TrustWay

BULL TRUSTWAY PCI CRYPTOGRAPHIC CARD

Common Criteria / ISO15408

Security Target

Version 3.9

TrustWay

—— this page has intentionally been left blank ——

Table of Contents

Table of Contents 4
Terminology 10
Terminology 10
1 ST introduction 12
1.1 ST identification 12
1.2 ST overview 12
1.3 CC conformance 13
2 TOE description 14
2.1 Product type 14
2.2 TOE description 14
2.2.1 Architecture 14
2.2.2 TrustWay Card Product life-cycle 16
2.3 TOE functionalities 17
2.3.1 Cryptographic operations 17
2.3.2 Key management 17
2.3.3 TOE roles 17
2.3.4 Administration 18
2.3.5 Security mechanisms 18
2.3.5.1 Physical security 18
2.3.5.2 Periodical tests 19
2.4 TOE Usage 19
3 TOE Security Environment 21
3.1 Assets to protect 21
3.2 Assumptions 21
3.3 Threats to Security 23
3.4 Organisational Security Policies 25
4 Security Objectives 26
4.1 Security Objectives for the TOE 26
4.2 Security Objectives for the Environment 28
5 IT Security Requirements 30
5.1 TOE Security Functional Requirements 30
5.1.1 Security audit (FAU) 30
5.1.1.1 Audit data generation (FAU_GEN.1) 30
5.1.1.2 User identity association (FAU_GEN.2) 31
5.1.1.3 Guarantees of audit data availability (FAU_STG.2/TOE) 31
5.1.2 Cryptographic support (FCS) 32
5.1.2.1 Cryptographic key generation (FCS_CKM.1) 32
5.1.2.2 Cryptographic key distribution (FCS_CKM.2/backup_keys) 33
5.1.2.3 Cryptographic key distribution (FCS_CKM.2/other_keys) 33
5.1.2.4 Cryptographic key destruction (FCS_CKM.4) 33
5.1.2.5 Cryptographic operation (FCS_COP.1/SIGN) 33
5.1.2.6 Cryptographic operation (FCS_COP.1/VERIF) 34
5.1.2.7 Cryptographic operation (FCS_COP.1/ENCRYPT) 35
5.1.2.8 Cryptographic operation (FCS_COP.1/DECRYPT) 36
5.1.2.9 Cryptographic operation (FCS_COP.1/DIGEST) 36
5.1.2.10 Cryptographic operation (FCS_COP.1/WRAP) 36
5.1.2.11 Cryptographic operation (FCS_COP.1/UNWRAP) 36

5.1.2.12 Cryptographic operation (FCS_COP.1/BACKUP_ENC) 37
5.1.2.13 Cryptographic operation (FCS_COP.1/BACKUP_INT) 37
5.1.2.14 Quality metrics for random numbers (FCS_RND.1) 37
5.1.3 User data protection (FDP) 37
5.1.3.1 Subset access control (FDP_ACC.1/CRYPTO) 37
5.1.3.2 Subset access control (FDP_ACC.1/AUDIT) 37
5.1.3.3 Subset access control (FDP_ACC.1/BACKUP) 38
5.1.3.4 Subset access control (FDP_ACC.1/LOAD) 38
5.1.3.5 Subset access control (FDP_ACC.1/KEYS_DISTRIBUTION) 38
5.1.3.6 Security attribute based access control (FDP_ACF.1/CRYPTO) 38
5.1.3.7 Security attribute based access control (FDP_ACF.1/AUDIT) 39
5.1.3.8 Security attribute based access control (FDP_ACF.1/BACKUP) 39
5.1.3.9 Security attribute based access control (FDP_ACF.1/KEYS_DISTRIBUTION) 40
5.1.3.10 Security attribute based access control (FDP_ACF.1/LOAD) 40
5.1.3.11 Backup and recovery (FDP_BKP.1) 41
5.1.3.12 Export of user data without security attributes (FDP_ETC.1) 41
5.1.3.13 Subset information flow control (FDP_IFC.1/BACKUP) 41
5.1.3.14 Subset information flow control (FDP_IFC.1/CRYPTO) 42
5.1.3.15 Partial elimination of illicit information flows (FDP_IFF.4/BACKUP) 42
5.1.3.16 Partial elimination of illicit information flows (FDP_IFF.4/CRYPTO) 42
5.1.3.17 Subset residual information protection (FDP_RIP.1) 43
5.1.3.18 Stored data integrity monitoring and action (FDP_SDI.2) 43
5.1.4 Identification and authentication (FIA) 43
5.1.4.1 Authentication failure handling (FIA_AFL.1) 43
5.1.4.2 User attribute definition (FIA_ATD.1) 44
5.1.4.3 Verification of secrets (FIA_SOS.1) 44
5.1.4.4 Timing of authentication (FIA_UAU.1) 44
5.1.4.5 Timing of identification (FIA_UID.1) 44
5.1.5 Security management (FMT) 44
5.1.5.1 Management of security functions behaviour (FMT_MOF.1) 44
5.1.5.2 Management of security attributes (FMT_MSA.1/ROLE_CRYPTO) 44
5.1.5.3 Management of security attributes (FMT_MSA.1/ROLE_AUDIT) 45
5.1.5.4 Secure security attributes (FMT_MSA.2) 45
5.1.5.5 Static attribute initialisation (FMT_MSA.3) 45
5.1.5.6 Management of TSF data (FMT_MTD.1/ACCESS_CONTROL) 45
5.1.5.7 Management of TSF data (FMT_MTD.1/USER_CRYPTO) 45
5.1.5.8 Management of TSF data (FMT_MTD.1/USER_AUDIT) 45
5.1.5.9 Management of TSF data (FMT_MTD.1/RAD) 45
5.1.5.10 Management of TSF data (FMT_MTD.1/AUDIT) 45
5.1.5.11 Specification of Management Functions (FMT_SMF.1) 46
5.1.5.12 Security roles (FMT_SMR.1) 46
5.1.6 Protection of the TOE Security Functions (FPT) 46
5.1.6.1 Abstract machine testing (FPT_AMT.1) 46
5.1.6.2 Failure with preservation of secure state (FPT_FLS.1) 46
5.1.6.3 Inter-TSF confidentiality during transmission (FPT_ITC.1) 46
5.1.6.4 Inter-TSF detection of modification (FPT_ITI.1) 46
5.1.6.5 Notification of physical attack (FPT_PHP.2) 47
5.1.6.6 Resistance to physical attack (FPT_PHP.3) 47
5.1.6.7 Manual recovery (FPT_RCV.1) 48

5.1.6.8 Time stamps (FPT_STM.1) ... 48
5.1.6.9 TSF testing (FPT_TST.1) ... 48
5.1.7 Trusted path (FTP) ... 48
5.1.7.1 Trusted path (FTP_TRP.1/TOE) ... 48
5.2 TOE Security Assurance Requirements ... 49
5.2.1 Configuration management (ACM) ... 49
5.2.1.1 Partial CM automation (ACM_AUT.1) ... 49
5.2.1.2 Generation support and acceptance procedures (ACM_CAP.4) ... 50
5.2.1.3 Problem tracking CM coverage (ACM_SCP.2) ... 50
5.2.2 Delivery and operation (ADO) ... 51
5.2.2.1 Detection of modification (ADO_DEL.2) ... 51
5.2.2.2 Installation, generation, and start-up procedures (ADO_IGS.1) ... 51
5.2.3 Development (ADV) ... 51
5.2.3.1 Fully defined external interfaces (ADV_FSP.2) ... 51
5.2.3.2 Security enforcing high-level design (ADV_HLD.2) ... 51
5.2.3.3 Implementation of the TSF (ADV_IMP.2) ... 52
5.2.3.4 Descriptive low-level design (ADV_LLD.1) ... 52
5.2.3.5 Informal correspondence demonstration (ADV_RCR.1) ... 53
5.2.3.6 Informal TOE security policy model (ADV_SPM.1) ... 53
5.2.4 Guidance documents (AGD) ... 53
5.2.4.1 Administrator guidance (AGD_ADM.1) ... 53
5.2.4.2 User guidance (AGD_USR.1) ... 54
5.2.5 Life cycle support (ALC) ... 55
5.2.5.1 Identification of security measures (ALC_DVS.1) ... 55
5.2.5.2 Systematic flaw remediation (ALC_FLR.3) ... 55
5.2.5.3 Developer defined life-cycle model (ALC_LCD.1) ... 56
5.2.5.4 Well-defined development tools (ALC_TAT.1) ... 56
5.2.6 Tests (ATE) ... 56
5.2.6.1 Analysis of coverage (ATE_COV.2) ... 56
5.2.6.2 Testing: high-level design (ATE_DPT.1) ... 56
5.2.6.3 Functional testing (ATE_FUN.1) ... 57
5.2.6.4 Independent testing - sample (ATE_IND.2) ... 57
5.2.7 Vulnerability assessment (AVA) ... 57
5.2.7.1 Covert channel analysis (AVA_CCA.1) ... 57
5.2.7.2 Validation of analysis (AVA_MSU.2) ... 58
5.2.7.3 Strength of TOE security function evaluation (AVA_SOF.1) ... 58
5.2.7.4 Highly resistant (AVA_VLA.4) ... 58
5.3 Security Requirements for the IT Environment ... 59
5.3.1 Security audit (FAU) ... 59
5.3.1.1 Audit review (FAU_SAR.1) ... 59
5.3.1.2 Protected audit trail storage (FAU_STG.1/ENVIRONMENT) ... 59
5.3.2 User data protection (FDP) ... 59
5.3.2.1 Subset access control (FDP_ACC.1/CLIENT) ... 59
5.3.2.2 Security attribute based access control (FDP_ACF.1/CLIENT) ... 60
5.3.2.3 Data exchange integrity (FDP_UIT.1) ... 60
5.3.3 Identification and authentication (FIA) ... 60
5.3.3.1 Timing of authentication (FIA_UAU.1/CLIENT) ... 60
5.3.3.2 Timing of identification (FIA_UID.1/CLIENT) ... 61
5.3.4 Non-IT requirements ... 61
6 TOE summary specification ... 63

6.1 TOE security functions 63
6.1.1 SF.SL (secure loading) 63
6.1.1.1 General mechanism 63
6.1.1.2 Signature mechanism used 63
6.1.2 SF.SI (secure installation) 63
6.1.3 SF.keys_distribution 64
6.1.4 SF.CO (cryptographic operation) 64
6.1.4.1 SF.CO.key_generation 64
6.1.4.2 SF.CO.key_destruction 65
6.1.4.3 SF.CO.cryptographic_functions 65
6.1.5 SF.backup 65
6.1.5.1 SF.backup.command 65
6.1.5.2 SF.backup.audit 66
6.1.5.3 SF.backup.data_protection 66
6.1.6 SF.authentification 66
6.1.6.1 SF.Authentification.Roles 66
6.1.6.2 SF. Authentification.Trusted_Path 66
6.1.6.3 SF.Authentification.policy 67
6.1.7 SF.Access_Control 68
6.1.8 SF.audit 68
6.1.8.1 SF.audit.events 68
6.1.8.2 SF.audit.file 69
6.1.9 SF.SM (security mechanisms) 69
6.1.9.1 SF.SM.hardware 69
6.1.9.2 SF.SM.tests 70
6.1.9.3 SF.SM.Alarms 71
6.2 Assurance measures 72
7 PP claims 74
7.1 PP reference 74
7.2 PP addition 74
8 Rationale 76
8.1 Introduction 76
8.2 Security Objectives Rationale 76
8.2.1 Security Objectives Coverage 76
8.2.2 Security Objectives Sufficiency 77
8.2.2.1 Policies and Security Objective Sufficiency 78
8.2.2.2 Threats and Security Objective Sufficiency 78
8.2.2.3 Assumptions and Security Objective Sufficiency 81
8.3 Security Requirements Rationale 83
8.3.1 Security Requirement Coverage 83
8.3.2 Security Requirements Sufficiency 84
8.3.2.1 TOE Security Requirements Sufficiency 84
8.3.2.2 TOE Environment Security Requirements Sufficiency 87
8.4 TOE Summary Specification Rationale 88
8.4.1 TOE Security functions Coverage 88
8.4.2 TOE Security functions Sufficiency 90
8.5 Dependency Rationale 94
8.5.1 Functional and Assurance Requirements Dependencies 94
8.5.2 Justification of Unsupported Dependencies 97
8.6 Security Requirements Grounding in Objectives 98

8.7 Rationale for Extensions 100
8.7.1 Rationale for Extension of Class FCS with Family FCS_RND 100
8.7.2 Rationale for Extension of Class FDP with Family FDP_BKP 101
8.8 Rationale for Assurance Level 4 Augmented 102
9 References 104
10 Appendix A – Acronyms 105

—— this page has intentionally been left blank ——

Terminology

Terminology

Administrator means a CSP user role that performs TOE initialisation or other TOE administrative functions. These tasks are mapped to the Crypto-officer role of the TOE.

Advanced electronic signature (defined in the Directive [1], article 2.2) means an electronic signature which meets the following requirements:

(a) it is uniquely linked to the signatory;
(b) it is capable of identifying the signatory;
(c) it is created using means that the signatory can maintain under his sole control, and
(d) it is linked to the data to which it relates in such a manner that any subsequent change of the data is detectable.

Authentication data is information used to verify the claimed identity of a user.

Auditor means a user exporting the TOE audit data and reviewing the audit data with tools in the TOE environment.
Backup means export 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. Note that backup is the only function which is allowed to export CSP-SCD and only if backup package is implemented.
CEN workshop agreement (CWA) is a consensus-based specification, drawn up in an open workshop environment of the European Committee for Standardization. CWA 14167-2 Protection Profile represents Annex A to the CWA that has been developed by the European Electronic Signature Standardisation Initiative (EESSI) CEN/ISSS electronic signature (E-SIGN) workshop, Area D2 on trustworthy systems.
Certificate means an electronic attestation which links the SVD to a person and confirms the identity of that person. (defined in the Directive [1], article 2.9)
CSP signature creation data (CSP-SCD) means 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 signature verification data (CSP-SVD) means SVD which corresponds to the CSP-SCD and which is used to verify the advanced electronic signature in the qualified certificate or for signing certificate status information.
Certification-service-provider (CSP) means an entity or a legal or natural person who issues certificates or provides other services related to electronic signatures (defined in the Directive [1], article 2.11).
Cryptographic operation means any cryptographic function performed in accordance with a specified algorithm and, optionally, with a cryptographic key of a specified size. Key generation and destruction are considered apart.
Data to be signed (DTBS) means the complete electronic data to be signed, such as QC content data or certificate status information.
Data to be signed representation (DTBS-representation) means the data sent to the TOE for signing and is
(a) a hash-value of the DTBS or
(b) an intermediate hash-value of a first part of the DTBS and a remaining part of the DTBS or
(c) the DTBS itself.
The client indicates to the TOE the case of DTBS-representation, unless implicitly indicated. The hash-value in case (a) or the intermediate hash-value in case (b) is calculated by the client. The final hash-value in case (b) or the hash-value in case (c) is calculated by the TOE.

Digital signature means 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 e.g. by the recipient. [ISO 7498-2]

Directive The Directive 1999/93/ec of the European parliament and of the council of 13 December 1999 on a Community framework for electronic signatures [1] is also referred to as the 'Directive' in the remainder of the ST.
Dual person control means a special form of access control of a task which requires two users with different identities to be authenticated and authorised to the defined roles at the time this task is to be performed or at the time authentication data to authorised the task are generated.
Hardware security module (HSM) means the cryptographic module used to generate the advanced signature in qualified certificates and which represents the TOE.
List of approved algorithms and parameters means cryptographic algorithms and parameters published in [5] for electronic signatures, secure signature creation devices and trustworthy systems
Reference authentication data (RAD) means data persistently stored by the TOE for verification of the authentication attempt as authorised user.
Restore means import of the backup data to recreate the state of the TOE at the time the backup was created.
Qualified certificate (QC) means a certificate which meets the requirements laid down in Annex I of the Directive [1] and is provided by a CSP who fulfils the requirements laid down in Annex II of the Directive [1]. (defined in the Directive [1], article 2.10)
Secure signature-creation device (SSCD) means configured software or hardware which is used to implement the SCD and which meets the requirements laid down in Annex III of the Directive [1]. (SSCD is defined in the Directive [1], article 2.5 and 2.6).
Side-channel means illicit information flow in result of the physical behaviour of the technical implementation of the TOE. Side-channels are but limited to interfaces not intended for data output like power consumption, timing of any signals and radiation. Side-channels might be enforced by influencing the TOE behaviour from outside.
Signature-creation data (SCD) means unique data, such as codes or private cryptographic keys, which are used by the signatory to create an electronic signature. (defined in the Directive [1], article 2.4)
Signature-verification data (SVD) means data, such as codes or public cryptographic keys, which are used for the purpose of verifying an electronic signature. (defined in the Directive [1], article 2.7)
Split knowledge procedure for key import is a process by which a cryptographic key is split into multiple key components, individually sharing no knowledge of the original key, that can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.
SSCD provision service means a service that prepares and provides a SSCD to subscribers.
User means any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
User data means data created by and for the user that does not affect the operation of the TSF.
Verification authentication data (VAD) means authentication data provided as input by knowledge or authentication data derived from user's biometric characteristics.

1 ST introduction

1.1 ST identification

Title: BULL TrustWay PCI cryptographic card Security Target

Author: Liliana Cabalantti

Version: 3.9 dated March 8th, 2010.

TOE commercial name: BULL TrustWay PCI 2400

TOE version: 76675628-220 (hardware-firmware) S507 and S709 (software)
S507: RSA, SHA-2, AES
S709: ECC, SHA-2, AES

1.2 ST overview

The aim of this document is to describe the Security Target of the BULL TrustWay cryptographic card product.

BULL TrustWay cryptographic card product is intended to be used as a general purpose cryptographic card 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.

1.3 CC conformance

The ST is compliant to Part 2 extended and Part 3 augmented of C.C. v2.3 with interpretation of Protection Profile CWA 14167-2 (Cryptographic Module for CSP Signing Operations with Backup – Protection Profile) Version 0.28 October 27th 2003.

The assurance level for this ST is EAL4, augmented with ADV_IMP.2 (implementation of the TSF), ALC_FLR.3 (systematic flaw remediation), AVA_CCA.1 (vulnerability assessment, covert channel analysis) and AVA_VLA.4 (vulnerability assessment, highly resistant). The minimum strength level for the TOE security functions is 'SOF high' (Strength of Functions High).

The ST is compliant with the Protection Profile CWA 14167-2 (Cryptographic Module for CSP Signing Operations with Backup – Protection Profile) Version 0.28 October 27th 2003

The ST also includes some of the security requirements of Protection Profile CWA 14167-3 (Cryptographic Module for CSP key generation services) Version 0.09 June 3rd 2002.

2 TOE description

2.1 Product type

BULL TrustWay Cryptographic card is a high performance PCI interface based cryptographic token. The product is 100% developed and manufactured in Europe (including the cryptographic components).

BULL TrustWay Cryptographic card provides encryption and digital signature acceleration (among all possible cryptographic operations) combined with a high level of key management and storage.

Operating systems supported are Linux, Windows NT, Windows 2000 and Windows XP. TrustWay Cryptographic card supports both the PKCS#11 API and the CDSA (Common Data Security Architecture) API. This open standards based approach allows an easy integration with industry leading applications and specific developments. The cards have been designed to accommodate future cryptographic algorithms. This technology known as "AES ready" allows a secure update to the on-card software.

2.2 TOE description

2.2.1 Architecture

The technical architecture of the TOE is represented on figure 1. The subsystem performing input/output tasks is indicated in red. The subsystem performing security functions is indicated in black. The two systems are physically and logically distinct from each other.

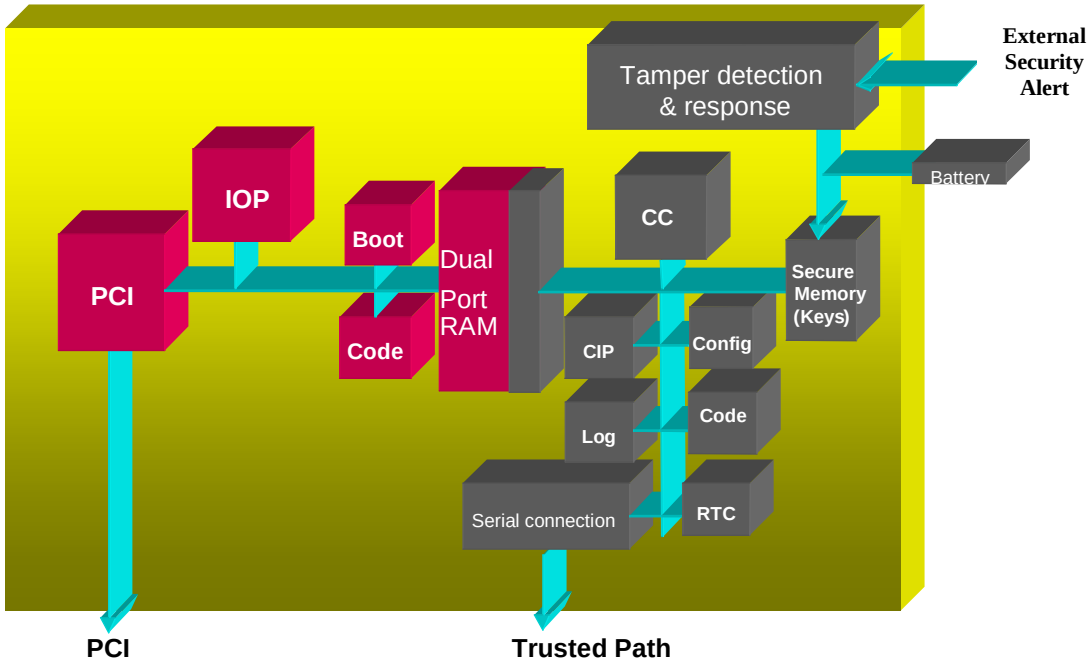
The TOE hardware contains:

- Logic for the PCI interface. The product conforms to the PCI 2.2 specification and occupies one single PCI slot.
- IOP: general purpose Input/Output Processor
- Code: Program memory for IOP
- Dual Port memory
- CC: Cryptographic Component performing cryptographic operations.
- CIP: Control and supervision processor performing, in particular, periodical self-tests.
- Code: Program memory for CIP and CC
- Secure Memory (battery backed up) containing secret data including cryptographic keys
- Serial Connection : Trusted Path for TOE installation, user authentication and keys import
- RTC: Real Time Clock
- LOG: Security Journal Log memory containing audit data
- Config: TOE Configuration memory (non volatile)
- Tamper detection and response logic after alarm detection

The embedded software can be divided in :

- Bios / Boot code / Initial Program Load code
- PKCS#11 application
- PKCS#11 proprietary extensions
- Secure key backup / restore
- Secure software update
- Secure initialisation and configuration
- Secure administration with strong authentication using trusted path with smart card
- Cryptographic configuration data
- Security and control supervision code

Figure 1: TOE Architecture



2.2.2 TrustWay Card Product life-cycle

The TrustWay cryptographic card product life cycle is spit into 5 phases where the following authorities are involved :

Phase		Phase Responsibility	Phase Environment
Phase 1	Token development	**The engineering team (BULL Les Clayes)** is in charge of hardware design and embedded software development.	These phases are performed in the developer environment (developer responsibility)
Phase 2	Token manufacturing and testing	**The manufacturer (SELCO)** is responsible for token manufacturing and testing.	
Phase 3	token personalisation	**The personaliser (BULL ANGERS)** is responsible for the Token personalisation (customisation). The personaliser prepares the token by loading the embedded software.	
Phase 4	Token delivery	**The issuer (BULL ANGERS)** is responsible for the token delivery to the end-user.	
Phase 5a	Secure installation and configuration	The secure installation and configuration are performed by **the end-user (administrator)**.	Theses phases are performed in the user environment (user responsibility)
Phase 5b	Embedded software update	If required, the secure software update of the token is performed by **the end-user (administrator)**.	
Phase 5C	Token use	Token final use is performed by **the end-user**	

The limits of the evaluation process correspond to phases 1 to 4 and do not include the secure installation, configuration and software update process of the token in the user environment.

These different phases may be performed at different sites. This implies that procedures on the delivery process of the TOE exist and are applied for every delivery within a phase or between phases.

2.3 TOE functionalities

2.3.1 Cryptographic operations

BULL TrustWay cryptographic card product is intended to be used as a general purpose cryptographic card that can be used in various cryptographic operations.
The Token supports the following cryptographic algorithms:

- 3DES (S507, S709)
- AES 256 bits (S507, S709)
- RSA 512 to 4096 bits (S507)
- HMAC SHA-1 (S507, S709)
- SHA-2 / HMAC SHA-2 (S507, S709)
- ECC 256 bits (S709)

3DES and HMAC SHA-1 are used internally only (3DES to decrypt the TOE software to be loaded and to encrypt/decrypt backup data, HMAC SHA-1 to assure the integrity of backup data). The other algorithms can be accessed by the user through PKCS#11 API.

The token supports PKCS#11 API including:

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

2.3.2 Key management

BULL TrustWay Cryptographic card provides a high level of key management and storage (see 2.3.5.1 physical security).

Key generation is performed by a hardware based random number generator passing FIPS140-1 and Diehard tests.

Key destruction is performed by zeroisation method.

2.3.3 TOE roles

The TOE supports the following user categories (roles):

- Crypto-officer (authorised to read audit data, authorised to install, configure and maintain the TOE and to create, destruct, backup/restore data, import keys)
- Crypto-user (authorised to perform cryptographic operations)
- Auditor (authorised to read and clear audit data generated by the TOE and exported for audit review in the TOE environment)

Authentication is performed on a trusted path (serial connection with smart card reader) using smart card. Smart cards supported are MPCOS EMV from Gemplus.

2.3.4 Administration

Product administration uses a Smart card based secure solution.

Administration covers :

- Secure embedded software loading process ;
- Secure installation using secret data supplied by two possible modes:
    - Input mode: the administrator enters two 32-digit secrets (the secret number and the authentication secret) which can be chosen freely. Each secret is entered on a trusted path in the form of two 16-digit “half-secrets” by two different administrators ;
    - Smart card mode: the installation procedure is performed under the control of a specific authentication mechanism which reconstructs a shared secret number in sections by reading 3 out of 5 eligible smart cards ;
- Secure token configuration (secure memory partition, access control policy …) ;
- Secure backup and restore of cryptographic keys ;
- Import of cryptographic keys ;
- Tests launching on demand ;
- Audit record export.

2.3.5 Security mechanisms

2.3.5.1 Physical security

The hardware security mechanisms ensure that the card operates properly and protect the integrity of its sensitive data 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 following parameters are monitored:

- Power voltage +5V originating from the PCI bus;
- 3.3V power voltage provided by the card. Should this voltage be interrupted, no error is generated but the power supply of the cryptographic memory is automatically switched to the backup supply provided by the battery.
- The voltage maintained by the battery is constantly monitored.
- The temperature of the physical module is monitored constantly.
- A security alert may be activated by an intrusion detection signal implemented on the server. A security alert can also be activated by pressing a “panic button” located on the outside and linked via a specific connector implemented on the card’s flat band connection strip. Finally, a security alert will also be activated in the event that the PCI connector is unplugged from the card.

Physical protection:
The card is cast in resin with a metal case. All components with the exception of the battery and the connectors are fully encased.

2.3.5.2 Periodical tests

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, RSA and SHA-2 cryptographic operations;
- The random number generator;
- The integrity of the executable code in the card's RAM;
- The integrity of the objects stored in the secure memory.

2.4 TOE Usage

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

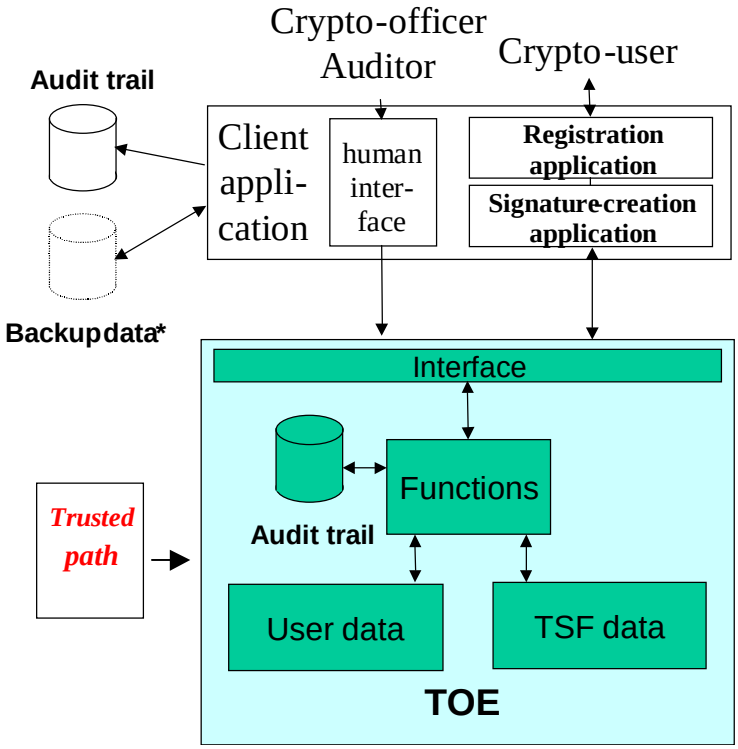


Figure 2: TOE general overview

As shown in figure 2, 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 identification authentication, access control and audit for users of its services. The client application in the TOE environment may mediate the TOE cryptographic (including signing) functions to its end-users. Therefore 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 Crypto-user role. The end-user authenticates himself or herself to the client application with his or her identity. The client application checks the authorisation of the end-user for the TOE cryptographic (including signing) services. If the end-user is allowed to use the cryptographic (including signing) functions the client application will authenticate him or her for the Crypto-user role to the TOE and will map the identity of the end-user to the Crypto-user role. The client application performs identity-based auditing to support accountability for the cryptographic operations. While the TOE will only perform auditing for the client application the TOE environment audit might distinguish between the end-users of the client application.

The TOE provides an appropriate interface and communication path (called trusted path in figures 1 and 2) 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. It can also be used to enter cryptographic keys.

The client application that communicates with the TOE may itself consist of different parts implemented on different systems. For example, a client application that initiates the generation of qualified certificate may consist of two parts:

1. A registration application, which initialises the information for the certificate.
2. A signature-creation application which may be
    a) a certification application, which verifies the integrity and authenticity of the request submitted by the registration application and then calls the TOE service to sign the certificate or
    b) other applications requesting the TOE to sign DTBS-representations, e.g. certificate status information. The application verifies integrity and authenticity of the signature request.

When exporting the CSP_SCD and other cryptographic keys or TSF data the TOE ensures the confidentiality and integrity of the CSP_SCD and other cryptographic keys and the TSF data by the backup and restore functions. The backup will export user data and TSF data (backup data) and the restore function will import the backup data for recreation the state of the TOE at the time the backup was created. The IT-environment provides means to support the backup and restore functions of the TOE by ensuring the availability of the backup data.

3 TOE Security Environment

3.1 Assets to protect

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

TOE services

- **R.SERVICES**: integrity and availability of the TOE services as well as protection against misuse is required.

TOE internal data:

- **R. USER_DATA**: confidential user data (CSP-SCD, other user related secret keys (if any), etc.) and data to be signed with CSP-SCD which has to be protected in integrity.
- **R.USERMGMT_DATA**: non-confidential user / role related data (identifier, access control lists, role definitions, etc.). Those data has to be protected in integrity.
- **R. SYSTEM_DATA**: TSF data (especially VAD and RAD) and other 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.BACKUP**: backup data exported by the TOE to the TOE environment and restored in the TOE. 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.

3.2 Assumptions

A.Admin

Authorized administrators are non-hostile, approprately 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.Audit_Support *CSP audit review*

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 according to the audit procedure of the CSP.

A.Correct_DTBS *Correct DTBS Content Data*

All user data (including DTBS-representation) submitted to the TOE are assumed to be correct. This requires that the user data (e.g. the certificate content data) have been

initialised correctly and maintains this correctness until it is passed to the TOE. For example, 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.Data_Store *Storage and Handling of TOE data*

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

A.User_Authentication *Authentication of Users*

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

Application note:

There are different users of the TOE services within a CSP environment. The TOE itself is only required to relate a request for a TOE service to a specific role and requires credentials to authenticate that the request was generated by a user having a specific role. In the following section we discuss the TOE role model and the users within the TOE environment.

In most cases the registration authority is separated from the certificate generation system. The registration authority system usually has its own protection features including the identification and authentication of individual users (“registration officers”) of the specific registration authority system.

Once the certificate request has been generated on the registration authority system it is submitted to the certificate generation system protected by a digital signature. This digital signature is used by the certificate generation system to verify that the request has been issued by a registration authority authorised to generate certification requests for this certificate generation system.

The registration authority may use its own internal user management and the individual users within the registration authorities may not be known to the certificate generation system and therefore also not known to the TOE. The registration authority may use one specific RA private key to sign a certification request and may use its own internal audit procedures to relate a specific certification request to an individual user within the RA system.

Management of the individual users for the System Administrator and the Crypto Administrator role of the CSP [7] needs to be performed within the TOE as Crypto-officer. The System Auditor [7] will use the TOE Auditor role.

3.3 Threats to Security

T.Bad_SW *Malicious Software during the Lifetime of the TOE*

When the TOE provides the ability to load new software or software updates or modify software when it is in operation, this function can be misused to load malicious software by unauthorised persons.

T.Secure_Human_Interface *Interface with Human Users*

An attacker could manipulate sensitive management data sent by an unauthorized personnel to the TOE within the TOE environment, and thus affect the TOE initialisation or configuration.

T.Keys_Derive *Deriving All or Parts of the* CSP_SCD *and other cryptographic keys*

The most valuable asset the TOE has to protect is the CSP_SCD and other cryptographic keys. The ability to derive all or parts of the CSP_SCD and other cryptographic keys in any way (including the legitimate use of the TOE services) presents a threat that needs to be countered by the TOE. This includes also any ability to derive all or part of the CSP_SCD and other cryptographic keys using knowledge about the CSP_SCD and other cryptographic keys generation and signing or cryptographic operations processes.

T. Keys _Disclose *Disclosing All or Part of the* CSP_SCD *and other cryptographic keys*

Direct disclosure of the CSP_SCD and other cryptographic keys or part of it presents a major threat to the TOE. This includes any way of disclosing all or part of the CSP_SCD and other cryptographic keys over any physical or logical TOE interface.

T. Keys _Distortion *Distortion of the* CSP_SCD *and other cryptographic keys*

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. There is also the danger that by the use of a distorted CSP-SCD, parts of the original CSP-SCD can be derived.

T.Data_Manipul *Manipulating Data outside of the TOE*

User data that is transmitted to the TOE from the client application may be manipulated within the TOE environment before it is passed to the TOE. This may result in the effect that the TOE signs data without the approval of the user under whose control the data is submitted to the TOE. When performed within the client application such manipulations may not be detectable by the TOE itself and therefore this threat needs to be countered within the TOE environment.
Manipulation of data in the TOE environment within the session of a Crypto-officer may also result in a compromise of the security of the TOE. The backup of user data and TSF data these data might be lost.

T.Malfunction *Malfunction of TOE*

Internal malfunction of TOE functions may result in the modification of user data (e.g. DTBS-representation), misuse of TOE services, disclosure or distortion of CSP_SCD and other cryptographic keys or denial of service for authorised users. This includes the destruction of the TOE as well as hardware failures which prevent the TOE from performing its services. This includes also the destruction of the TOE by deliberate action or environmental failure. Technical failure may result in an insecure operational 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. A failure of such a critical hardware component could result in the disclosure or distortion of the CSP_SCD and other cryptographic keys, the modification of user data (e.g. DTBS-representation) or the ability to misuse services of the TOE. 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 CSP_SCD and other cryptographic keys or the user data (e.g. DTBS-representation)
- physical I/O device drivers

T.Insecure_Init *Insecure Initialisation of the TOE*

Unauthorised CSP personnel or authorised CSP personnel without using adequate organisational controls may initialise the TOE with insecure system data, management data or user data.

An attacker may manipulate the backup data to initialise the TOE insecurely by the restore procedure.

T.Insecure_Oper *Insecure Operation of the TOE*

The TOE may be operated in an insecure way not detectable by the TOE itself. This includes the use and operation of the TOE within another environment than the intended one (e. g. the TOE may be connected to a hostile system).

T.Management *Misuse of Management*

CSP personnel may misuse the TOE services to forge user data as CSP_SCD or any other cryptographic keys, user management data, system data or TSF data.

T.Misuse_Operation *Misuse of cryptographic function including signature-creation function*

An user of the client application or of the TOE misuses the TOE service for cryptographic operation, e.g. signature-creation to sign with the SCP-SCD forged qualified certificates or forged certificate status information.

T.Phys_Manipul *Physical Manipulation of the TOE*

An attacker may try to physically manipulate the TOE with the intent to derive all or part of the CSP_SCD or any other cryptographic keys, to manipulate the user data (e.g. DTBS) within the TOE or to misuse services of the TOE. The TOE may be physically attacked by even an authorised user of TOE services.

T.Crypto_Forgery *Forgery of cryptographic results including digital signature*

An attacker exploits weaknesses in the cryptography and/or key management in the TOE in order to forge the output data, e.g. a CSP digital signature in a way that is not detectable by the verifier of the signature.

3.4 Organisational Security Policies

P.Algorithms *Use of Approved Algorithms and Algorithm Parameter*

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. A list of approved algorithms and parameters is given in [5]. Where confidentiality protection is required such as for backup of CSP_SCD and other cryptographic keys, only cryptographic strong algorithms and algorithm parameters shall be used.

P.Keys_Distribution *Import of cryptographic keys*

The TOE must be able to authorize the administrator to import cryptographic keys through the trusted communication path.

4 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.

4.1 Security Objectives for the TOE

O.Audit_CM *Generation and Export of Audit Data*

The TOE shall audit the following events:
- TOE initialisation (including changes to the time)
- TOE start-up
- TOE software update
- Generation of CSP-SCD
- Destruction of CSP-SCD
- Unsuccessful authentication
- Modification of TOE management data
- Modification of functions behaviour
- Adding new users or roles
- Deleting users or roles
- Unsuccessful self test operations
- Execution of the TSF self tests during initial start-up, at the request of the authorised user, at the conditions installation and maintenance
- Reading and deleting audit trail records
- Generation and export of backup data
- Import of backup keys
- Restore of backup data
- Unsuccessful restore attempt

The audit data shall associate each auditable event with the identity of the user that caused the event. The integrity of the audit trail shall be ensured. The TOE shall export the audit data upon request the Auditor and the Crypto-officer. The TOE shall provide the management function for the audit to the Auditor only.

O. Keys _Secure *Secure* CSP_SCD *and other cryptographic keys Generation and Management*

The confidentiality and integrity of the CSP_SCD and other cryptographic keys shall be ensured during their whole lifetime. The TOE shall ensure cryptographic secure CSP_SCD and other cryptographic keys generation, use and management. This includes protection against disclosing completely or partly the CSP_SCD 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 CSP-SCD/CSP-SVD pairs) chosen from [5].

O.Check_Operation *Check for Correct Operation*

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

O.Control_Services *Management and Control of TOE Services*

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

O.Detect_Attack *Detection of Physical Attacks*

The TOE shall detect attempts of physical tampering and securely destroy the CSP_SCD and other cryptographic keys in this case.

O.Error_Secure *Secure State in Case an Error is detected*

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 CSP_SCD and other cryptographic keys.

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

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. Backup and restore shall be audited and the audit data shall associate these events with the identity of the users. The TOE implements secure cryptographic algorithms and parameters for the encryption and data integrity protection chosen from [5].

O.Crypto_Secure *Secure cryptographic operation and advanced signature-creation*

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 CSP-SCD and
- can not be forged without knowledge of the CSP-SCD.

The TOE implements secure cryptographic algorithms and parameters for all cryptographic operations (including the signing operation) chosen from [5].

O.User_Authentication *Authentication of Users interacting with the TOE*

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.

O.Human_Interface *Reliable Human Interface*

The TOE shall supply a trusted communication path with human users physically independent from application path. This trusted path will ensure that the identification, authentication and management data of TOE users, the same as keys imported using this interface are transmitted correctly and in a confidential way to the TOE.

O.Secure_loading *Secure loading of the TOE*

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.

4.2 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

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_Application *Security in the Client 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 can not be performed within the TOE. Security controls in the TOE environment shall also prevent unauthorised manipulation of data submitted to the TOE.

O.ENV_Audit *Audit review*

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_Personnel *Reliable 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 *Prevention of Unauthorised Physical Access*

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

O.ENV_Recovery *Secure Recovery in Case of Major Failure*

Recovery plans and procedures shall exist that allow a secure and timely recovery in the case of a major problem with the TOE (i.e. if TOE is blocked in its secure state after a failure, service discontinuity or detected physical tampering). These procedures shall ensure that the confidentiality and integrity of TOE assets are maintained 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 *Secure Initialisation Procedures*

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 *Secure Operating Procedures*

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.

5 IT 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 5.1 “TOE security functional requirements” are drawn from Common Criteria part 2 [3]. Some security functional requirements represent extensions to [3], with a reasoning given in section 8.5.

The TOE security assurance requirements statement given in section 5.2 “TOE Security Assurance Requirement” are drawn from the security assurance components from Common Criteria part 3 [4].

Section 5.3 identifies the IT security requirements that are to be met by the IT environment of the TOE.

The non-IT environment is described in section 5.4.

5.1 TOE 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 CWA 14167-2 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. Parts of application notes not fitted by the TOE are indicated with crossed characters.

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1)

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) Initialisation of the TOE (including time and date setting),
Start-up after power up,
Shutdown of the TOE,
Software update of the TOE,
Cryptographic key generation (FCS_CKM.1): CSP-SCD/CSP-SVD pair generation,
Cryptographic key distribution (FCS_CKM.2): entry of key(s)
Cryptographic key destruction (FCS_CKM.4): CSP-SCD destruction, destruction of backup key(s)
Backup and recovery (FDP_BKP.1): Use of the backup function, Use of the recovery function, Unsuccessful recovery because of detection of modification of the backup data
Authentication failure handling (FIA_AFL.1): the reaching of the threshold for the unsuccessful authentication attempts and the actions,
Timing of authentication (FIA_UAU.1): all unsuccessful use of the authentication mechanism,

Management of security attributes (FMT_MSA.1) /(all instantiations): all modifications of the values of security attributes,
Management of functions (FMT_MOF.1), modification in the behaviour of the functions,
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;
Management of TSF data (FMT_MTD.1/ACCESS_CONTROL): All modifications to the values of TSF data,
Management of TSF data (FMT_MTD.1/AUDIT: Export of audit data, Clear of audit data,
Abstract machine testing (FPT_AMT.1): Execution of the tests of the underlying machine and the results of the tests,
Failure with preservation of secure state (FPT_FLS.1): Failure detection of the TSF and secure state,
Inter-TSF detection of modification (FPT_ITI.1): The detection of modification of imported backuped TSF data
Notification of physical attack (FPT_PHP.2): Detection of intrusion,
TSF testing (FPT_TST.1): Execution of the TSF self tests during initial start-up, at the request of the authorised user, at the conditions installation and maintenance and the results of the tests, unsuccessful self test operations.

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the ST, identity of the user and sequence data

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

Application note:
The audit data for the Crypto-user role can only identify the client application. Further refinement of audit data might be provided by audit functions in the TOE environment distinguishing between end-users using the services of the client application.

5.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.3 Guarantees of audit data availability (FAU_STG.2/TOE)

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:
The TSF may overwrite the audit trail data after reading (export) by the Auditor.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1/ RSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [RSA bi-keys generation] and specified cryptographic key sizes [512 to 4096 bits (step 256)] that meet the following: FIPS PUB 186-2 REV01 (05/10/2001), appendix 3.1.

FCS_CKM.1.1/ TDES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [TDES key generation] and specified cryptographic key sizes [168 bits] that meet the following: FIPS PUB 186-2 REV01 (05/10/2001), appendix 3.1.

FCS_CKM.1.1/ AES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [AES keys generation] and specified cryptographic key sizes [256 bits] that meet the following: FIPS PUB 186-2 REV01 (05/10/2001), appendix 3.1.

FCS_CKM.1.1/ ECC The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [ECC key generation] and specified cryptographic key sizes [256 bits] that meet the following: FIPS PUB 186-2 REV01 (05/10/2001), appendix 3.1.

FCS_CKM.1.1/ Generic secret The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [Generic secret key generation] and specified cryptographic key sizes [40 to 192 bits] that meet the following: FIPS PUB 186-2 REV01 (05/10/2001), appendix 3.1.

FCS_CKM.1.1/ backup The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [TDES key generation] and specified cryptographic key sizes [168 bits] that meet the following: FIPS PUB 186-2 REV01 (05/10/2001), appendix 3.1. including diversification according to secret number entered at initialisation time.

5.1.2.2 Cryptographic key distribution (FCS_CKM.2/backup_keys)

FCS_CKM.2.1 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 are encrypted using a cryptographic algorithm from the list of approved algorithms and parameters [5]. The key entry are performed using electronic methods.

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. Note that according to FDP_BKP.1.4 the CSP-SCD and other cyptographic keys shall be exported for backup and imported for restore in encrypted form only.

5.1.2.3 Cryptographic key distribution (FCS_CKM.2/other_keys)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [keys are imported using the trusted path] that meets the following: [none].

5.1.2.4 Cryptographic key destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [zeroization] that meets the following: [FIPS140-2, Section 4.7.6, Key Zeroization].

Application note:
The TSF will destroy the CSP-SCD and all other plaintext secret or private keys, if the TSF required by FPT_PHP.2 detects physical tampering.

5.1.2.5 Cryptographic operation (FCS_COP.1/SIGN)

FCS_COP.1.1/ SIGN HMAC SHA The TSF shall perform digital signature-creation in accordance with a specified cryptographic algorithm [HMAC SHA] and cryptographic key sizes [40 to 192 bits] that meet the following: RFC2104.

FCS_COP.1.1/ SIGN HMAC SHA2 The TSF shall perform digital signature-creation in accordance with a specified cryptographic algorithm [HMAC SHA2] and cryptographic key sizes [256 bits] that meet the following: RFC2104.

FCS_COP.1.1/ SIGN RSA The TSF shall perform digital signature-creation in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [512 to 4096 bits (step 256)] that meet the following: PKCS#1.

FCS_COP.1.1/ SIGN SHA2-RSA	The TSF shall perform digital signature-creation in accordance with a specified cryptographic algorithm [SHA2-RSA] and cryptographic key sizes [512 to 4096 bits (step 256)] that meet the following: PKCS#1.
FCS_COP.1.1/ SIGN ECDSA	The TSF shall perform digital signature-creation in accordance with a specified cryptographic algorithm [ECDSA] and cryptographic key sizes [192 to 512 bits, see *Note* below] that meet the following: PKCS#1.

Note:

PCA2 CryptoCard software supports named curves over Fp based domains. Only some ANSI X9.62 and CERTICOM curves are available, these curves have the following ASN1 identifiers :

{ansi-x9-62, 3, 1, 1}

{ansi-x9-62, 3, 1, 2}

{ansi-x9-62, 3, 1, 3}

{ansi-x9-62, 3, 1, 7}

with ansi-x9-62 = { 1, 2, 840, 10045}

{1, 3, 132, 0, 33}

{1, 3, 132, 0, 34}

{1, 3, 132, 0, 35}

5.1.2.6 Cryptographic operation (FCS_COP.1/VERIF)

FCS_COP.1.1/ VERIF HMAC SHA	The TSF shall perform digital-signature verification in accordance with a specified cryptographic algorithm [HMAC SHA] and cryptographic key sizes [40 to 192 bits] that meet the following: RFC2104.
FCS_COP.1.1/ VERIF HMAC SHA2	The TSF shall perform digital-signature verification in accordance with a specified cryptographic algorithm [HMAC SHA2] and cryptographic key sizes [256 bits] that meet the following: RFC2104.
FCS_COP.1.1/ VERIF RSA	The TSF shall perform digital-signature verification in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [512 to 4096 bits (step 256)] that meet the following: PKCS#1.

FCS_COP.1.1/ VERIF SHA2-RSA	The TSF shall perform digital-signature verification in accordance with a specified cryptographic algorithm [SHA2-RSA] and cryptographic key sizes [512 to 4096 bits (step 256)] that meet the following: PKCS#1.
FCS_COP.1.1/ VERIF ECDSA	The TSF shall perform digital-signature verification in accordance with a specified cryptographic algorithm [ECDSA] and cryptographic key sizes [192 to 512 bits, see *Note* below] that meet the following: PKCS#1.

Note:

PCA2 CryptoCard software supports named curves over Fp based domains. Only some ANSI X9.62 and CERTICOM curves are available, these curves have the following ASN1 identifiers :

{ansi-x9-62, 3, 1, 1}

{ansi-x9-62, 3, 1, 2}

{ansi-x9-62, 3, 1, 3}

{ansi-x9-62, 3, 1, 7}

with ansi-x9-62 = { 1, 2, 840, 10045}

{1, 3, 132, 0, 33}

{1, 3, 132, 0, 34}

{1, 3, 132, 0, 35}

5.1.2.7 Cryptographic operation (FCS_COP.1/ENCRYPT)

FCS_COP.1.1/ ENCRYPT TDES	The TSF shall perform encryption in accordance with a specified cryptographic algorithm [TDES ECB and CBC] and cryptographic key sizes [168 bits] that meet the following: FIPS46-3.
FCS_COP.1.1/ ENCRYPT RSA	The TSF shall perform encryption in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [512 to 4096 bits (step 256)] that meet the following: PKCS#1.
FCS_COP.1.1/ ENCRYPT AES	The TSF shall perform encryption in accordance with a specified cryptographic algorithm [AES ECB and CBC] and cryptographic key sizes [256 bits] that meet the following: PKCS#1.

5.1.2.8 Cryptographic operation (FCS_COP.1/DECRYPT)

FCS_COP.1.1/ DECRYPT TDES	The TSF shall perform decryption in accordance with a specified cryptographic algorithm [TDES ECB and CBC] and cryptographic key sizes [168 bits] that meet the following: FIPS46-3.
FCS_COP.1.1/ DECRYPT RSA	The TSF shall perform decryption in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [512 to 4096 bits step 256] that meet the following: PKCS#1.
FCS_COP.1.1/ DECRYPT AES	The TSF shall perform decryption in accordance with a specified cryptographic algorithm [AES ECB and CBC] and cryptographic key sizes [256 bits] that meet the following: PKCS#1.

5.1.2.9 Cryptographic operation (FCS_COP.1/DIGEST)

FCS_COP.1.1/ DIGEST SHA-2	The TSF shall perform digest in accordance with a specified cryptographic algorithm [SHA-2] that meet the following: FIPS180-2.

5.1.2.10 Cryptographic operation (FCS_COP.1/WRAP)

FCS_COP.1.1/ WRAP RSA	The TSF shall perform secret keys wrapping in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [512 to 4096 bits (step 256)] that meet the following: PKCS#1.
FCS_COP.1.1/ WRAP AES	The TSF shall perform private keys wrapping in accordance with a specified cryptographic algorithm [AES] and cryptographic key sizes [256 bits] that meet the following: PKCS#8.

5.1.2.11 Cryptographic operation (FCS_COP.1/UNWRAP)

FCS_COP.1.1/ UNWRAP RSA	The TSF shall perform secret keys unwrapping in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [512 to 4096 bits (step 256)] that meet the following: PKCS#1.

FCS_COP.1.1/ UNWRAP AES	The TSF shall perform private keys unwrapping in accordance with a specified cryptographic algorithm [AES] and cryptographic key sizes [256 bits] that meet the following: PKCS#8.

5.1.2.12 Cryptographic operation (FCS_COP.1/BACKUP_ENC)

FCS_COP.1.1/ BACKUP_ENC	The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm [3DES CBC] and cryptographic key sizes [168 bits] that meet the following: FIPS46-3

5.1.2.13 Cryptographic operation (FCS_COP.1/BACKUP_INT)

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: RFC2104.

5.1.2.14 Quality metrics for random numbers (FCS_RND.1)

FCS_RND.1.1	The TSF shall provide a mechanism for generating random numbers that meet [diehard tests criteria].
FCS_RND.1.2	The TSF shall be able to enforce the use of TSF-generated random numbers for FCS_CKM.1.

5.1.3 User data protection (FDP)

5.1.3.1 Subset access control (FDP_ACC.1/CRYPTO)

FDP_ACC.1.1/ CRYPTO	The TSF shall enforce the Crypto-SFP on User, CSP_SCD, CSP-SVD, Data to be processed including DTBS representation; generate CSP-SCD/CSP-SVD pair, (FCS_CKM.1), destruction of CSP-SCD and CSP-SVD, (FCS_CKM.4); cryptographic operation including sign DTBS representation (FCS_COP.1/all iterations).

5.1.3.2 Subset access control (FDP_ACC.1/AUDIT)

FDP_ACC.1.1/ AUDIT	The TSF shall enforce the Audit-SFP on User; Audit data; export and delete.

5.1.3.3 Subset access control (FDP_ACC.1/BACKUP)

FDP_ACC.1.1/ BACKUP — The TSF shall enforce the Backup SFP on User; CSP-SCD 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).

5.1.3.4 Subset access control (FDP_ACC.1/LOAD)

FDP_ACC.1.1/ LOAD — The TSF shall enforce the load SFP on User; software code, load software update.

5.1.3.5 Subset access control (FDP_ACC.1/KEYS_DISTRIBUTION)

FDP_ACC.1.1/ KEYS_DISTRIBUTION — The TSF shall enforce the Keys distribution SFP on User; CSP-SCD and other cryptographic keys, key entry (FCS_CKM.2/other_keys).

5.1.3.6 Security attribute based access control (FDP_ACF.1/CRYPTO)

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 CSP-SCD and CSP-SVD under dual person control.
(2) User with security attribute Role Crypto-officer is allowed to destruct (FCS_CKM.4) the objects CSP-SCD and CSP-SVD
(3) User with security attribute Role Crypto-officer is allowed to export CSP-SVD.
(4) User with security attribute Role Crypto-user is allowed to perform cryptographic operations and create signature of the DTBS-representation with CSP-SCD (FCS_COP.1/all iterations).
(5) User with security attribute Role Crypto-user is allowed to export CSP-SVD.

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

(a) generate (FCS_CKM.1) the objects CSP-SCD and CSP-SVD
(b) destruct (FCS_CKM.4) the objects CSP-SCD and CSP-SVD

Application note:
The dual person control requires two users to be authenticated with different identities and with the same role Crypto-officer at the same time or at the time authentication data are generated.

5.1.3.7 Security attribute based access control (FDP_ACF.1/AUDIT)

FDP_ACF.1.1/ AUDIT — The TSF shall enforce the Audit-SFP to objects based on 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
  (1) to export Audit data,
  (2) to clear Audit data.
(2) Users with security attribute Role Crypto-officer are allowed to export Audit data

FDP_ACF.1.3/ AUDIT — The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none.

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

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

5.1.3.8 Security attribute based access control (FDP_ACF.1/BACKUP)

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

FDP_ACF.1.2/ BACKUP — The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: User with security attribute Role Crypto-officer is allowed under dual person control

(a) to backup all keys including CSP-SCD and CSP-SVD (FDP_BKP.1),
(b) to restore all keys including CSP-SCD and CSP-SVD (FDP_BKP.1),
(c) to enter backup keys (FCS_CKM.2/backup_keys)

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
(a) to backup CSP_SCD and other cryptographic keys (FDP_BKP.1),
(b) to restore CSP_SCD and other cryptographic keys (FDP_BKP.1),
(c) to enter a backup key (FCS_CKM.2/backup_keys).

5.1.3.9 Security attribute based access control (FDP_ACF.1/KEYS_DISTRIBUTION)

FDP_ACF.1.1/ KEYS_DISTRIBUTION — The TSF shall enforce the Keys distribution SFP to objects based on Identity and Role.

FDP_ACF.1.2/ KEYS_DISTRIBUTION — 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 enter keys (FCS_CKM.2/other_keys)

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

FDP_ACF.1.4/ KEYS_DISTRIBUTION — The TSF shall explicitly deny access of subjects to objects based on the User with security attribute Role Crypto-user is not allowed to enter keys (FCS_CKM.2/other_keys).

5.1.3.10 Security attribute based access control (FDP_ACF.1/LOAD)

FDP_ACF.1.1/ LOAD — The TSF shall enforce the Load-SFP to objects based on 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 Crypto-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
(3) Users with security attribute Role auditor are not allowed to perform software update
(4) Users with security attribute Role Crypto-user are not allowed to perform software update.

5.1.3.11 Backup and recovery (FDP_BKP.1)

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 CSP-SCD and other cryptographic keys 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.

5.1.3.12 Export of user data without security attributes (FDP_ETC.1)

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.

5.1.3.13 Subset information flow control (FDP_IFC.1/BACKUP)

FDP_IFC.1.1/ BACKUP — The TSF shall enforce the Side-channel of backup-functions SFP on Anybody; Information about CSP_SCD and other cryptographic keys;backup (FDP_BKP.1, FCS_COP.1/BACKUP_ENC, FCS_COP.1/BACKUP_INT), restore (FDP_BKP.1, FCS_COP.1/BACKUP_ENC, FCS_COP.1/BACKUP_INT), key entry (FCS_CKM.2/backup_keys).

5.1.3.14 Subset information flow control (FDP_IFC.1/CRYPTO)

FDP_IFC.1.1/ CRYPTO — The TSF shall enforce the Side-channels of Crypto-functions SFP on Anybody; Information about CSP_SCD and other cryptographic keys; generation of keys including CSP-SCD/SVD pair (FCS_CKM.1), destruction of keys including CSP-SCD (FCS_CKM.4), cryptographic operations including signing DTBS-representation (FCS_COP.1/all iterations).

5.1.3.15 Partial elimination of illicit information flows (FDP_IFF.4/BACKUP)

FDP_IFF.4.1/ BACKUP — The TSF shall enforce the Side-channel of backup-functions SFP to limit the capacity of covert channels information flow of

(1) the backup function including encryption of the backup data (FDP_BKP.1),
(2) the backup key(s) entry (FCS_CKM.2/backup_keys),
(3) the encryption and decryption of the backup data (FCS_COP.1/BACKUP_ENC)

through physical behaviour of the TOE interfaces and emanation compromising information about the CSP_SCD and other cryptographic keys to a high resistance level.

FDP_IFF.4.2/ BACKUP — The TSF shall prevent the following types of side-channels information flow within the backup data (FDP_BKP.1) about the CSP_SCD and other cryptographic keys.

Application note:
The TOE shall prevent side-channel attacks against the CSP-SCD and other secret data where the attack is based on external observable physical phenomena of the TOE as mentioned in the application note to FDP_IFF.4/CRYPTO.

The TOE shall prevent side-channel attacks against the CSP_SCD and other cryptographic keys through the intended output data of the TOE e.g. the backup data encrypted with an initial vector containing information about the used backup key.

5.1.3.16 Partial elimination of illicit information flows (FDP_IFF.4/CRYPTO)

FDP_IFF.4.1/ CRYPTO — The TSF shall enforce the Side-channels of Crypto-functions SFP to limit the capacity of side-channels information flow of

(1) the keys (including CSP-SCD/CSP-SVD) generation (FCS_CKM.1),
(2) the cryptographic operations including signature-creation (FCS_COP.1/all iterations),

through physical behaviour of the TOE interfaces and emanation compromising information about the CSP-SCD to a high resistance level.

FDP_IFF.4.2/ CRYPTO — The TSF shall prevent side-channels information flow within the data exported

(1) by the TSF CSP-SCD / SVD pair and other cryptographic keys generation (FCS-CKM.1),
(2) by the TSF cryptographic (including signature-creation) functions (FCS-COP.1) about the CSP-SCD.

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

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

5.1.3.17 Subset residual information protection (FDP_RIP.1)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the de-allocation of the resource from the following objects: CSP_SCD and other cryptographic keys and RAD.

5.1.3.18 Stored data integrity monitoring and action (FDP_SDI.2)

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.

5.1.4 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.

5.1.4.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when [3] unsuccessful authentication attempts occur related to [the crypto-officer 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 is defined with respect to the high strength of the authentication function.

5.1.4.2 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: identity and role.

5.1.4.3 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [secure proprietary mechanism based on HMAC SHA].

5.1.4.4 Timing of authentication (FIA_UAU.1)

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.

5.1.4.5 Timing of identification (FIA_UID.1)

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.

5.1.5 Security management (FMT)

5.1.5.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1/ The TSF shall restrict the ability to disable, enable the functions :

- SF.SL ;
- SF.backup ;
- SF.keys_distribution ;

to Crypto-officer.

5.1.5.2 Management of security attributes (FMT_MSA.1/ROLE_CRYPTO)

FMT_MSA.1.1/ ROLE_CRYPTO The TSF shall enforce the Backup SFP, Load-SFP, Keys distribution-SFP and Crypto-SFP to restrict the ability to query, modify and delete the security attributes Role Crypto-user and Role Crypto-officer to Crypto-officer.

5.1.5.3 Management of security attributes (FMT_MSA.1/ROLE_AUDIT)

FMT_MSA.1.1/ ROLE_AUDIT	The TSF shall enforce the Audit-SFP to restrict the ability to query, modify and delete the security attributes Role Auditor to Auditor.

5.1.5.4 Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for security attributes.

5.1.5.5 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1	The TSF shall enforce the Audit-SFP, Backup SFP, Load-SFP, Keys distribution-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 Auditor and Crypto-officer to specify alternative initial values to override the default values when an object or information is created.

5.1.5.6 Management of TSF data (FMT_MTD.1/ACCESS_CONTROL)

FMT_MTD.1.1/ ACCESS_CONTROL	The TSF shall restrict the ability to query and modify the access control lists to Crypto-officer.

5.1.5.7 Management of TSF data (FMT_MTD.1/USER_CRYPTO)

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 Crypto-officer and Crypto-user to Crypto-officer.

5.1.5.8 Management of TSF data (FMT_MTD.1/USER_AUDIT)

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 to Auditor.

5.1.5.9 Management of TSF data (FMT_MTD.1/RAD)

FMT_MTD.1.1/ RAD	The TSF shall restrict the ability to modify the RAD to User for its own RAD.

5.1.5.10 Management of TSF data (FMT_MTD.1/AUDIT)

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.

5.1.5.11 Specification of Management Functions (FMT_SMF.1)

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).

5.1.5.12 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles Crypto-officer, Crypto-user and Auditor.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application note:
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 singing of DTBS by the HSM.

5.1.6 Protection of the TOE Security Functions (FPT)

5.1.6.1 Abstract machine testing (FPT_AMT.1)

FPT_AMT.1.1 The TSF shall run a suite of tests at the request of an authorised user to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

5.1.6.2 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: failures detected by the TSF FPT_AMT.1 and FPT_TST.1.

Refined by adding:
The TSF shall destroy the plaintext SCP-SCD and other confidential secret and private keys if failures occur.

5.1.6.3 Inter-TSF confidentiality during transmission (FPT_ITC.1)

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.

5.1.6.4 Inter-TSF detection of modification (FPT_ITI.1)

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 alarm indication to the Crypto-officer 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.

5.1.6.5 Notification of physical attack (FPT_PHP.2)

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 local user 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 is given through functional interfaces (stopping any other services but alarm signalisation). The TOE non-IT environment should ensure that notification about physical tampering attempts given by the TOE shall be noticed by the CSP security personnel.

5.1.6.6 Resistance to physical attack (FPT_PHP.3)

FPT_PHP.3.1 The TSF shall resist physical tampering by opening the device, removal or penetration of a cover to the components which
- generates keys (FCS_CKM.1)
- creates the signature with CSP-SCD (FCS_COP.1)
- perform any other cryptographic operation
- stores CSP-SCD
- stores other secret or private keys

by responding automatically such that the TSP is not violated.

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

Application Note:
The TOE protect the confidentiality of the SCP-CSD and other secret and private keys in case of physical maintenance or physical tampering. The TOE will invoke the TSF required by FCS_CKM.4 to destroy the SCP-SCD and all other plaintext secret and private keys. The

destruction of the CSP-SCD will prevent the use of an attacked TOE for signing until restoring the operational state.

5.1.6.7 Manual recovery (FPT_RCV.1)

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.

5.1.6.8 Time stamps (FPT_STM.1)

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

5.1.6.9 TSF testing (FPT_TST.1)

FPT_TST.1.1 The TSF shall run a suite of self-tests during initial start-up, periodically during normal operation and at the request of the authorised user, at the conditions installation and maintenance 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

1. **Initialisation**
   Extended software/firmware integrity test

2. **Power-Up Tests**
   Software/firmware integrity test
   Internal TSF data integrity test.
   Cryptographic algorithm test.
   Random number generator tests
   Critical functions test.

3. **Conditional Tests**
   Pair-wise consistency test (for public and private keys).
   Continuous random number generator test.

5.1.7 Trusted path (FTP)

5.1.7.1 Trusted path (FTP_TRP.1/TOE)

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 or 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), 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) and key entry (FCS_CKM.2/other_keys, FDP_ACC.1/KEYS_DISTRIBUTION, FDP_ACF.1/ KEYS_DISTRIBUTION).

Application Note:
The protection of the communicated data is mainly achieved by the secure IT environment through a local serial connection with the TOE.

5.2 TOE Security Assurance Requirements

Table 5.1 Assurance Requirements: EAL 4 augmented

Assurance Class	Assurance Components
ACM	ACM_AUT.1 ACM_CAP.4 ACM_SCP.2
ADO	ADO_DEL.2 ADO_IGS.1
ADV	ADV_FSP.2 ADV_HLD.2 ADV_IMP.2 ADV_LLD.1 ADV_RCR.1 ADV_SPM.1
AGD	AGD_ADM.1 AGD_USR.1
ALC	ALC_DVS.1 ALC_FLR.3 ALC_LCD.1 ALC_TAT.1
ATE	ATE_COV.2 ATE_DPT.1 ATE_FUN.1 ATE_IND.2
AVA	AVA_CCA.1 AVA_MSU.2 AVA_SOF.1 AVA_VLA.4

5.2.1 Configuration management (ACM)

5.2.1.1 Partial CM automation (ACM_AUT.1)

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

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

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

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

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

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

system.

5.2.1.2 Generation support and acceptance procedures (ACM_CAP.4)

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

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

ACM_CAP.4.3D The developer shall provide CM documentation.

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

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

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

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

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

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

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

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

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

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

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

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

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

5.2.1.3 Problem tracking CM coverage (ACM_SCP.2)

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

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

5.2.2 Delivery and operation (ADO)

5.2.2.1 Detection of modification (ADO_DEL.2)

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

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

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

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

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

5.2.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

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

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

5.2.3 Development (ADV)

5.2.3.1 Fully defined external interfaces (ADV_FSP.2)

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

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

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

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

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

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

5.2.3.2 Security enforcing high-level design (ADV_HLD.2)

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

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

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

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

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

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

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

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

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

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

5.2.3.3 Implementation of the TSF (ADV_IMP.2)

ADV_IMP.2.1D The developer shall provide the implementation representation for the entire TSF.

ADV_IMP.2.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.2.2C The implementation representation shall be internally consistent.

ADV_IMP.2.3C The implementation representation shall describe the relationships between all portions of the implementation.

5.2.3.4 Descriptive low-level design (ADV_LLD.1)

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

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

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

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

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

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

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

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

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

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

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

5.2.3.5 Informal correspondence demonstration (ADV_RCR.1)

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

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

5.2.3.6 Informal TOE security policy model (ADV_SPM.1)

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

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

ADV_SPM.1.1C The TSP model shall be informal.

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

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

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

5.2.4 Guidance documents (AGD)

5.2.4.1 Administrator guidance (AGD_ADM.1)

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

administrative personnel.

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

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

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

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

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

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

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

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

5.2.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1D The developer shall provide user guidance.

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

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

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

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

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

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

5.2.5 Life cycle support (ALC)

5.2.5.1 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1D	The developer shall produce development security documentation.
ALC_DVS.1.1C	The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.2C	The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

5.2.5.2 Systematic flaw remediation (ALC_FLR.3)

ALC_FLR.3.1D	The developer shall provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.3.2D	The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_FLR.3.3D	The developer shall provide flaw remediation guidance addressed to TOE users..
ALC_FLR.3.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.3.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.3.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.3.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR3.5C	The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE..
ALC_FLR3.6C	The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.
ALC_FLR.3.7C	The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
ALC_FLR.3.8C	The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE. .
ALC_FLR.3.9C	The flaw remediation procedures shall include a procedure requiring

timely response for the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

5.2.5.3 Developer defined life-cycle model (ALC_LCD.1)

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

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

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

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

5.2.5.4 Well-defined development tools (ALC_TAT.1)

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

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

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

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

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

5.2.6 Tests (ATE)

5.2.6.1 Analysis of coverage (ATE_COV.2)

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

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

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

5.2.6.2 Testing: high-level design (ATE_DPT.1)

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

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in

accordance with its high-level design.

5.2.6.3 Functional testing (ATE_FUN.1)

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

ATE_FUN.1.2D The developer shall provide test documentation.

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

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

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

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

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

5.2.6.4 Independent testing - sample (ATE_IND.2)

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

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

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

5.2.7 Vulnerability assessment (AVA)

5.2.7.1 Covert channel analysis (AVA_CCA.1)

AVA_CCA.1.1D The developer shall conduct a search for covert channels for each information flow control policy.

AVA_CCA.1.2D The developer shall provide covert channel analysis documentation.

AVA_CCA.1.1C The analysis documentation shall identify covert channels and estimate their capacity.

AVA_CCA.1.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

AVA_CCA.1.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA_CCA.1.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

AVA_CCA.1.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

5.2.7.2 Validation of analysis (AVA_MSU.2)

AVA_MSU.2.1D The developer shall provide guidance documentation.

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

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

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

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

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

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

5.2.7.3 Strength of TOE security function evaluation (AVA_SOF.1)

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

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

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

5.2.7.4 Highly resistant (AVA_VLA.4)

AVA_VLA.4.1D The developer shall perform a vulnerability analysis.

AVA_VLA.4.2D The developer shall provide vulnerability analysis documentation.

AVA_VLA.4.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.4.2C The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.4.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.4.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.4.5C The vulnerability analysis documentation shall show that the search for vulnerabilities is systematic.

AVA_VLA.4.6C The vulnerability analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.

5.3 Security Requirements for the IT Environment

5.3.1 Security audit (FAU)

5.3.1.1 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The IT environment shall provide System auditor of the CSP with the capability to read all audit information produced by the TOE from the audit records.

FAU_SAR.1.2 The IT environment shall provide the audit records in a manner suitable for the user to interpret the information.

5.3.1.2 Protected audit trail storage (FAU_STG.1/ENVIRONMENT)

FAU_STG.1.1/ ENVIRONMENT The IT environment shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2/ ENVIRONMENT The IT environment shall be able to prevent modifications to the audit records.

Application note:
The SFR FAU_STG.1/ENVIRONMENT addresses the protection of the IT environment for the audit trail generated and exported by the TOE.

5.3.2 User data protection (FDP)

The client application shall provide the TOE signing and other cryptographic functions to its authorised end-user only and shall prevent unauthorised transmission and manipulation of sensitive data (including DTBS representation to be signed) by the TOE.

5.3.2.1 Subset access control (FDP_ACC.1/CLIENT)

FDP_ACC.1.1/ CLIENT The IT environment shall enforce the Client application SFP on end-user, Cryptographic module signing and other cryptographic functions, use.

5.3.2.2 Security attribute based access control (FDP_ACF.1/CLIENT)

FDP_ACF.1.1/ CLIENT	The IT environment shall enforce the Client application SFP to objects based on authorisation for Cryptographic module signing and other cryptographic functions.
FDP_ACF.1.2/ CLIENT	The IT environment shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: authorised end-user is allowed to use Cryptographic module signing and other cryptographic functions.
FDP_ACF.1.3/ CLIENT	The IT environment shall explicitly authorise access of subjects to objects based on the following additional rules: none.
FDP_ACF.1.4/ CLIENT	The IT environment shall explicitly deny access of subjects to objects based on the rule: non-authorised end-user is not allowed to use Cryptographic module signing and other cryptographic functions.

Application Note:
The security attribute "authorisation for Cryptographic module signing and other cryptographic functions" is assigned to end-users of the client application with two possible values:

(a) authorised to use Cryptographic module signing and other cryptographic functions,
(b) not authorised to use Cryptographic module signing and other cryptographic functions.

5.3.2.3 Data exchange integrity (FDP_UIT.1)

FDP_UIT.1.1	The IT environment shall enforce the Client application SFP to be able to transmit user data in a manner protected from modification, deletion and insertion errors.
FDP_UIT.1.2	The IT environment shall be able to determine on receipt of user data, whether modification, deletion and insertion has occurred.

Application note:
The user data to be protected by the IT environment are data to be signed by the Cryptographic module.

5.3.3 Identification and authentication (FIA)

The client application shall identify and authenticate its end-user for use of the Cryptographic module services.

5.3.3.1 Timing of authentication (FIA_UAU.1/CLIENT)

FIA_UAU.1.1/ CLIENT	The IT environment shall allow identification (FIA_UID.1/CLIENT) on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2/ CLIENT	The IT environment shall require each user to be successfully authenticated before allowing any other actions on behalf of that user.

5.3.3.2 Timing of identification (FIA_UID.1/CLIENT)

FIA_UID.1.1/ CLIENT	The IT environment shall allow [assignment: none] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2/ CLIENT	The IT environment shall require each user to be successfully identified before allowing any other actions of the IT environment on behalf of that user.

5.3.4 Non-IT requirements

RE.ENV_Personnel *Personnel security measures*
The CSP shall define the obligations and the services of management and operation roles for the TOE. The CSP shall inform and train the personnel for their roles. The CSP shall inform the personnel using the TOE about their civil, financial and legal responsibilities.

RE.ENV_Protect_Access *Physical protection of the TOE*
The CSP shall establish physical and organisational security measures to protect the TOE against modification of TOE hardware, firmware and software. These measures shall restrict the access to the TOE and protected assets to authorised persons. If the TOE detects and notifies about physical tampering the local users shall inform the CSP security staff. The TOE shall not be used until the physical integrity of the TOE is established.

RE.ENV_Recovery *Recovery procedures for the TOE*
The CSP shall define and apply recovery plans and procedures which allow a secure and timely recovery of the TOE operational state. These procedures shall ensure at least
(1) secure initialisation of new TOE devices replacing other TOE devices,
(2) re-initialisation of TOE devices establishing the secure state by the TSF FPT_FLS.1 after detecting failures by the TSF FPT_AMT.1 and FPT_TST.1,
(3) integrity check of the TOE hardware, firmware and software and re-initialisation of TOE devices if the TOE indicates physical tampering by TSF FPT_PHP.2 and destroyed the plaintext SCP-SCD and other confidential secret and private keys by TSF FPT_PHP.3.
To support the TOE backup of the CSP-SCD, other user data and TSF data the CSP will ensure the availability of the backup data and the cryptographic quality, confidentiality and availability of the backup keys.

RE.ENV_Secure_Init *Secure initialisation of the TOE*
The CSP shall define and apply procedures and controls in the TOE environment which allow to securely set-up and initialise the TOE for the generation of CSP-SCD and signatures. This includes
(1) secure installation using secret data supplied by one or several administrators and entered on a trusted path using split knowledge mechanisms,
(2) initialisation of the TOE in the CSP,
(3) the CSP-SCD / CSP-SVD pair generation,
(4) the export of the CSP-SVD by the TOE and the securing the authenticity of the CSP-SVD,
(5) the secure initial configuration of the TSF data user's identity, roles and user authentication information.

RE.ENV_Secure_Oper *Secure operation of the TOE*

The CSP shall define and apply procedures and controls in the TOE environment which allow operating the TOE within a CA system in compliance with the requirements of the EU

directive, the Qualified Certificates Policy for the issued certificates, the secure operation of the client application and the TOE guidance.
The TOE user shall ensure that notification about physical tampering attempts given by the TOE will be noticed by the CSP security personnel.

6 TOE summary specification

6.1 TOE security functions

6.1.1 SF.SL (secure loading)

The executable code is loaded into the TOE via the PCI interface in two cases:
- When customising the TOE;
- When updating the TOE card's software.

The main reason for customising the TOE is to load the binary code required for its operating in its end-user environment;
The update operation involves replacing the binary code in the card by new binary code. This operation is carried out on the host machine of the card by the users themselves (whether or not under the supervision of the crypto-officer). The operation is recorded in the security events log.

6.1.1.1 General mechanism

The principle of the secure loading procedure involves authenticating an RSA signature against a hash of the [encrypted] file containing 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 customisation phase, the TOE's flash memory contains all the binary code that it needs in order to operate.

In order to load code during the customisation or update phases, the following are required:
- An encrypted Code_IOP file containing the binary code to be loaded into the IOP flash memory,
- An encrypted Code_CIP file containing the binary code to be loaded into the CIP flash memory,
- An encrypted Code_SCE file containing the binary code to be loaded into the Cryptographic Component flash memory (optional),
- A file containing the signature of the Code_IOP file provided by the developer, BULL,
- A file containing the signature of the Code_CIP file provided by the developer, BULL,
- A file containing the signature of the Code_SCE file provided by the developer, BULL (optional),
- A versions.dat file indicating the version to be loaded,

6.1.1.2 Signature mechanism used

The signature authentication public key is provided in the signature file in the form of a certificate that must be authenticated by the public key stored in the flash memory.

6.1.2 SF.SI (secure installation)

A software installation of the TOE must be performed 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 in the user's machine. The aim is to initialise the date and time used by audit events time stamping and to insert a number of

private or public elements into the secure memory of the TOE using a so-called Installation Smart Card.

Furthermore, upon installing the TOE, the installer (which can in fact be one or several people) must:

- Either enter manually, when performing the first installation of the TOE, a 32 figure “secret number” (consisting of two 16-figure half-numbers) using the keypad of the Smart Card reader terminal, or authenticate themselves (subsequent installations) by entering the 32-figure sequence of the secret number;
- Or insert 3 authentication Smart Cards (reconstruction of a key split into several fragments by reading 3 out of a possible 5 cards). The algorithm used to calculate the key based on the 3 key fragments is the Shamir algorithm.

The secret number is stored in the EEPROM memory of the TOE and can only be modified upon authentication during a reinstallation; its purpose is to enable:
- The authentication of the installer(s);
- The diversification of user keys (backup keys for example).

In addition to this key, another 32-figure key, the “client authentication secret” is initialised, stored and eventually modified in a similar way. Its role is to enable the generation of a client authentication key used as part of the crypto-officer’s local authentication procedure.

At the end of the installation process, the smart card contains encrypted customer secret data that enables to generate the backup key after token desinstallation or into another token (see SF.backup.command).

The token desinstallation can be performed by an administrative command that empties the secure memory and only saves the audit file and the secret numbers. The administrative and cryptographic commands are prohibited and authentication is impossible. The crypto-officer must install the TOE again to make possible any further TOE utilisation.

To perform any re-installation, the user needs to know the PIN code of the installation smart card and the 32-figure secret number, so the strength level for this security function is SOF high.

6.1.3 SF.keys_distribution

Keys distribution must be done through the trusted communication path. Only secret keys can be entered through the trusted communication path. They are entered in plaintext and they can be entered only by the administrator.

6.1.4 SF.CO (cryptographic operation)

6.1.4.1 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 in appendix 3.1 of revision 1 of the FIPS PUB 186-2 dated 5/10/2001, using the appropriate parameters.

Key-pair consistancy test is performed according to FIPS140-2.

Backup keys are 3DES keys diversified by the secret number entered at initialisation time. They are securely stored on the initialisation smart card and entered again by the crypto-officer at re-installation phase (SF.SI).

The token uses a hardware based random number generator passing Diehard tests.

6.1.4.2 SF.CO.key_destruction

The destruction of the keys complies with FPS140-2, Section 4.7.6, Key zeroization.
The activation of an alarm causes the secure memory to be earthed, which effectively wipes out its contents.
The destruction of a particular key stored in the secure memory takes place by setting the relevant memory location to zero.

6.1.4.3 SF.CO.cryptographic_functions

The TOE implements the following standard cryptographic algorithms:
- The 3DES symmetrical key algorithm. The size of the keys is 168 bits.
- The AES symmetrical key algorithms. The size of the keys is 256 bits.
- The SHA (HMAC SHA) and SHA-2 (HMAC SHA2) hashing algorithms.
- The RSA and ECC asymmetric key algorithm. For RSA, the size of the keys varies between 512 and 4096 bits in steps of 256. For ECC, the size of the keys is 256.

The TOE implements the following cryptographic operations:
- Encryption and decryption of data using the 3DES (ECB and CBC), RSA and AES (ECB and CBC) algorithms,
- Signature and authentication using the HMAC SHA, HMAC SHA2, RSA, SHA-RSA, SHA2-RSA and ECDSA with SHA-2 algorithms,
- Hashing of data using the SHA, SHA-2 algorithm,
- Wrapping and unwrapping of secret keys and private keys.

3DES and HMAC SHA-1 are used internally only (3DES to decrypt the TOE software to be loaded and to encrypt/decrypt backup data, HMAC SHA-1 to assure the integrity of backup data). The other algorithms can be accessed by the user through PKCS#11 API.

Note: No cryptographic operation makes possible to export clear secret data. Secret data are exported from the TOE in encrypted form only (using wrapping mechanisms for example).

6.1.5 SF.backup

6.1.5.1 SF.backup.command

The TOE saves and loads the user's private or secret keys under crypto officer control using one of two possible modes: global mode or unique mode.

Under global mode, the keys saved by the user can be reloaded into any of the user's TOEs (i.e. TOE initialised with the same installation Smart Card) whereas under unique mode, keys that are saved by the user can only be reloaded into the TOE where they were stored in the first place.

The procedure for storing keys is as follows:

The crypto-officer of the TOE requests a key storage operation via his administration tool by defining among other things the storage mode, the final destination of the data (the host machine) and a particular key (if no key is selected, the storage operation will apply to all the keys). Once this operation has been launched and accepted by the TOE, the latter sends the host machine the key or set of keys concatenated to one another in a single data buffer.

6.1.5.2 SF.backup.audit

The key storage and reloading operations are all recorded into the security events log.

6.1.5.3 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 (see SF.SI) 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.

6.1.6 SF.authentification

6.1.6.1 SF.Authentification.Roles

The TOE supports the following user categories :

- Crypto-officer (authorized to read audit data generated by the TOE, authorised to install, configure and maintain the TOE and to create, destruct, backup/restore CSP-SCDs and other cryptographic keys). 254 crypto officers with different identities can be defined. The crypto-officer authenticates with an administration card. The creation of administration card is restricted to the crypto officer at installation time.
- Crypto-user (authorised to perform authorised cryptographic operations). The crypto-user role is associated with only one user : the client application.
- Auditor (authorised to read and clear audit data generated by the TOE and exported for audit review in the TOE environment). The auditor authenticates with an auditor card (created at installation time) handled by the auditor.

6.1.6.2 SF. Authentification.Trusted_Path

The authentication of the crypto-officer or the auditor takes place via the Safepad (Smart Card reader) 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.

6.1.6.3 SF.Authentification.policy

When authentication is required, the safepad asks for user Smart Card and the TOE is blocked in this state.

Authentication consists in verifying that crypto officer or auditor 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.
The crypto-officer presents (to the SafePad) his administration card created upon installation of the CC2000 card. Access control takes place at two levels:
- Authentication of the PIN code of the administration Smart Card;
- Validity check of the administration Smart Card itself.

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 3 tries.
The CC2000 card generates warning messages that are sent to the Safepad :
Bad code Try again (upon first failure)
Bad code Last try (second failure)
Card blocked (third and last failure)
The administration card then becomes unusable.

Failed authentication due to bad administration card

The TOE allows 2 consecutive failed authentication attempts.
The failure sequence is as follows:
- Upon the first failed authentication attempt, the Safepad displays the following message: “Bad authent Remove card”;
- upon the second consecutive failed authentication attempt (regardless of the length of time that has elapsed between the 2 attempts), the Safepad displays: “Bad authent last try” for 2 seconds, followed by “remove card”;
- Upon the third consecutive failed authentication attempt, the Safepad displays the following message: “Bad Authent attempts surpassed” for 2 seconds, followed by “remove card”.

Each of the 3 consecutive failed authentication attempts is recorded in the events log. A counter is used to memorise the failed authentication attempts. The counter is initialised to 0 upon installation of the TOE and is reset (or remains) at 0 upon each successful authentication, unless the counter has logged 3 consecutive failed authentication attempts, in which case the TOE must be reinstalled to reset the counter to 0 and allow new authentication attempts.
The event that is recorded in the log is CCE_FAILED_AUTHENTICATION (0X10).

The authentication smart cards are created at initialisation time. To perform any new initialisation including the creation of an authentication smart card, the user needs to know the PIN code of the installation smart card, the 32-figure secret number and the 32-figure authentication number. If the user is in possession of an authentication smart card, he still needs to know the PIN code of the smart card. So the strength level for this security function is SOF high.

6.1.7 SF.Access_Control

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

The TOE can be configured so as to require either prohibition or supervision by the crypto-officer of :

- Administration functions (loading / reloading of keys, updating the code on the card, etc.),
- The use of certain algorithms;
- Some encryption functions or processes (signature, key generation and destruction, etc.).

The TOE imposes supervision by the crypto-officer and the auditor of the audit file management. The auditor role is created at installation time and cannot be deleted. The crypto officer role (including several identities) is created at installation time or when re installing the TOE. The default role is the user role, i.e. the application.

The TOE is configured via an administration function with or without authentication by the crypto-officer.

There are 2 possible modes for performing operations requiring crypto-officer supervision:

- Operations allowed in a login window. Once the security officer has been authenticated, all operations requiring his supervision are allowed until the officer disconnects;
- Systematic authentication requested for each operation requiring supervision.

6.1.8 SF.audit

6.1.8.1 SF.audit.events

The TOE 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 …).

Each event is described in a fixed format defined by the CC_EVENT structure:
typedef struct CC_EVENT {
CC_CHAR eventNumber;
CC_CHAR eventType;
CC_CHAR infoEventType;
CC_DATE date;}
where:
- *eventNumber* is the order number of the event in the log (upon each new entry the order number is incremented by one),
- *eventType* is a number between 1 and 255 that indicates the event type,
- *infoEventType* contains a description of the particular event that occurred. In particular, this field indicates the identity (crypto officer number) of the crypto officer that is authenticated by the token.
- *date* reflects the internal date and time of the TOE when the event occurred (expressed in terms of the number of seconds since the first of January 2000).

6.1.8.2 SF.audit.file

All security events are recorded in a standard file stored in EEPROM. The audit file can be read via an administration command under auditor or crypto-officer control. When storage exhaustion occurs, the new audit data overwrite the oldest audit data to guaranty service continuity. The auditor is allowed to clear the whole audit file (after extraction).

6.1.9 SF.SM (security mechanisms)

6.1.9.1 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. The following parameters are monitored:
→ Power voltage +5V originating from the PCI bus. This voltage, based on which the secondary voltages of the module are generated is monitored constantly. A security alert is activated in the event that a particular threshold level is exceeded.
→ 3.3V power voltage provided by the card. Although this voltage is considered secondary since it is derived from the main 5V one, it plays a vital role in the module since it powers the cryptographic memory area that contains the keys. Should this voltage be interrupted, no error is generated but the power supply of the cryptographic memory is automatically switched to the backup supply provided by the battery. This switchover takes place whenever the voltage drops below 3 V.
→ The voltage maintained by the battery is constantly monitored. A security alert is activated if the voltage maintained by the battery falls below a minimum threshold level. A warning signal is activated and the low battery LED indicator is switched on (assuming that the main power source is on) whenever the battery voltage drops below 3V.

→ The temperature of the physical module is monitored constantly using two detectors one of which is powered by the module itself with the other not being mains powered. The thresholds for activating a security alert are set by default as follows:

- Powered up: 0°C < T <65°C
- Not powered up (in storage) : -20°C < T < +65°C

These values are in keeping with the parameters defined by the manufacturers of the components of the module.
The components of the module are so-called commercial grade (operating temperature 0°C to 70°C) with the exception of those components that implement the security mechanisms which are industrial grade (-40°C à + 85 °C).
→ A security alert may be activated by an intrusion detection signal implemented on the server. A 3-prong berg connector is provided on the card to support such a function. A security alert can also be activated by pressing a “panic button” located on the outside and linked via a specific connector implemented on the card’s flat band connection strip (microD connector). A 3-prong berg connector is serially mounted to enable the corresponding signal to be invalidated in the event of non-use. Finally, a security alert will also be activated in the event that the PCI connector is unplugged from the card. These security mechanisms are active by default.

Physical protection:
The card is cast in resin with a metal case. All components with the exception of the battery and the connectors are fully encased.

6.1.9.2 SF.SM.tests

6.1.9.2.1 General overview

At power on, all TOE security elements are tested before allowing any other action.

The software security mechanisms involve a set of periodic tests (fully compliant with FPT_TST.1) that constantly monitor the proper operation and integrity of the sensitive functions of the card, to wit:
- The AES, RSA and SHA-2 cryptographic operations;
- The random number generator;
- The integrity of the executable code in the card's RAM;
- The integrity of the objects stored in the secure memory;

This mechanism is handled by the cipher micro-controller under the control of a task management system. Two types of tests are carried out:
- Macro-periodic tests, run at intervals measured in minutes. These comprise tests of the cryptographic applications, the integrity of the code and the control electronics.
- Micro-periodic tests, run at intervals measured in seconds, involving tests of the integrity of the secure memory.

An administrative command makes possible to verify tests activity.

6.1.9.2.2 Tests of the cryptographic applications

The algorithms that are tested are:
- Symmetrical algorithms: AES (encryption / decryption);
- The RSA public key algorithm (encryption / decryption);
- The SHA-2 hashing algorithm.
- The HMAC SHA2 algorithm.

The test process checks the algorithms one by one and hands over process control between each check.

6.1.9.2.3 Tests of the pseudo-random number generator

The CIP requests a 20 Kbit pseudo-random number and carries out all the FIPS 140 tests on this number.

6.1.9.2.4 Tests of the integrity of the on-board code

This involves testing all the code in the respective RAM spaces of the IOP and CIP.
The signature tool creates hash (SHA) of the code (unencrypted) space of the IOP and the CIP. These hashes are appended to the start of the Code_IOP and Code_CIP files and loaded into the TOE.
During an update, the TOE extracts the hashes and stores them in its EEPROM memory.

The test consists of recalculating periodically the hashes of the code space in the IOP and CIP's RAM and comparing the outcome with the initial values.

6.1.9.2.5 Tests of the integrity of the secure memory

Upon creation or modification of any object in the secure RAM, the CIP calculates a checksum based on the attribute and value of the object. The checksum is stored in a new field of the object. The checksums are recalculated by sets of 10 objects and by test process to verify the integrity of the objects stored in the memory (except where an error is detected, in which case all the objects are checked).

6.1.9.3 SF.SM.Alarms

6.1.9.3.1 Error report

The presence of an error or anomaly is reported via an event code recorded into the security log file. The log file can be read once the card has been re-launched.

6.1.9.3.2 Dealing with an error

a) All hardware security alarms trigger the activation of a stop signal in the IOP processor, which immediately grinds to a halt. The cryptographic processors are reset to zero. A memory-retention circuit powered by the battery will detect the occurrence of an alarm if the card is not powered up (a logging procedure is implemented for each type of alarm). Finally, the secure memory is earthed, which results in its contents being wiped out. The activation of a security alert by the hardware monitoring systems is processed as follows by the software. Upon the next activation of the card, the IOP checks the security alert status register. If any one of the security alerts has been activated (e.g.: INTS = intrusion detected), the following action is taken:
- The TOE writes the event code to the log file,
- The IOP resets the status register to zero (reset_security command),
- The TOE switches to de-installed mode.

Concerning the periodical tests, there are several options depending on which test threw up the error:

b) In the case of a malfunction of a cryptographic algorithm or should the integrity of the code have been compromised, the following action is taken:
- The TOE writes the event code to the log file,
- The TOE activates the flashing red LED (ERROR LED),
- The cryptographic applications and the CIP are halted.

c) In the case of an error such as CCE_INTEGRITY_MEM_SEC (relating to the integrity of a key in the secure memory), the following action is taken:
- The CIP labels the key with an INVALID_KEY attribute,
- The TOE writes the event code to the log file,
- The IOP is notified that the key can no longer be used except for the purpose of its own destruction by the application. If the application attempts to use the key in any other way, the IOP returns a CKR_KEY_HANDLE_INVALID error code,
- Upon detection of a faulty key, the entire secure memory is checked by the CIP which only hands over process control upon completion of the test.

d) In the case of an error such as CCE_FAILED_TEMP, the following action is taken:
- The TOE writes the event code to the log file,

- The TOE activates the flashing red LED (ERROR LED),
- The CIP and the cryptographic applications are halted only if both detectors report the same fault.

6.1.9.3.3 Leds meaning

The TOE provides a visual indication of the status of its operations and internal security. The visual indication is provided via 4 status LEDs whose meaning is described in the table below.

LED	COLOUR	USAGE
READY	Green	Indicates the status of the card: - OFF: card not initialised - ON: automatic internal tests currently under way or software in the process of being updated - FLASHING: card initialised (internal tests completed, PCI bus connected)
ERROR	Red	Indicates the presence of power-on tests errors - OFF: no errors reported - ON: error(s) reported by the internal tests - FLASHING: error(s) reported by the periodic tests
BATTERY	Yellow	Indicates the battery power status - OFF: normal power rating - ON: battery power low
ALARM	Yellow	Indicates the security alert status - OFF: no security alert - ON: security alert activated

6.2 Assurance measures

Appropriate assurance measures are employed to satisfy the security assurance requirements. The evaluation will confirm whether the assurance measures are sufficient to satisfy the assurance requirements. The assurance measures consist of the set of evaluation evidence listed in Table 6.1, below. The documents listed in the table will be used as to satisfy assurance evaluation requirements.

Table 6-1 Assurance evaluation evidence

Components	**Documents**
ACM_AUT.1	Environnement de développement de la carte PCA2 v7.0
ACM_CAP.2	Procédure documentation pour les cartes PCA2 v10.0 Environnement de développement de la carte PCA2 v7.0 Processus de développement de la carte PCA2 v6.0
ACM_SCP.2	Environnement de développement de la carte PCA2 v7.0 Processus de développement de la carte PCA2 v6.0
ADO_DEL.2	Processus de développement de la carte PCA2 v6.0

	Dispositions sécuritaires requises pour la fabrication des cartes PCA2 et PCA3 v1.4 Dispositions sécuritaires requises pour l'intégration des cartes PCA2 et PCA3 v4.0 Personnalisation des cartes PCA2 à l'usine d'Angers v7.0
ADO_IGS.1	API d'administration de la carte TrustWay PCI v00 Guide d'installation de la carte TrustWay PCI v01 Guide d'utilisation de la carte TrustWay PCI v02
ADV_FSP.2	Spécifications fonctionnelles de sécurité de la carte PCA2 v1.6 Chargement sécurisé de la carte PCA2 v1.2
ADV_HLD.2	Document chapeau HLD et LLD de la carte PCA2 v1.5 Spécifications matérielles phase 2 v2.0
ADV_IMP.2	Implémentation de la carte PCA2 v1.5
ADV_LLD.1	Document chapeau HLD et LLD de la carte PCA2 v1.5 Visibilité micrologicielle phase 2 v2.0 Architecture du logiciel IOP v1.6 Architecture du logiciel CIP v2.6 Spécification des échanges entre les processeurs IOP et CIP v1.3
ADV_RCR.1	Spécifications fonctionnelles de sécurité de la carte PCA2 v1.6 Document chapeau HLD et LLD de la carte PCA2 v1.5 Implémentation de la carte PCA2 v1.5
ADV_SPM.1	Modélisation de la politique de sécurité de la carte PCA2 v2.3
AGD_ADM.1	API d'administration de la carte TrustWay PCI v00 Guide d'installation de la carte TrustWay PCI v01 Guide d' d'utilisation de la carte TrustWay PCI v02
AGD_USR.1	Interface PKCS#11 de la carte TrustWay PCI v2.0
ALC_DVS.1	Environnement de développement de la carte PCA2 v7.0
ALC_FLR.3	Processus de développement de la carte PCA2 v6.0
ALC_LCD.1	Processus de développement de la carte PCA2 v6.0
ALC_TAT.1	Environnement de développement de la carte PCA2 v7.0
ATE_COV.2	Tests de la carte PCA2 v1.6 Validation de la carte PCA2 v1.1
ATE_DPT.1	Tests de la carte PCA2 v1.6 Validation de la carte PCA2 v1.1
ATE_FUN.1	Tests de la carte PCA2 v1.4 Tests des cartes TrustWay PCI v1.3 Tests des cartes TrustWay crypto PCI v1.0
ATE_IND.2	Performed by the evaluator
AVA_CCA.1	Analyse des vulnérabilités de la carte PCA2 v1.2
AVA_MSU.2	Analyse des vulnérabilités de la carte PCA2 v1.2 API d'administration de la carte TrustWay PCI v00 Guide d'installation de la carte TrustWay PCI v01 Guide d'utilisation de la carte TrustWay PCI v02
AVA_SOF.1	Analyse des vulnérabilités de la carte PCA2 v1.2
AVA_VLA.4	Performed by the evaluator

7 PP claims

7.1 PP reference

The ST is compliant with the Protection Profile CWA 14167-2 (Cryptographic Module for CSP Signing Operations with Backup – Protection Profile) Version 0.28 October 27th 2003

The ST also includes most of the security requirements of Protection Profile CWA 14167-3 (Cryptographic Module for CSP key generation services) Version 0.09 June 3rd 2002.

7.2 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.

The following assumption has been added to the PP :
- A_Admin

The following threat has been added to the PP :
- T_Secure_Human_Interface

The following organisational security policy has been added to the PP :
- P.Keys_Distribution

The following security objectives have been added to the PP :
- O.Human_Interface
- O.Secure_loading

The following security objective for the environment has been added to the PP :
- O.ENV_Admin

The following IT security requirements have been added to the PP :
- FCS_COP.1/VERIF
- FCS_COP.1/ENCRYPT
- FCS_COP.1/DECRYPT
- FCS_COP.1/DIGEST
- FCS_COP.1/WRAP
- FCS_COP.1/UNWRAP
- FDP_ACC.1/LOAD
- FDP_ACF.1/LOAD
- FDP_ACC.1/KEYS_DISTRIBUTION
- FDP_ACF.1/ KEYS_DISTRIBUTION
- FMT_MOF.1
- FPT_STM.1

ALC.FLR.3 (Systematic flaw remediation) security assurance requirement has been added to the PP.

CWA 14167-2 Protection Profile considers a TOE without any trusted path. The client application will provide a human interface between human users and the TOE in order to transfer authentication and management data (O. ENV_Human_Interface). This security objective for the environment maps A.Human_Interface assumption imposing an appropriate interface provided by the application to transmit identification, authentication and management data of TOE users correctly and in a confidential way to the TOE. This objective is covered by FTP_TRP.1/CLIENT SFR.
The TOE related to this security target has a trusted path physically independent from application path. This trusted path ensures secure transmission of the identification, authentication and management data. Therefore A.Human_Interface assumption has been replaced by T.Secure_Human_Interface threat imposing a trusted path for all management issues. This threat is mapped by a new security objective for the TOE (O.Human_Interface) replacing O. ENV_Human_Interface in the original PP. The objective is covered by FTP_TRP.1/TOE SFR (present in the original PP). FTP_TRP.1/CLIENT SFR is then useless and has been suppressed.

8 Rationale

8.1 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 8.2.1 “Security Objectives Coverage” and 8.3.1 “Security Requirement Coverage” provide the mapping of the security objectives and security requirements for these TOE types.

8.2 Security Objectives Rationale

8.2.1 Security Objectives Coverage

Table 8-1 Security Environment to Security Objectives Mapping

Policy/Threat/Assumptions	Objectives
Policies	
P.Algorithms	O.Keys_Secure, O.Crypto_Secure, O.Protect_Exported_Data
P. Keys_Distribution	O.Human_Interface
Threats	
T.Bad_SW	O.Check_Operation, O.Control_Services, O.Secure_loading
T.Secure_Human_Interface	O.Human_Interface, O.ENV_Secure_Init
T.Keys_Derive	O.Keys_Secure, O. Crypto _Secure, O.ENV_Protect_Access
T.Keys_Disclose	O.Keys_Secure, O.Check_Operation, O.Protect_Exported_Data, O. Crypto _Secure, O.ENV_Protect_Access
T.Keys_Distortion	O.Check_Operation, O.Detect_Attack, O.Error_Secure, O.Protect_Exported_Data, O.ENV_Protect_Access
T.Data_Manipul	O.ENV_Application, O.ENV_Secure_Oper
T.Insecure_Init	O.Audit_CM, O.Keys_Secure, O.Control_Services, O.Protect_Exported_Data, O.ENV_Application, O.ENV_Personnel, O.ENV_Protect_Access, O.ENV_Recovery, O.ENV_Secure_Init
T.Insecure_Oper	O.ENV_Personnel, O.ENV_Protect_Access, O.ENV_Secure_Oper, O.Human_Interface
T.Malfunction	O.Check_Operation, O.Error_Secure, O.ENV_Protect_Access, O.ENV_Recovery
T.Management	O.Audit_CM, O.Control_Services, O.Protect_Exported_Data, O.User_Authentication, O.ENV_Personnel, O.ENV_Protect_Access, O.ENV_Secure_Oper
T.Misuse_Operation	O.Audit_CM, O.Control_Services, O.User_Authentication, O.ENV_Application
T.Phys_Manipul	O.Check_Operation, O.Detect_Attack, O.Error_Secure, O.ENV_Protect_Access
T.Crypto_Forgery	O. Crypto _Secure
Assumptions	
A.Admin	O.ENV_Admin
A.Audit_Support	O.ENV_Audit, O.ENV_Personnel

A.Correct_DTBS	O.ENV_Application, O.ENV_Secure_Oper
A.Data_Store	O.ENV_Recovery, O.ENV_Secure_Init, O.ENV_Secure_Oper
A.User_Authentication	O.ENV_Application, ~~O.ENV_Human_Interface~~

Table 8-2 Tracing of Security Objectives to the TOE Security Environment

Objectives	Policy/Threat/Assumptions
Security Objectives for the TOE	
O.Audit_CM	T.Insecure_Init, T.Management, T.Misuse_Operation
O.Keys_Secure	P.Algorithms, T.Insecure_Init, T.Keys_Derive, T.Keys_Disclose
O.Check_Operation	T.Bad_SW, T.Keys_Disclose, T.Keys_Distortion, T.Malfunction, T.Phys_Manipul
O.Control_Services	T.Bad_SW, T.Insecure_Init, T.Management, T.Misuse_Operation
O.Detect_Attack	T.Keys_Distortion, T.Phys_Manipul
O.Error_Secure	T.Keys_Distortion, T.Malfunction, T.Phys_Manipul
O.Protect_Exported_Data	P.Algorithms, T.Keys_Disclose, T.Keys_Distortion, T.Insecure_Init, T.Management
O. Crypto _Secure	P.Algorithms, T.Keys_Derive, T.Keys_Disclose, T. Crypto_Forgery
O.User_Authentication	T.Management, T.Misuse_Operation
O.Human_Interface	T.Secure_Human_Interface, T.Insecure_Oper, P.Keys_Distribution
O.Secure_loading	T.Bad_SW
Security Objectives for the Environment	
O.ENV_Admin	A.Admin
O.ENV_Application	A.Correct_DTBS, A.User_Authentication, T.Insecure_Init, T.Data_Manipul, T.Misuse_Operation
O.ENV_Audit	A.Audit_Support
O.ENV_Personnel	A.Audit_Support, T.Insecure_Init, T.Insecure_Oper, T.Management
O.ENV_Protect_Access	T.Insecure_Init, T.Insecure_Oper, T.Malfunction, T.Management, T.Phys_Manipul T.Keys_Derive, T.Keys_Disclose, T.Keys_Distortion
O.ENV_Recovery	A.Data_Store, T.Insecure_Init, T.Malfunction
O.ENV_Secure_Init	A.Data_Store, T.Secure_Init, T.Secure_Human_Interface
O.ENV_Secure_Oper	A.Correct_DTBS, A.Data_Store, T.Data_Manipul, T.Insecure_Oper, T.Management

8.2.2 Security Objectives Sufficiency

The overall objective of this Security target is to provide a basis for cryptographic devices used within a CA environment to store and apply the private keys of a CA to sign certificates, certificate revocation lists, time stamp certificates or OCSP responses. Basic requirements for such a device are defined in the EU directive [1] as well as in the ETSI document on

policy requirements for certification authorities issuing qualified certificates [6]. In addition the objectives of FIPS 140-2 for cryptographic modules have been taken into account.
In this chapter we will map the security objectives, threats and assumptions on the requirements stated in those documents to demonstrate compliance with the EU directive. In addition we will present the arguments for the consistency of the objectives, assumptions and threats defined.

8.2.2.1 Policies and Security Objective Sufficiency

P.Algorithms addresses the problem to use cryptographic algorithms and parameters that provide the required level of security against cryptographic attacks resulting in the ability to generate false signatures for example. These properties are addressed in the objectives O.Keys_Secure, O. Crypto _Secure and O.Protect_Exported_Data.

P.Keys_Distribution addresses the possibility offered by the TOE to import cryptographic keys through the trusted communication path. This possibility is addressed in the objective O.Human_Interface.

8.2.2.2 Threats and Security Objective Sufficiency

T.Bad_SW deals with the threat of introducing potentially malicious or faulty code into the TOE after it has been checked and released for use. Not all CSP signing devices may provide a capability to modify the operational software in those stages of the life-cycle, but many CSP signing devices may provide the ability to install software updates. In this case O.Control_Services will ensure that only authorised users can perform such an update. O.Check_Operation detects unauthorised software changes by means of integrity checks of TOE software and firmware during initial start-up, at the request of the authorised user, at the conditions installation and maintenance. O.Secure_loading ensures that the TOE provides a secure loading process.

T.Secure_Human_Interface deals with the threat of disclosing or corrupting management data during transmission between human user and the TOE. O.Human_Interface will ensure that all sensitive authentication and management data will be transmitted in a secure way using a dedicated trusted path. O.ENV_Secure_Init imposes that the trusted path must be used at initialisation time.

T.Keys_Derive deals with the threat that the CSP-SCD and other cryptographic keys can be derived from the reaction and responses of the CSP signing device. This includes any type of covert storage channel which can be used to extract information about the CSP-SCD and other cryptographic keys as well as the problem of timing channels or other signals of the CSP signing device, for example, that may carry information about the CSP-SCD. Examples are power consumption or radiation.
O.Keys_Secure is responsible to ensure that no information about the CSP-SCD and other cryptographic keys is directly transmitted to any entity outside the TOE. O. Crypto _Secure ensures that the algorithms and the specific implementation will not reveal the CSP-SCD. Leakage of information via e. g. the power consumption or via radiation may require sufficient physical protection of the CSP signing device in its operational environment, which is addressed by O.ENV_Protect_Access.

T.Keys_Disclose deals with the threat of disclosing directly all or part of the CSP-SCD and other cryptographic keys via the defined interfaces. This may happen, for example, either because a defined function allows the unencrypted export of CSP-SCD, the CSP-SCD is not protected sufficiently when exported because of the incorrect operation of an element of the TOE. Unencrypted export of the CSP-SCD is prohibited by O.CSP-SCD_Secure and O.Protect_Exported_Data, and the incorrect operation is addressed by O.Check_Operation.

In addition O. Crypto _Secure ensures that the CSP-SCD is not disclosed as part of the signed data exported to the user.
Physical, logical and organisational protection measures addressed by O.ENV_Protect_Access strengthen the prevention of CSP-SCD disclosure by tampering.

T.Keys_Distortion deals with the threat that the CSP-SCD and other cryptographic keys gets corrupted either by a software or hardware malfunction or by a deliberate physical attack on the TOE. This threat is only relevant, if the TOE will use the distorted keys (e.g. CSP-SCD). Therefore it has to be the objective to detect the distortion of the CSP-SCD and other cryptographic keys, not only to prevent such a distortion.
O.Check_Operation will ensure that the TOE will check the CSP-SCD and other cryptographic keys regularly. O.Error_Secure will prevent the TOE to use distorted keys (including CSP-SCD) after it has detected the distortion and O.Detect_Attack will prohibit the use of a distorted key (e.g. CSP-SCD) after a physical attack (of course in the case of a physical attack the TOE will itself destroy the CSP-SCD and other cryptographic keys and enter a state where it can only be reused after a secure re-initialisation).
O.Protect_Exported_Data addresses the integrity and confidentiality protection measures to CSP-SCD and other cryptographic keys when they are exported from the TOE e.g. for the purpose of backup and restore.
Physical, logical and organisational protection measures addressed by O.ENV_Protect_Access strengthen the prevention of CSP-SCD and other cryptographic keys distortion by tampering.

T.Data_Manipul deals with the threat that data to be signed is manipulated before it is submitted to the TOE. As a result the TOE may sign false certificates or certificate status information. This threat does not address manipulations the TOE is able to detect (e. g. data protected by secure checksums or digital signatures). Instead it addresses the threat of false data to be signed generated by those system components that are allowed to generate data to be signed. An example is a Registration Authority where an authorised operator has made a mistake in defining the certificate content data. Another example is a directory service generating wrong certificate status information which is then submitted to the TOE for signing. This threat has to address in the TOE environment by the objective O.ENV_Secure_Oper and O.ENV_Application.

T.Insecure_Init deals with the threat of a CSP signing device initiated in an insecure way. Each CSP signing device will need to be initialised correctly and in a secure way before it can be used within a CA environment for issuing and managing qualified certificates. Secure initialisation includes the secure generation or import of the CA keys as well as the secure setup of the CSP signing device TSF management data. This threat is countered by O.Keys_Secure with respect to the secure CSP-SCD and other cryptographic keys generation and management, O.Control_Services with respect to the unauthorised use of services (also in the initialisation phase) as well as by objectives on the TOE environment O.ENV_Secure_Init and O.ENV_Recovery. In addition O.Audit_CM provides the ability to check if the initialisation process has been performed correctly.
Procedures within the TOE environment have to be in place that monitor the correct initialisation of the TOE before it is accepted to sign qualified certificates or certificate status information. To counter this threat, organisational controls addressed by O.ENV_Recovery shall be in place. O.ENV_Recovery covers the case where a CSP signing device has to be initialised to take over the task of another CSP signing device e. g. in the case this device works incorrectly.
In addition, applications running on systems within the TOE environment have to perform the necessary checks within the initialisation procedure e. g. if those applications generate data that is then downloaded to the TOE and used there as TSF data. O.ENV_Protect_Access addresses the aspect of physical access to an un-initialised TOE by unauthorised personnel, O.ENV_Secure_Init addresses the organisational aspects while O.ENV_Application

addresses the aspect of security checks and controls within the applications used in the TOE environment for the initialisation of the TOE. In addition, the personnel performing the initialisation actions must be aware of the implications of their activities and trained to perform their task correctly. This is covered by the objective O.ENV_Personnel.
A TOE may also be initialised to be 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.Insecure_Oper deals with the threat that the TOE might be operated in an insecure way and where the TOE itself is not able to detect this. This includes the possibility to operate the TOE in a hostile system that simulates the intended system environment or a valid system environment is operated without in violation of the requirements stated in the EU directive, national laws or regulations. This threat is addressed by the objective O.ENV_Secure_Oper and by the objective O.Human_Interface imposing secure local administrative operations independent from the environment. Physical protection of the TOE, which is also necessary to operate the TOE securely, is addressed by O.ENV_Protect_Access. In addition all personnel performing operational activities with the TOE or within the TOE environment must be aware of their duties and responsibilities and must be trained to perform their actions in accordance with the defined procedures. This is addressed by the objective O.ENV_Personnel.

T. Malfunction deals with the threat that a failure may prohibit the TOE to operate correctly. Examples are faults within hardware components of the TOE, loss or corruption of programs and/or data within the TOE due to component failures or ageing, accidental or deliberate destruction of the TOE or its components As a result the DTBS-representation, the CSP-SCD or TSF management data may be corrupted or the result of TOE operations may be false. As a consequence CSP-SCD may be disclosed or distorted data may be signed by the TOE. This threat is countered by O.Check_Operation and O.Error_Secure (which ensures that the TOE will not continue to operate with the CSP-SCD when it has detected a malfunction). Due to the criticality of the TOE and the requirement for resistance to physical attacks, maintenance of the TOE is also critical and repairing the TOE might be impossible without deleting the CSP-SCD. Therefore the TOE should be protected as far as possible from defects caused by deliberate or accidental mishandling (this is covered by the objective O.ENV_Protect_Access). On the other hand, if a defect occurs procedures within the TOE environment have to exist that allow the organisation operating the TOE to recover in a secure way from this defect. This is covered by the objective O.ENV_Recovery.
This security target does not state specific details of the recovery procedure, because the requirements on this procedure depend on the overall requirements and architecture of the system where the TOE is used to sign qualified certificates or certificate status information.

T.Management deals with the threat of misuse TOE management functions during initialisation and operation. The only way the TOE can deal with this threat is by restricting the use of TOE management functions to users authorised to use those functions and by auditing the actions of those users. Therefore the threat is countered by O.Control_Services, which restricts the use of TOE management functions to authorised users, O.User_Authentication, which ensures that the invoking a management function has the authorisation and O.Audit_CM, which allows to trace the actions of those users. In addition the objective O.Protect_Exported_Data prohibits the modification of data exported by the TOE when it is imported again (which otherwise could be used to manipulate TSF management data).
The TOE environment will limit the access to the TOE to authorised personnel only according to O.ENV_Protect_Access. Because of O.ENV_Personnel this personnel will be aware of their responsibility to manage the TOE securely as addressed by O.ENV_Secure_Oper.

T.Misuse_Operation deals with the threat of misuse of the TOE to create a forged signature for example. This could be achieved, if an unauthorised user could invoke the signature function. O.Control_Services counters this threat for the user known to the TOE. O.User_Authentication prevents the misuse by persons not authorised to use the TOE and O.Audit_CM allows checking, if an unauthorised user has attempted to get access to the TOE or if an authorised user has attempted to misuse the TOE by attempting to use functions he is not allowed to use. O.ENV_Application extents this protection to the end-users of the client application by their user authentication and access control.

T.Phys_Manipul deals with physical manipulation of the TOE. An attacker may try to get access to the CSP-SCD and other cryptographic keys by trying to get physical access to the location where it is stored. O.Detect_Attack counters this threat as long as the TOE is directly able to detect that it is under attack. This includes manipulation by authorised users. O.Check_Operation counters the case where the TOE does not detect the physical manipulation directly but detects an error during operation that might have been caused by a physical attack. O.Error_Secure enforce a secure state of the TOE if such error is detected. Since it is obvious that the TOE is not able to withstand all kind of physical manipulation, O.ENV_Protect_Access shall prohibit (as far as possible) the likelihood that an attacker is able to perform any physical manipulation on the TOE.

T. Crypto_Forgery deals with the threat that an attacker is able to generate a forged cryptographic result, e.g. 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 the CSP-SCD is covered elsewhere, this threat deals with the problem that it might be able for someone to forge a signature without knowledge of the CSP-SCD. O. Crypto _Secure counters this threat by stating that it should not be possible to generate a valid signature without knowledge of the CSP-SCD.

8.2.2.3 Assumptions and Security Objective Sufficiency

A.Admin is met by the objective OE.Admin, which ensures that the administrators are non-hostile and approprately trained.

A.Audit_Support is addressed by the objective O.ENV_Audit, which ensures that the audit trail (generated and exported by the TOE) is properly analysed. The personnel performing this analysis must be aware of their duties and responsibilities, which is addressed by the objective O.ENV_Personnel.

A.Correct_DTBS is addressed by the objective O_ENV_Application ensures that the applications that use the TOE will perform the required checks on the data they pass to the TOE. O.ENV_Secure_Oper ensures that the necessary operational procedures are in place for the organisation operating the TOE as part of their certification system. With the sum of these objectives the assumption is covered.

A.Data_Store is addressed by the objectives O.ENV_Secure_Init and O.ENV_Secure_Oper, which deals with the security of data necessary for secure initialisation and operation of the TOE if they are stored in the TOE environment. In addition O.ENV_Recovery addresses the availability of data stored in the TOE environment.

A.User_Authentication deals with the authentication function of the client application for its end-users gaining access to the TOE signing function. O.ENV_Application addresses the TOE environment task to support the authentication of an individual end-user outside of the TOE (e. g. within the system of a registration authority).

Note in contrast to O.ENV_Application the objective O.User_Authentication addresses the direct authentication of the Crypto-officer and Auditor by the TOE as individual users.

8.3 Security Requirements Rationale

8.3.1 Security Requirement Coverage

Table 8-3 Functional and Assurance Requirement to Security Objective Mapping

Objectives	Requirements
Security Objectives for the TOE	
O.Audit_CM	FAU_GEN.1, FAU_GEN.2, FAU_STG.2/TOE, FDP_ACC.1/AUDIT, FDP_ACF.1/AUDIT, FMT_MTD.1/AUDIT, FMT_SMF.1, FPT_ITI.1, FPT_STM.1
O.Protect_Exported_Data	FAU_GEN.1, FAU_GEN.2, FCS_CKM.1 (backup), FCS_CKM.2/backup_keys, FCS_COP.1/BACKUP_ENC, FCS_COP.1/BACKUP_INT, FDP_ACC.1/BACKUP, FDP_ACF.1/BACKUP, FDP_BKP.1, FDP_ETC.1, FDP_IFC.1/BACKUP, FDP_IFF.4/BACKUP, FMT_MSA.1/ROLE_CRYPTO, FMT_MSA.3, FPT_ITC.1, FPT_ITI.1, AVA_CCA.1, FMT_MOF.1,
O.Keys_Secure	FCS_CKM.1, FCS_CKM.4, FCS_COP.1/all iterations, FCS_RND.1, FDP_ACC.1/CRYPTO, FDP_ACF.1/CRYPTO, FDP_BKP.1, FDP_IFC.1/CRYPTO, FDP_IFF.4/CRYPTO, FDP_RIP.1, FDP_SDI.2
O.Check_Operation	FAU_GEN.1, FPT_TST.1, FPT_AMT.1
O.Control_Services	FDP_ACC.1/AUDIT, FDP_ACC.1/BACKUP, FDP_ACC.1/CRYPTO, FDP_ACC.1/LOAD, FDP_ACF.1/AUDIT, FDP_ACF.1/BACKUP, FDP_ACF.1/CRYPTO, FDP_ACF.1/LOAD,FMT_MSA.1/ ROLE_CRYPTO, FMT_MSA.1/ROLE_AUDIT, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1/AUDIT, FMT_MTD.1/ACCESS_CONTROL, FMT_SMF.1, FMT_SMR.1, FPT_TST.1, ACM_CAP.4, ADO_DEL.2, ADO_IGS.1, ALC_DVS.1, ALC_LCD.1
O.Detect_Attack	FPT_PHP.2, FPT_PHP.3
O.Error_Secure	FPT_AMT.1, FPT_FLS.1, FPT_RCV.1, FPT_TST.1
O. Crypto _Secure	FCS_COP.1/all iterations, FDP_IFC.1/CRYPTO, FDP_IFF.4/CRYPTO, AVA_CCA.1, AVA_VLA.4
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_SMF.1, FTP_TRP.1/TOE
O.Human_Interface	FTP_TRP.1/TOE
O.Secure_loading	FAU_GEN.1, FCS_CKM.2/backup_keys, FCS_COP.1/VERIF RSA, FCS_COP.1/DECRYPT TDES, FMT_MOF.1, FDP_ACC.1/LOAD, FDP_ACF.1/LOAD,
Security Objectives for the Environment	
O.ENV_Application	FIA_UID.1/CLIENT, FIA_UAU.1/CLIENT, FDP_ACC.1/CLIENT, FDP_ACF.1/CLIENT, FDP_UIT.1
O.ENV_Audit	FAU_SAR.1, FAU_STG.1/ENVIRONMENT,
O.ENV_Personnel	RE.ENV_Personnel
O.ENV_Protect_Access	RE.ENV_Protect_Access
O.ENV_Recovery	RE.ENV_Recovery

Objectives	Requirements
O.ENV_Secure_Init	RE.ENV_Secure_Init
O.ENV_Secure_Oper	RE.ENV_Secure_Oper
Security Assurance Requirements	
O.Keys_Secure	ADV_IMP.2, AVA_CCA.1, AVA_VLA.4
O.Protect_Exported_Data	ADV_IMP.2, AVA_CCA.1, AVA_VLA.4
O. Crypto _Secure	AVA_CCA.1, AVA_VLA.4

8.3.2 Security Requirements Sufficiency

8.3.2.1 TOE Security Requirements Sufficiency

O.Audit_CM (Audit record generation and export) addresses the generation and protection of audit data by the TOE. The audit generation is implemented by the SFR FAU_GEN.1 and FAU_GEN.2 with the audit events matching the list in O.Audit_CM. Additional audit is implemented by the SFR FAU_GEN.1 and FAU_GEN.2. The date and time of the event is recorded thanks to FPT_STM.1 ensuring reliable time stamp. The TOE stores the audit data according to the SFR FAU_STG.2/TOE until the audit trail is exported upon request of the Auditor or Crypto-officer under control of the SFR FDP_ACC.1/AUDIT, FDP_ACF.1/AUDIT and FMT_MTD.1/AUDIT. FMT_SMF.1 and FMT_MTD.1/AUDIT require management function for the audit. These management functions are provided to the Auditor only. The integrity of the audit data will be ensured by the SFR FAU_STG.2/TOE inside the TOE.

O.Keys_Secure (secure CSP-SCD and other cryptographic keys generation and management) addresses the confidentiality and integrity of the CSP-SCD and other cryptographic keys which shall be ensured during their whole life time. The SFR ensure the cryptographic secure CSP-SCD and other cryptographic keys generation by FCS_CKM.1 and FCS_RND.1 as well as operation by FCS_COP.1/all iterations according to the list of approved algorithms and parameters. The confidentiality and integrity of the CSP-SCD and other cryptographic keys will be protected by SFR FDP_RIP.1 and FDP_SDI.2 while internal processing. The SFR FCS_CKM.4 requires secure key destruction to prevent any misuse of CSP-SCD and other cryptographic keys after operational life time. The all CSP-SCD and other cryptographic keys management and operation is under access control of the SFR FDP_ACC.1/CRYPTO and FDP_ACF.1/CRYPTO. The TOE shall protect CSP-SCD and other cryptographic keys against side-channels by the SFR FDP_IFC.1/CRYPTO and FDP_IFF.4/CRYPTO. The SAR AVA_CCA.1 requires subject side-channels to the vulnerability analysis.
Note that the special protection of the CSP-SCD and other cryptographic keys is needed if the CSP-SCD and other cryptographic keys are 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 CSP-SCD (or any other cryptographic key) is exported. The complex protection of the CSP-SCD and other cryptographic keys as most valuable asset requires a systematic and complete vulnerability analysis considering high attack potential by SAR AVA_VLA.4.

O.Check_Operation (check for correct operation) addresses regular checks to verify that its components operate correctly. This security objective is implemented in the TOE by the

SFR for abstract machine testing FPT_AMT.1 and TSF testing FPT_TST.1. If these tests detect an error the TOE will transit into a secure state (see O.Error_secure) and prevent the normal operation. FAU_GEN.1 generates audit records about the test results of the SFR FPT-AMT.1 and FPT_TST.1 to inform the user (Auditor or Crypto-officer) about the performed self-tests and their results. The FPT_TST.1 includes checks of the executable code.

O.Control_Services (Management and control of TOE services) addresses the access control to TOE services and its management. The access control is implemented in the TOE by:

a) FDP_ACC.1/CRYPTO and FDP_ACF.1/CRYPTO for the cryptographic functions (Crypto-SFP),
b) FDP_ACC.1/AUDIT and FDP_ACF.1/AUDIT for the audit function (Audit-SFP),
c) FDP_ACC.1/LOAD and FDP_ACF.1/LOAD for the software update function (Load-SFP),
d) FDP_ACC.1/BACKUP and FDP_ACF.1/BACKUP for the backup function (Backup-SFP),
e) FDP_ACC.1/KEYS_DISTRIBUTION and FDP_ACF.1/ KEYS_DISTRIBUTION for the Keys enter function (Keys distribution-SFP)

with the roles Auditor, Crypto-officer and Crypto-user as defined by the SFR FMT_SMR.1. The SFR FMT_MSA.1/ROLE_CRYPTO, FMT_MSA.1/ROLE_AUDIT, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1/ACCESS_CONTROL, FMT_MTD.1/AUDIT and FMT_SMF.1 assign the management functions for the cryptographic to the Crypto-officer and audit functions to the Auditor. The SFR FMT_MSA.1/ROLE_CRYPTO extend the Crypto-officer's management functions to backup and restore. The SFR require the TSF to enforce the Audit-SFP, Backup-SFP, Keys distribution-SFP and Crypto-SFP to provide restrictive default values for security attributes which may be changed by the Auditor and the Crypto-officer. Note that the user management is addressed by O.User_authentication. The assurance requirements in the development environment, especial ACM_CAP.4 (Generation support and acceptance procedures), ADO_DEL.2 (Detection of modification), ADO_IGS.1 (Installation, generation, and start-up procedures), ALC_DVS.1 (Identification of security measures) and ALC_LCD.1 (Developer defined life-cycle model), prevent that malicious code is installed or hardware is manipulated during the development, production or delivery of the TOE.

O.Detect_Attack (detection of physical attacks) addresses the detection of physical tampering attempts and the secure destruction of the CSP-SCD and other cryptographic keys 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 RE.ENV_Protect_Access requires CSP security measures for physical protection of the TOE.

O.Error_secure (secure state in case of error) addresses a secure state and protection of CSP-SCD and other cryptographic keys confidentiality whenever the TOE detects an error. The SFR FPT_AMT.1 and FPT_TST.1 require 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 SCP-SCD 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 RE.ENV_Recovery describes the related security measures in the TOE environment.

O.Protect_Exported_Data (protection of data exported by the TOE) 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 requires the SFR FDP_BKP.1 the confidentiality and integrity protection of backup data. The backup and restore of CSP-SCD and other cryptographic keys, 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: (i) generate the backup keys by FCS_CKM.1/backup or re-import the backup keys by FCS_CKM.2/backup_keys, (ii) encryption of backup data by FCS_COP.1/BACKUP_ENC, (iii) data integrity protection by FCS_COP.1/BACKUP_INT. The SFR FDP_BKP.1 requires encrypting the CSP-SCD and other cryptographic keys 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 and FAU_GEN.2 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 and other cryptographic keys outside the TSC the TOE shall protect against side-channels to prevent any illicit information flow. The SFR FDP_IFC.1/BACKUP and FDP_IFF.4/BACKUP implements this protection and the SFR AVA_CCA.1 requires subject side-channels to the vulnerability analysis. FMT_MOF.1 ensures that the backup function be disable to fit some particular national laws.

O. Crypto_Secure (Secure advanced signature-creation) addresses the security of the signatures, i.e. the signature does not reveal the CSP-SCD and cannot be forged without knowledge of the CSP-SCD. The cryptographic security of signature is implemented by the SFR FCS_COP.1/all iterations with reference to the list of approved algorithms and parameters [5]. The SFR FDP_IFC.1/CRYPTO and FDP_IFF.4/CRYPTO requires TSF to prevent illicit information flow about the CSP-SCD through side-channels in the signatures. The SAR AVA_CCA.1 and AVA_VLA.4 requires covert-channel analysis and a systematic and complete vulnerability analysis considering high attack potential. That is because the signature-creation with CSP-SCD especially for certificates is the most important and critical service of the TOE.

O.User_authentication (authentication of users interacting with the TOE) 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 in the Crypto-user role 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.Human_Interface (reliable human interface) addresses the confidentiality and integrity of the data transferred between the TOE and the human user. This is performed through a trusted path ensured by SFR FTP_TRP.1/TOE for transmission of authentication and management data, the same as imported keys between the human user and the TOE.

O.Secure_loading (Secure loading of the TOE) 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 TDES and verified according to FCS_COP.1/VERIF RSA. Software update keys are previously distributed at initialisation phase according to FCS_CKM.2/backup_keys. 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.

8.3.2.2 TOE Environment Security Requirements Sufficiency

O.ENV_Application (Security in the Client Application) addresses the client application which acts as agent for the end-user gaining access to the TOE signing function provided and passes the DTBS representation to the TOE. The client application shall implement end-user identification and authentication required by the SFR FIA_UID.1/CLIENT and FIA_UAU.1/CLIENT. It shall implement access control for the DTBS representation sent to the TOE for signing according to the SFR FDP_ACC.1/CLIENT and FDP_ACF.1/CLIENT. Security controls in the TOE environment shall also prevent unauthorised manipulation of data submitted to the TOE as required by SFR FDP_UIT.1.

O.ENV_Audit (Audit review) addresses the review of the audit trail recorded by the TOE. The audit review of TOE's audit data is implemented in the IT environment by the SFR FAU_SAR.1. Because the TOE implements access control on reading the TOE's audit trail only the SFR FAU_STG.1/ENVIRONMENT ensures the availability of the TOE audit trail and prevents the modification of the TOE audit trail outside the TOE.

O.ENV_Personnel (Reliable Personnel) addresses the awareness of civil, financial and legal responsibilities, as well as the obligations the CSP personnel have to face, depending on their role. The RE.ENV_Personnel implements the definition of the obligations, the services and the roles of the TOE users. The CSP shall inform about their civil, financial and legal responsibilities and train the personnel for their roles.

O.ENV_Protect_Access (Prevention of Unauthorised Physical Access) addresses the physical and logical protection of the TOE, the restriction the TOE usage and the limitation of the access to TOE assets to authorised persons only. The RE.ENV_Protect_Access requests the CSP to establish physical and organisational security measures against modification of TOE hardware, firmware and software. These measures shall restrict the access to the TOE and protected assets to authorised persons. Note that the TOE itself protects by FPT_PHP.2 and FPT_PHP.3 the confidentiality of the CSP-SCD and other cryptographic keys against physical access because even the CSP personnel do not need to know the CSP-SCD and other cryptographic keys in plaintext.

O.ENV_Recovery (Secure Recovery in Case of Major Failure) addresses the recovery plans and procedures for a secure and timely recovery in the case of a major problem with the TOE. The RE.ENV_Recovery implements such recovery plans and procedures using the TOE TSF according to FDP_BKP.1 and other SFR. It takes recovery in case of detected errors or physical tampering into account.

O.ENV_Secure_Init (Secure Initialisation Procedures) addresses secure set-up and initialisation the TOE for the CSP services. The RE.ENV_Secure_Init implements the definition and application of procedures and controls set-up the TOE for the secure generation of CSP-SCD and other cryptographic keys and initialisation of the signature function.

O.ENV_Secure_Oper (Secure Operating Procedures) addresses the procedures and controls in the TOE environment to operate the TOE within a CA system in compliance with the requirements of the EU directive and the Policy for certification authorities issuing qualified certificates. The RE.ENV_Secure_Oper requires the implementation of such procedures and controls and the observance of the TOE guidance.

8.4 TOE Summary Specification Rationale

8.4.1 TOE Security functions Coverage

Table 8-4 TOE security functions to Security Requirements Mapping

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.2/backup_keys	SF.SI
FCS_CKM.2/other_keys	SF.keys_distribution
FCS_CKM.4	SF.CO.key_destruction
FCS_COP.1/SIGN	SF.CO.cryptographic_functions
FCS_COP.1/VERIF	SF.CO.cryptographic_functions, SF.SL
FCS_COP.1/ENCRYPT	SF.CO.cryptographic_functions
FCS_COP.1/DECRYPT	SF.CO.cryptographic_functions, SF.SL
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/KEYS_DIST RIBUTION	SF.Access_Control
FDP_ACC.1/LOAD	SF.Access_Control
FDP_ACF.1/CRYPTO	SF.authentication.roles
FDP_ACF.1/AUDIT	SF.authentication.roles
FDP_ACF.1/BACKUP	SF.authentication.roles
FDP_ACF.1/KEYS_DISTR IBUTION	SF.authentication.roles
FDP_ACF.1/LOAD	SF.authentication.roles
FDP_BKP.1	SF.backup.command, SF.backup.data_protection
FDP_ETC.1	SF.backup.data_protection
FDP_IFC.1/BACKUP	SF.Access_Control

Security requirements	TOE security functions
FDP_IFC.1/CRYPTO	SF.Access_Control
FDP_IFF.4/BACKUP	SF.SM.hardware
FDP_IFF.4/CRYPTO	SF.SM.hardware
FDP_RIP.1	SF.SI
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
FPT_AMT.1	SF.SM.tests
FPT_FLS.1	SF.SM.alarms
FPT_ITC.1	SF.backup.data_protection
FPT_ITI.1	SF.backup.data_protection
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

8.4.2 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 embedded EEPROM 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 consistancy 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 keys are distributed securely. SF.keys_distribution ensures key distribution through the trusted path.

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 HMAC SHA, HMAC_SHA2, RSA, SHA-RSA, SHA2_RSA and ECDSA with SHA-2 algorithms. SF.SL uses FCS_COP.1/VERIF RSA to check executable code integrity during software update.

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 TDES, RSA, AES and ECC algorithms. SF.SL uses FCS_COP.1/DECRYPT TDES 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 standards 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 and **FDP_ACC.1/KEYS_DISTRIBUTION (Subset access control)** defines the scope of control of the policies that form the identified access control portion of the SFP. This component is met with SF.Access_Control.

FDP_ACF.1/CRYPTO, FDP_ACF.1/AUDIT, FDP_ACF.1/BACKUP and FDP_ACF.1/ KEYS_DISTRIBUTION (**Security attribute based access control)** describes the rules of the access control policy to the backup function. SF.authentication.roles defines the roles to access cryptographic functions, backup functions and audit file.

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 attibutes) defines function to backup data without security attributes. SF.backup.data_protection defines the backup policy to meet this component.

FDP_IFC.1/BACKUP and FDP_IFC.1/CRYPTO (**Subset Information Flow Control)** identifies the entities involved in the information flow control SFP. SF.Access_Control identifies those entities.

FDP_IFF.4/BACKUP and FDP_IFF.4/CRYPTO (Partial elimination of illicit information flows) identifies rules to eliminate partial illicit information flows. This components is covered with SF.SM.hardware.

FDP_RIP.1 (**Subset residual information protection)** ensures that keys and authentication data are no longer available after TOE de allocation data. SF.SI ensures that TOE desinstallation cleared 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 periodical tests to control stored keys and embedded code integrity.

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 3 failures the 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 Behavior)** ensures that the TSF restricts the ability to modify the behavior 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, crypto officer and 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.

FPT_AMT.1 (Abstract Machine Testing) guaranties the correct operation of the underlying abstract machine upon which the TSF relies. SF.SM.tests describes the TOE test policy and ensures correct functioning of all security components.

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-TSFconfidentiality 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_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 installed again by the crypto officer 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_ITC.1 (inter-TSF trusted channel) ensures that software update (SF.SL) is performed in a secure way. Data (executable code) are encrypted (protection from modification) and signed (BULL code identification and protection from disclosure). The TOE performs data integrity verification and authentication and subsequently decrypt the data to update the TOE software.

FPT_TRP.1/TOE (Trusted Path) ensures that authentication process is performed through a secure path logically and physically independant from user data path. SF.authentication.Trusted_Path guaranties that any authentication takes place via a Safepad (Smart Card reader) which is linked by a serial connection (trusted path) to the TOE.

8.5 Dependency Rationale

8.5.1 Functional and Assurance Requirements Dependencies

Table 8.5 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 8.5.2)
FAU_GEN.2	FAU_GEN.1, FIA_UID.1	
FAU_STG.2/TOE	FAU_GEN.1, FAU_GEN.1	
FCS_CKM.1	FCS_COP.1/all iterations, FCS_CKM.4, FMT_MSA.2 FCS_CKM.2/backup_keys, FCS_CKM.2/other_keys	
FCS_CKM.2/backup_keys	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	Back-up key material is provided by the TOE (FCS_CKM.1/backup)
FCS_CKM.2/other_keys	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	The dependency of FCS_CKM.1 is not satisfied (keys are entered through the trusted communication path)
FCS_CKM.4	FCS_CKM.1, FMT_MSA.2	
FCS_COP.1/ BACKUP_ENC	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	Back-up key material is provided by the TOE (FCS_CKM.1/backup)
FCS_COP.1/ BACKUP_INT	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	Back-up key material is provided by the TOE (FCS_CKM.1/backup)
FCS_COP.1/all iterations	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	
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/KEYS_DISTRIBUTION	FDP_ACF.1/ KEYS_DISTRIBUTION	
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/ KEYS_DISTRIBUTION	FDP_ACC.1/ KEYS_DISTRIBUTION, FMT_MSA.3	
FDP_BKP.1	FCS_CKM.2/backup_keys, FCS_COP.1/BACKUP_ENC, FCS_COP.1/BACKUP_INT	

Requirement	CC-required Dependencies	Remark
FDP_ETC.1	FDP_ACC.1/CRYPTO, FDP_ACC.1/BACKUP, FDP_ACC.1/KEYS_DISTRIBUTION, FDP_ACC.1/AUDIT, FDP_IFC.1/CRYPTO, FDP_IFC.1/ BACKUP	
FDP_IFC.1/BACKUP	FDP_IFF.1	dependency is not satisfied by the CWA 14167-2 PP (see justification in section 8.5.2)
FDP_IFC.1/CRYPTO	FDP_IFF.1	dependency is not satisfied by the CWA 14167-2 PP (see justification in section 8.5.2)
FDP_IFF.4/BACKUP	AVA_CCA.1, FDP_IFC.1/BACKUP	
FDP_IFF.4/CRYPTO	AVA_CCA.1, FDP_IFC.1/CRYPTO	
FIA_AFL.1	FIA_UAU.1	
FIA_UAU.1	FIA_UID.1	
FMT_MOF.1	FMT_SMR.1	
FMT_MSA.1/ ROLE_CRYPTO	FDP_ACC.1/BACKUP, FDP_ACC.1/CRYPTO, FDP_ACC.1/LOAD, FDP_ACC.1/KEYS_DISTRIBUTION, FMT_SMF.1, FMT_SMR.1	
FMT_MSA.1/ ROLE_AUDIT	FDP_ACC.1/AUDIT, FMT_SMF.1, FMT_SMR.1	
FMT_MSA.2	ADV_SPM.1, FDP_ACC.1/AUDIT, FDP_ACC.1/BACKUP, FDP_ACC.1/CRYPTO, FDP_ACC.1/LOAD, FDP_ACC.1/KEYS_DISTRIBUTION, 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_SMF.1, FMT_SMR.1	
FMT_MTD.1/ ACCESS_CONTROL	FMT_SMF.1, FMT_SMR.1	
FMT_MTD.1/ USER_CRYPTO	FMT_SMF.1, FMT_SMR.1	
FMT_MTD.1/ USER_AUDIT	FMT_SMF.1, FMT_SMR.1	
FMT_MTD.1/RAD	FMT_SMF.1, FMT_SMR.1	
FMT_SMF.1	(no dependencies)	
FMT_SMR.1	FIA_UID.1	
FPT_FLS.1	ADV_SPM.1	
FPT_ITI.1	(no dependencies)	

Requirement	CC-required Dependencies	Remark
FPT_PHP.2	FMT_MOF.1	dependency is not satisfied by the CWA 14167-2 PP (see justification in section 8.5.2)
FPT_RCV.1	FPT_TST.1, AGD_ADM.1, ADV_SPM.1	
FPT_STM.1	(no dependencies)	
FPT_TST.1	FPT_AMT.1	
FTP_TRP.1/TOE	(no dependencies)	
Assurance Requirements		
ACM_AUT.1	ACM_CAP.3	ACM_CAP.4 is hierarchical to ACM_CAP.3.
ACM_CAP.4	ALC_DVS.1	
ACM_SCP.2	ACM_CAP.3	ACM_CAP.4 is hierarchical to ACM_CAP.3
ADO_DEL.2	ACM_CAP.3	ACM_CAP.4 is hierarchical to ACM_CAP.3
ADO_IGS.1	AGD_ADM.1	
ADV_FSP.2	ADV_RCR.1	
ADV_HLD.2	ADV_FSP.1, ADV_RCR.1	ADV_FSP.2 is hierarchical to ADV_FSP.1
ADV_IMP.2	ADV_LLD.1, ADV_RCR.1, ALC_TAT.1	
ADV_LLD.1	ADV_HLD.2, ADV_RCR.1	
ADV_SPM.1	ADV_FSP.1	ADV_FSP.2 is hierarchical to ADV_FSP.1
AGD_ADM.1	ADV_FSP.1	ADV_FSP.2 is hierarchical to ADV_FSP.1
AGD_USR.1	ADV_FSP.1	ADV_FSP.2 is hierarchical to ADV_FSP.1
ALC_FLR.3	(no dependencies)	
ALC_TAT.1	ADV_IMP.1	ADV_IMP.2 is included and hierarchical to ADV_IMP.1
ATE_COV.2	ADV_FSP.1, ATE_FUN.1	ADV_FSP.2 is hierarchical to ADV_FSP.1
ATE_DPT.1	ADV_HLD.1, ATE_FUN.1	ADV_HLD.2 is hierarchical to ADV_HLD.1
ATE_IND.2	ADV_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1	ADV_FSP.2 is hierarchical to ADV_FSP.1
AVA_CCA.1	ADV_FSP.2, ADV_IMP.2, AGD_ADM.1, AGD_USR.1	
AVA_MSU.2	ADO_IGS.1, ADV_FSP.1, AGD_ADM.1, AGD_USR.1	ADV_FSP.2 is hierarchical to ADV_FSP.1
AVA_SOF.1	ADV_FSP.1, ADV_HLD.1	ADV_FSP.2 is hierarchical to ADV_FSP.1, ADV_HLD.2 is hierarchical to ADV_HLD.1
AVA_VLA.4	ADV_FSP.1, ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, AGD_ADM.1, AGD_USR.1	ADV_FSP.2 is hierarchical to ADV_FSP.1, ADV_IMP.2 is included and hierarchical to ADV_IMP.1

Requirement	CC-required Dependencies	Remark
Functional Requirements for the IT environment		
FAU_SAR.1	FAU_GEN.1	The IT environment provides the audit trail generated by TOE (as required by FAU.GEN.1) to the System auditor of the CSP.
FAU_STG.1/ ENVIRONMENT	FAU_GEN.1	The IT environment protects the audit trail generated by TOE (as required by FAU.GEN.1).
FDP_ACC.1/CLIENT	FDP_AFC.1/CLIENT	
FDP_AFC.1/CLIENT	FDP_ACC.1/CLIENT	FMT_MSA.3 is not fulfilled, a rationale is given in section 8.5.2
FDP_UIT.1	FDP_ACC.1/CLIENT, FTP_TRP.1/CLIENT	
FIA_UAU.1/CLIENT	FIA_UID.1/CLIENT	
FIA_UID.1/CLIENT	(no dependencies)	

8.5.2 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.
FDP_IFC.1/BACKUP	FDP_IFF.1	FDP_IFC.1/Backup is defined for the CSP-SCD and other cryptographic keys without reference to any security attribute.
FDP_IFC.1/CRYPTO	FDP_IFF.1	FDP_IFC.1/CRYPTO is defined for the CSP-SCD and other cryptographic keys without reference to any security attribute.
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.

Component	Justification for not including	Remark
Functional Requirements for the IT environment		
FDP_AFC.1/CLIENT	FMT_MSA.3	The cryptographic module does not need of specific requirements for management of security attributes of the client application. It is up to the CSP to define which kind of static attribute initialisation of the client application (either permissive or restrictive in nature) ensures that the default values of security attributes are appropriate.

8.6 Security Requirements Grounding in Objectives

Table 8-6 Requirements to Objectives Mapping

Requirement	Security Objectives
Security Requirements for the TOE	
ACM_AUT.1	EAL4
ACM_CAP.4	EAL4
ACM_SCP.2	EAL4
ADO_DEL.2	EAL4
ADO_IGS.1	EAL4
ADV_FSP.2	EAL4
ADV_HLD.2	EAL4
ADV_IMP.2	O.Keys_Secure, ADV_IMP.2 is hierarchical to ADV_IMP.1 required for EAL4
ADV_LLD.1	EAL4
ADV_RCR.1	EAL4
ADV_SPM.1	EAL4
AGD_ADM.1	EAL4
AGD_USR.1	EAL4
ALC_DVS.1	EAL4
ALC_FLR.3	EAL4
ALC_LCD.1	EAL4
ALC_TAT.1	EAL4
ATE_COV.2	EAL4
ATE_DPT.1	EAL4
ATE_FUN.1	EAL4
ATE_IND.2	EAL4
AVA_CCA.1	O. Crypto_Secure, O.Protect_Exported_Data, O.Keys_Secure O.Secure_Loading
AVA_MSU.2	EAL4
AVA_SOF.1	EAL4
AVA_VLA.4	O.Keys_Secure, O.Protect_Exported_Data, O. Crypto_Secure, O.secure_loading

Requirement	Security Objectives
FAU_GEN.1	O.Audit_CM, O.Protect_Exported_Data, O.Check_Operation O.Secure_Loading
FAU_GEN.2	O.Audit_CM, O.Protect_Exported_Data
FAU_STG.2/TOE	O.Audit_CM
FCS_CKM.1	O.Keys_Secure, O.Protect_Exported_Data
FCS_CKM.2/backup_keys	O.Protect_Exported_Data O.Secure_Loading
FCS_CKM.2/other_keys	O.Human_Interface
FCS_CKM.4	O.Keys_Secure
FCS_COP.1/ BACKUP_ENC	O.Protect_Exported_Data
FCS_COP.1/ BACKUP_INT	O.Protect_Exported_Data
FCS_COP.1/all iterations	O. Crypto_Secure, O.Keys_Secure O.Secure_Loading
FCS_RND.1	O.Keys_Secure
FDP_ACC.1/BACKUP	O.Protect_Exported_Data, O.Control_Services
FDP_ACC.1/AUDIT	O.Audit_CM, O.Control_Services
FDP_ACC.1/CRYPTO	O.Keys_Secure, O.Control_Services
FDP_ACC.1/LOAD	O.secure_loading, O.Control_Services
FDP_ACC.1/KEYS_DISTRIBUTION	O.Control_Services
FDP_ACF.1/BACKUP	O.Protect_Exported_Data, O.Control_Services
FDP_ACF.1/AUDIT	O.Control_Services, O.Audit_CM
FDP_ACF.1/CRYPTO	O.Keys_Secure, O.Control_Services
FDP_ACF.1/LOAD	O.secure_loading , O.Control_Services
FDP_ACF.1/ KEYS_DISTRIBUTION	O.Control_Services
FDP_BKP.1	O.Protect_Exported_Data
FDP_ETC.1	O.Protect_Exported_Data
FDP_IFC.1/BACKUP	O.Protect_Exported_Data, O.Keys_Secure
FDP_IFC.1/CRYPTO	O.Keys_Secure, O. Crypto_Secure
FDP_IFF.4/BACKUP	O.Protect_Exported_Data, O.Keys_Secure
FDP_IFF.4/CRYPTO	O.Keys_Secure, O. Crypto_Secure
FDP_RIP.1	O.Keys_Secure
FDP_SDI.2	O.Keys_Secure
FIA_AFL.1	O.User_Authentication
FIA_ATD.1	O.User_Authentication
FIA_SOS.1	O.User_Authentication
FIA_UAU.1	O.User_Authentication
FIA_UID.1	O.User_Authentication
FMT_MOF.1	O.Control_Services O.Secure_Loading O.Protect_Exported_Data
FMT_MTD.1/ USER_CRYPTO	O.User_Authentication
FMT_MTD.1/ USER_AUDIT	O.User_Authentication
FMT_MTD.1/RAD	O.User_Authentication
FMT_MSA.1/ ROLE_AUDIT	O.Control_Services

Requirement	Security Objectives
FMT_MSA.1/ ROLE_CRYPTO	O.Control_Services
FMT_MSA.2	O.Control_Services
FMT_MSA.3	O.Protect_Exported_Data, O.Control_Services
FMT_MTD.1/AUDIT	O.Audit_CM
FMT_MTD.1/ ACCESS_CONTROL	O.Control_Services
FMT_SMF.1	O.Audit_CM, O.Control_Services, O.User_Authentication
FMT_SMR.1	O.Control_Services
FPT_AMT.1	O.Check_Operation, O.Error_Secure
FPT_ITC.1	O.Protect_Exported_Data
FPT_ITI.1	O.Protect_Exported_Data
FPT_FLS.1	O.Error_Secure
FPT_PHP.2	O.Detect_Attack
FPT_PHP.3	O.Detect_Attack
FPT_RCV.1	O.Error_Secure
FPT_STM.1	O.Audit_CM
FPT_TST.1	O.Error_Secure, O.Check_Operation
FTP_TRP.1/TOE	O.User_Authentication, O.Human_Interface
Security Objectives for the Environment	
FAU_SAR.1	O.ENV_Audit
FAU_STG.1/ ENVIRONMENT	O.ENV_Audit
FDP_ACC.1/ CLIENT	O.ENV_Application
FDP_ACF.1/ CLIENT	O.ENV_Application
FDP_UIT.1	O.ENV_Application
FIA_UAU.1	O.ENV_Application
FIA_UID.1	O.ENV_Application

8.7 Rationale for Extensions

8.7.1 Rationale for Extension of Class FCS with Family FCS_RND

The TOE shall generate CSP-SCD and other cryptographic keys with high cryptographic quality using random number generators. The family FCS_RNG.1 requires the ST editor to define the quality metric of the random numbers used by the TOE to generate the CSP-SCD and other cryptographic keys. The component similar to FCS_RND.1 in CC part 2 is limited in their application to secrets used as authentication information.

FCS_RND generation of random numbers

Family behaviour
This family defines quality metrics for generating random numbers intended for cryptographic purposes.

Component levelling
FCS_RND.1 The generation of random numbers using TSFs requires the random numbers to meet the defined quality metrics.

Management: FCS_RND.1
No management functions are provided for.

Audit: FCS_RND.1
There are no events identified that should be auditable if FCS_RND generation of random numbers data generation is included in the ST.

FCS_RND.1 Quality metrics for random numbers
Hierarchical to: no other components.

FCS_RND.1.1 The TSFs shall provide a mechanism for generating random numbers that meet [assignment: *a defined quality metric*].

FCS_RND.1.2 The TSFs shall be able to enforce the use of TSF-generated random numbers for [assignment: *list of TSF functions*].

Dependencies: FPT_TST.1 TSF testing.

8.7.2 Rationale for Extension of Class FDP with Family FDP_BKP

The HSM supports backup of CSP-SCD ***and other cryptographic keys, oth***er 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 CSP-SCD, other cryptographic keys and TSF data for backup and recovery.

Backup and recovery (FDP_BKP)

Family behaviour

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

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

Management: FDP_BKP.1

There are no management activities foreseen.
Audit: FDP_BKP.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:
a) Use of the backup function,
b) Use of the recovery function,
c) Unsuccessful recovery because of detection of modification of the backup data.

FDP_BKP.1 Backup and recovery
Hierarchical to: No other components.

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 CSP-SCD and other cryptographic keys 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.

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

8.8 Rationale for Assurance Level 4 Augmented

The assurance level for this security target is EAL4 augmented.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:

ADV_IMP.2 Development - Implementation of the TSF
ALC_FLR.3 **Systematic flaw remediation**
AVA_CCA.1 Vulnerability Assessment - Covert channel analysis
AVA_VLA.4 Vulnerability Assessment - Vulnerability Analysis – Highly resistant

The security objective O.Keys_Secure includes protection against disclosing completely or partly the CSP-SCD and other cryptographic keys through any physical or logical TOE interface. This calls for security functional requirements as FDP_IFF.4/Crypto and security assurance requirements as AVA_CCA.1. ADV_IMP.2 is required to fulfil the dependencies for AVA_CCA.1.

The TOE generates, uses and manages the most sensitive data of the CSP – the CSP-SCD. Any loss of confidentiality or integrity of the CSP-SCD threaten the security of the certificates signed with this CSP-SCD and therefore the security of all signatures created with the SCD which correspond to the certificates. The cryptographic security of the CSP-SCD/CSP-SVD pair generation and the signing with the CSP-SCD can be ensured only by the TOE itself. The TOE shall be free of any covert channel which might compromise the CSP-SCD. The TOE environment shall support the TOE in CSP-SCD protection against physical and some other attacks but cannot make up for TOE security. The protection of the CSP-SCD shall be solely and in tabloid form provided by the CM as part of the trustworthy system. The complex protection of the CSP-SCD requires a systematic and complete vulnerability analysis by SAR AVA_VLA.4. The TOE protecting the CSP-SCD as most valuable asset shall be shown to be highly resistant to penetration attacks. Therefore the strength of function “high” for AVA_SOF.1 and AVA_VLA.4 is chosen.

9 References

[1] DIRECTIVE 1999/93/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 13 December 1999 on a Community framework for electronic signatures

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

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

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

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

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

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

[8] Common Methodology for Information Technology Security Evaluation CEM-99/045 Part 2: Evaluation Methodology, Version 1.0, August 1999

[9] FIPS 46-3 Data Encryption Standard (DES) October 25, 1999

[10] FIPS PUB 140-2 Security requirements for cryptographic modules May 25, 2001.

[11] FIPS 180-1 Secure hash standard April 17,1995

[12] FIPS PUB 186-2 Digital Signature Standard January 27,2000

[13] RFC 1321 The MD5 Message-Digest algorithm April 1992

[14] RFC 2104 HMAC: Keyed-Hashing for message Authentication February 1997

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

[16] PKCS#1 RSA Cryptography Standard V2.1 June 14,2002.

[17] PKCS#8 Private-Key information syntax standard V1.2 November 1, 1993.

[18] PKCS#11 Cryptographic Token interface standard V2.11 November 2001

[19] B. SCHNEIER Applied Cryptography Second Edition John Woley & Sons, 1996

[20] The DieHard Package is obtained from Dr. George Marsaglia at
http://www.stat.fsu.edu/~geo/diehard.html.

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