUSE_OF_TOE

Objectives	Policy/Threat/Assumptions
Security Objectives for the Environment	
O.ENV_APPLICATION	T.BACKUP_RESTORE, T.BAD_SW, T.CSP_SVD_ALTERATION, T.DATA_MANIPUL, T.INSECURE_INIT, A.AUDIT_SUPPORT, A.CORRECT_DTBS, A.CRYPTOUSER_AGENT
O.ENV_AUDIT	T.BACKUP_RESTORE, T.BAD_SW, T.MALFUNCTION, T.INSECURE_INIT, T.MISUSE_OF_TOE, T.PHYS_MANIPUL , A.AUDIT_SUPPORT
O.ENV_PERSONNEL	T.BACKUP_RESTORE, T.BAD_SW, T.MALFUNCTION, T.INSECURE_INIT, T.MISUSE_OF_TOE, A.AUDIT_SUPPORT, A.DATA_STORE
O.ENV_PROTECT_ACCESS	T.KEYS_ALTERATION, T.MALFUNCTION, T.PHYS_MANIPUL, A.TRUSTED_ENVIRONMENT
O.ENV_RECOVERY	T.BACKUP_RESTORE, T.MALFUNCTION, A.DATA_STORE
O.ENV_SECURE_INIT	T.KEYS_DISCLOSE, T.KEYS_ALTERATION, T.INSECURE_INIT
O.ENV_SECURE_OPER	T.MISUSE_OF_TOE, T.MISUSE_OPERATION, A.TRUSTED_ENVIRONMENT, A.CORRECT_DTBS, A.DATA_STORE,
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

Table 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 Threats

T.BACKUP_RESTORE

Backup and Restore operations shall be auditable events, recorded in a general events log file (O.AUDIT) and protected in integrity (O.CHECK_OPERATION). HSM security officer (O.RBAC) shall only perform them after identification of the user (O.USER_AUTHENTICATION) thanks to a reliable client application (O.ENV_APPLICATION). Auditor has access to previous restore and backup operations by analysis of stored audit logs (O.ENV_AUDIT)

The data export mechanism shall use adequate confidentiality and integrity cryptographic mechanism to prevent 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.ENV_RECOVERY) and personnel shall be trained to perform these tasks (O.ENV_PERSONNELl). 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 client 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 covert 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 check 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) 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).

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 (O.ENV_APPLICATION). 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

Malfunction shall be detected by monitoring operation of the TOE (O.CHECK_OPERATION). In case a malfunction arise, 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 voluntary physical destruction of the TOE, the TOE shall be used in a secure place (O.ENV_PROTECT_ACCESS).

T.INSECURE_INIT

Insecure initialisation can lower the security level of the TOE. Therefore, activities upon the TOE shall be secured by a user authentication (O.USER_AUTHENTICATION) performed by the client application (O.ENV_APPLICATION) to check if the authenticated user has the necessary rights to perform the initialisation operation (O.RBAC). Critical operation shall be logged (O.AUDIT + O.ENV_AUDIT) and 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). If a problem appears during the initialisation process, the TOE shall jump or remain in a secure state (O.SECURE_STATE).

Of course, 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).

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.

T.MISUSE_OF_TOE

The TOE is used to generate R.USER_DATA /R.USER_PUB_KEYS pair. This is a critical operation that has to be performed by authorised personnel (O.USER_AUTHENTICATION) with a Crypto-Officer role (O.RBAC + O.ENV_APPLICATION). This operation shall be log for accountability (O.AUDIT + O.ENV_AUDIT).

Only trusted personnel should access the Crypto-Officer role (O.ENV_PERSONNEL) and they will have to follow procedures and controls in the TOE environment that allow operating the TOE within a CA system in compliance with the requirements of the EU directive (O.ENV_SECURE_OPER).

T.MISUSE_OPERATION

This threat describes the fact that an attacker succeeds in using the signature mechanism of the TOE with a forged R.USER_DATA. This should not happen because the Crypto-Officer is the only role that is allowed to generate the R.USER_DATA (O.RBAC). Therefore, the user that acts as Crypto-Officer shall be authenticated (O.USER_AUTHENTICATION) and be aware of its responsibilities (O.ENV_PERSONNEL).

Once generated, the R.USER_DATA is kept confidential within the TOE boundary (O.KEYS_SECURE) and without knowledge of the actual R.USER_DATA it shall be impossible to produce an advanced electronic signature (O.CRYPTO_SECURE). Finally, the TOE environment shall not facilitate misuse of the TOE (O.ENV_SECURE_OPER).

T.PHYS_MANIPUL

Physical manipulation (box opening, penetration of secure area of the TOE) of the TOE could lead to a loss of confidentiality or integrity of the R.USER_DATA. Therefore, the TOE shall prevent this tampering by detecting physical manipulation by entering a secure state (O.SECURE_STATE) or even destroy R.USER_DATA (O.ATTACK_RESPONSE). Physical manipulation can be also detect by a loss of integrity of critical data, therefore they need to be regularly checked (O.CHECK_OPERATION). To prevent physical manipulation, the TOE shall be placed in a secure place (O.ENV_PROTECT_ACCESS) and every physical manipulation shall be logged (O.AUDIT + O.ENV_AUDIT).

T.INSECURE_CHANNEL

An attacker could manipulate sensitive data from applications sent by an authorized user to the TOE could be manipulated during the network transmission, and thus affect the TOE initialisation or configuration. O.ENV_SECURE_CHANNEL ensure the confidentiality and integrity of the data transferred between the applications (client applications and administration application) and the TOE.

T. TRUSTED_PATH

An attacker could manipulate sensitive authentication data sent by an unauthorized personnel to the TOE, and thus affect the TOE initialisation or configuration. O.TRUSTED_PATH counters this threat.

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.

9.2.2.2 Organisational Security Policies

P.ALGORITHMS

Cryptographical algorithms used in O.CRYPTO_SECURE have to comply with the referentials for “renforcé” strength level edited by ANSSI ([2], [3], [4]).

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 referentials.

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 integrity to the data to be signed. 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 assumes that TOE's environment provides a compliant operational framework as expected by European regulation (O.ENV_SECURE_OPER). This induces also that personnel perform their tasks efficiently (O.ENV_PERSONNEL) and that in case of trouble backup will allow a quick restart of the system (O.ENV_RECOVERY) and that secure initialization procedure will be followed (O.ENV_SECURE_INIT) during the recovery.

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 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 and appropriately trained.

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 authorized remote applications.

9.3 Security Requirements Rationale

9.3.1 Security Requirement Coverage

Objectives	Requirements
Security Objectives for the TOE	
O.AUDIT	FAU_GEN.1, 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, 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_CRYPTO, FMT_MSA.3, FPR_UNO.1FPT_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, 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_CRYPTO, FMT_MSA.1/ROLE_AUDIT, FMT_MOF.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1/ACCESS_CONTROL, FMT_MTD.1/USER_CRYPTO, FMT_MTD.1/USER_AUDIT, FMT_MTD.1/RAD, FMT_MTD.1/AUDIT, FPT_TST.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_CRYPTO, 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, FCS_COP.1/VERIFY, FCS_COP.1/DECRYPT, FDP_ACC.1/LOAD, FDP_ACF.1/LOAD
O. DEPERSONALIZATION	FDP_ACC.1/DEPERSONALIZATION, FDP_ACC.1/DEPERSONALIZATION, FDP_RIP.1

Table 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.

Additional audit is implemented by the SFR FAU_GEN.1. 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 HSM security 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:

- Import the backup keys by FCS_CKM.2,
- 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. 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, the SFR FMT_MOF.1 ensures that the backup function be disable to fit some particular national laws and the SFR AVA_VAN.5 requires subject side-channels to the vulnerability analysis.

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/SIGN according to the list of approved algorithms and parameters. The confidentiality and integrity of the R.USER_DATA will be protected by SFR FDP_RIP.1 and FDP_SDI.2 while internal processing. The SFR FCS_CKM.2/Other_keys assures th distribution of cryptographic keys. The SFR FCS_CKM.4 requires secure key destruction to prevent any misuse of R.USER_DATA after operational lifetime. The 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. The complex protection of the R.USER_DATA as most valuable asset requires a systematic and complete vulnerability analysis considering high attack potential by SAR AVA_VAN.5.

O.CHECK_OPERATION

This objective addresses 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 or HSM security officer) about the performed self-tests and their results. 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 3 SFP:

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 security officer and 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 HSM security officer and audit functions to the Auditor.

The SFR FMT_MSA.1/ROLE_CRYPTO extends the 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 HSM security officer. Note that 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 CSP security measures for physical protection of the TOE.

O.SECURE_STATE

This objective addresses a secure state and protection of R.USER_DATA confidentiality whenever the TOE detects an 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 TSF shall destroy the plaintext R.USER_DATA and other confidential secret and private keys if failures occur. The SFR FPT_RCV.1 requires a maintenance mode where the ability to return the TOE to a secure state is provided.

Note that the 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 with respect to the organisational Security Policy P.ALGORITHMS. The SFR FPR_UNO.1 requires TSF to prevent illicit information flow about the R.USER_DATA through side-channels in the signatures. The SAR AVA_VAN.5 requires covert-channel analysis and a systematic and complete vulnerability analysis considering high attack potential. That is because the signature-creation with R.USER_DATA especially for certificates is the most important and critical service of the TOE.

O.USER_AUTHENTICATION

This objective addresses the identification and authentication the users before having any access to TOE protected assets. The SFR require timing identification by FIA_UID.1 and timing authentication by FIA_UAU.1. 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). Therefore these actions support the TOE protection and do not allow any access to the TOE protected assets. The SFR FIA_ATD.1 defines the security attributes for identity based authentication. Note that the client application might be the only user and may act as agent for several end-users in the TOE environment (see O.ENV_APPLICATION). The SFR FIA_SOS.1 ensures the verification of the quality of the secret used for authentication. The SFR FIA_AFL.1 protects the VAD against guessing. The SFR FMT_MTD.1/USER_CRYPTO, FMT_MTD.1/USER_AUDIT, FMT_MTD.1/RAD and FMT_SMF.1 provide management functions for identification.

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 SFR.

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. The software update process is performed through a logical secured channel initiated by an administrative command under crypto-officer control: The software is decrypted according to FCS_COP.1/DECRYPT and verified according to FCS_COP.1/VERIF RSA. The load TSF will be under access control required by the SFR FDP_ACF.1/LOAD according to FDP_ACC.1/LOAD. FMT_MOF.1 ensures that the loading function be disabled to forbid any new update. FAU_GEN.1 generates audit records about the software update results.

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.

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/bckup	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/SIGN	SF.CO.CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/VERIFY	SF.CO. CRYPTOGRAPHIC_FUNCTIONS, SF.SL
FCS_COP.1/ENCRYPT	SF.CO. CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/DECRYPT	SF.CO. CRYPTOGRAPHIC_FUNCTIONS, SF.SL, SF.CHIFFREMENT_LOGICIEL
FCS_COP.1/MESSAGE AUTHENTICATION	SF.CO. CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/MESSAGE AUTHENTICATION VERIFY	SF.CO. CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/DIGEST	SF.CO. CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/WRAP	SF.CO. CRYPTOGRAPHIC_FUNCTIONS
FCS_COP.1/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
FDP_ACF.1/CRYPTO	SF.AUTHENTICATION.ROLES
FDP_ACF.1/AUDIT	SF.AUTHENTICATION.ROLES
FDP_ACF.1/BACKUP	SF.AUTHENTICATION.ROLES

Security requirements	TOE security functions
FDP_ACF.1/LOAD	SF.AUTHENTICATION.ROLES
FDP_ACF.1/CONFIG	SF.AUTHENTICATION.ROLES
FDP_ACF.1/DEPERSONALIZATION	SF.AUTHENTICATION.ROLES
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
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_CRYPTO	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_CRYPTO	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
FTP_TRP.1/TOE	SF.AUTHENTICATION.TRUSTED_PATH

Table 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 (User Identity association) 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 and 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/SIGN and FCS_COP.1/VERIF (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 ECKCDSA-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 and FCS_COP.1/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 SHA and SHA-2 algorithms.

FCS_COP.1/WRAP and FCS_COP.1/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 and 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 desinstallation 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 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 Authorized 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 Authorized Administrators from one another. SF.AUTHENTICATION.ROLES defines roles and crypto officer identities.

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 Authorized Administrator identity is identified to the TOE before anything occurs on behalf of the Authorized 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 Authorized Administrator. SF.ACCESS_CONTROL implements such a control with several control options and SF.AUTHENTICATION_ROLES defines administrator (crypto officer) role.

FMT_MSA.1/ROLE_CRYPTO 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 and 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 and 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, security officer, auditor, crypto officer and HSM auditor.

FMT_MTD.1/ACCESS_CONTROL, FMT_MTD.1/USER_CRYPTO, 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 Authorized 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 Authorized 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 and clears the whole secure memory.

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 guaranties 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 date and time setting at installation phase.

FPT_TST.1 (TSF Testing) ensures the integrity of the operation of the TSF and to provide the Authorized Administrator a means to verify the integrity of the 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-wide consistency tests for public and private keys.

FPT_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 guaranties that any authentication takes place via a smart card reader housed into the appliance, which is linked by a serial connection (trusted path) to the TOE.

9.5 Dependency Rationale

9.5.1 Functional and Assurance Requirements Dependencies

Requirement	CC-required Dependencies	Remark
Functional Requirements for the TOE		
FAU_GEN.1	FPT_STM.1	dependency is not satisfied by the CWA 14167-2 PP (see justification in section 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 nor satisfied (the keys are entered through the PKCS#11API)
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 dependencies)	
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 dependencies)	
FMT_SMR.1	FIA_UID.1	
FPR_UNO.1	(no dependencies)	
FPT_FLS.1	(no dependencies)	
FPT_ITI.1	(no dependencies)	

Requirement	CC-required Dependencies	Remark
FPT_ITT.1	(no dependencies)	
FPT_PHP.2	FMT_MOF.1	dependency is not satisfied by the CWA 14167-2 PP (see justification in section 9.5.2)
FPT_PHP.3	(no dependencies)	
FPT_RCV.1	AGD_OPE.1	
FPT_STM.1	(no dependencies)	
FPT_TST.1	(no dependencies)	
FTP_TRP.1	(no dependencies)	

Assurance Requirements		
ADV_ARC.1	ADV_FSP.1, ADV_TDS.1	ADV_FSP.2 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 dependencies)	
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 dependencies)	
ALC_DEL.1	(no dependencies)	
ALC_DVS.2	(no dependencies)	
ALC_FLR.3	(no dependencies)	
ALC_LCD.1	(no dependencies)	
ALC_TAT.1	ADV_IMP.1	ADV_IMP.2 is included and 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 included and 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.2 is hierarchical to ADV_FSP.1 ATE_COV.2 is included and 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 included and hierarchical to ADV_IMP.1

Table 9-5. Functional and Assurance Requirements Dependencies

9.5.2 Justification of Unsupported Dependencies

Component	Justification for not including	Remark
Security Functional Requirements for the TOE		
FAU_GEN.1	FPT_STM.1	FAU_GEN.1 uses sequence data, which may be a sequence number or reliable time stamp. If sequence number is used FPT_STM.1 is not needed. The application note directs the ST editor to include FPT_STM.1 if reliable time stamp is used by the TOE.
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. No management of security functions behaviour is needed.

Table 9-6. 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**.

9.6.1 AVA_VAN.5 Advanced methodical vulnerability analysis

The TOE generates uses and manages the most sensitive data of the CSP – 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.

The protection of the R.USER_DATA shall be solely and in tabloid form provided by the CM as part of the trustworthy system. 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.

Chapitre 10. Appendix A – Acronyms

CC	Common Criteria
EAL	Evaluation Assurance Level
IT	Information Technology
PP	Protection Profile
SF	Security Function
SAR	Security assurance requirements
SFP	Security Function Policy
SFR	Security functional requirements
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy