Keyper Hardware Security Module (HSM) v2.0

Security Target

Common Criteria: EAL4

Version 1.3

13-DEC-2012

Document information

Document identification

Document ID	KPR_EAL4_ASE
Document title	Keyper HSM EAL4 Security Target
Release authority	David Miller
Product version	Keyper Hardware Security Module Version 2.0

Document history

Version	Date	Description
0.1	20-APR-11	Initial draft for internal review.
0.2	11-MAY-11	Addressed initial comments.
0.3	20-MAY-11	Updated TOE title from Series K to Keyper.
0.4	01-JUN-11	Update to address additional client comments.
0.5	15-JUN-11	Update to address comments on cryptographic algorithm.
0.6	13-Apr-12	Updated to address comments in EOR001
0.7	24-Aug-12	Updated to address comments by AL
0.8	02-Oct-12	Updated to address a number of identified issues
1.0	23-Oct-12	Final release
1.1	26-Oct-12	Correction to the claimed security assurance requirements
1.2	01-Nov-12	Corrected inaccuracies in the rationale for T.Misuse_Sign
1.3	13-Dec-12	Corrected reference to TOE version numbering.

Table of Contents

1 Security Target Introduction (ASE_INT)

1.1 Background

Where cryptographic services are used to protect an information system, trust and integrity are derived from the security of the underlying signing and encryption keys. This makes protection of these keys critical to the overall trust and integrity of a system.

Cryptographic key material can be stored and protected in a variety of ways and on a variety of media including software, smart cards and USB tokens. However, where protection is critical and potentially of national security importance, the use of these types of key storage devices is considered inadequate.

The advanced method for storing and securing key material is the use of a dedicated and physically separate device known as a Hardware Security Module (HSM). AEP Networks has designed the Keyper range of HSMs which provides the following core security functions that protect key material:

a) Secure key management

b) Secure storage

c) User authentication

d) Security management

e) Access control for key management functions

f) Auditing of security relevant events

g) Tamper protection

h) Self-testing

The AEP Keyper is ideally suited to businesses deploying a cryptographic system where the protection of cryptographic keys is a priority, for example, in organizations requiring certificate signing, code or document signing, bulk generation or ciphering of keys or data. Keyper is available in four models offering various levels of scale: Professional, Enterprise, Professional High Availability, and Enterprise High Availability. However, only two models are within the scope of the evaluation: Enterprise and Professional.

1.2 ST reference

ST Title	Security Target for the Keyper Hardware Security Module (HSM) v2.0
ST Version/Date	1.3 (13-DEC-2012)
TOE Reference	Keyper Hardware Security Module (HSM) v2.0: a) Enterprise (Hardware: 9720, Software: 011126) b) Professional (Hardware: 9720, Software: 010405)
CC Identification	Common Criteria for Information Technology (IT) Security Evaluation, Version 3.1 (REV 3) July 2009, incorporating: • Part One – Introduction and General Model (Ref. [1]), • Part Two – Security Functional Components (Ref. [2]), and • Part Three – Security Assurance Components (Ref. [3]).

1.3 Document organization

This document is organized into the following major sections:

a) Section 1 provides the introductory material for the ST as well as the Target of Evaluation (TOE) overview (ASE_INT).

b) Section 2 provides the conformance claims for the evaluation (ASE_CCL).

c) Section 3 provides the definition of the security problem addressed by the TOE (ASE_SPD).

d) Section 4 defines the security objectives for the TOE and the environment (ASE_OBJ).

e) Section 5 contains the security functional requirements derived from the Common Criteria, Part 2 (ASE_REQ).

f) Section 6 contains the security assurance requirements derived from the Common Criteria, Part 3 (ASE_REQ).

g) Section 7 provides the TOE summary specification that demonstrates how the TOE implements the claimed security functions.

h) Annex A provides the extended components definition (ASE_ECD).

1.4 References

[1] Common Criteria for Information Technology Security Evaluation: Part 1: Introduction and general model, July 2009, Version 3.1 Revision 3, CCMB-2009-07-001

[2] Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, July 2009, Version 3.1, Revision 3 Final, CCMB-2009-07-002.

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, July 2009, Version 3.1, Revision 3 Final, CCMB-2009-07-003.

[4] Directive 1999/93/EC of the European parliament and of the council of 13 December 1999 on a Community framework for electronic signatures, Official Journal of the European Communities, Jan 2000.

[5] Cryptographic Module for CSP Signing Operations with Backup - Protection Profile (CMCSOB-PP), May 2003, version 0.28; CWA 14167-2:2004.

[6] Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 1: System Security Requirements, June 2003, CWA 14167-1.

[7] Cryptographic Module for CSP Key Generation Services – Protection Profile (CMCKG-PP), February 2004, version 0.12, CWA 14167-3:2004

[8] Cryptographic Module for CSP Signing Operations - Protection Profile (CMCSO-PP), October 2004, version 0.28; CWA 14167-4:2004.

[9] Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures, version 1.1.1, March 2003, ETSI SR 002 176.

[10] RFC 3447 PKCS #1: RSA Cryptography Specifications, version 2.1, February 2003.

[11] Federal Information Processing Standard Publication (FIPS-PUB) 46-3, Data Encryption Standard, 25 October 1999.

[12] Federal Information Processing Standard Publication (FIPS-PUB) 197, Advanced Encryption Standard (AES), 26 November 2001.

[13] Federal Information Processing Standard Publication (FIPS-PUB) 186-2, Digital Signature Standard (DSS), January 2000.

[14] Federal Information Processing Standard Publication (FIPS-PUB) 180-1, Secure Hash Standard, August 1 2002.

[15] NIST Special Publication 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised), March 2007.

[16] Federal Information Processing Standard Publication (FIPS-PUB) 140-2, Security Requirements for Cryptographic Modules, May 2001.

1.5 Defined terms (ASE_REQ)

The following table defines all subjects, objects, operations, security attributes, external entities and other key terms that are used within the statements of security functional and assurance requirements.

Term/Acronym	Definition
Adapter Authorisation Key (AAK)	The AAK protects the TOE from unauthorised access by providing the means to authenticate Security Officer smart cards. There is only one AAK per device which is generated in the TOE during the initialisation phase.
Administrator	A CSP user role that performs TOE initialisation or other TOE administrative functions. These tasks are mapped to the Security Officer.
Application Keys	Application Keys are generated by a request for cryptographic services from an external application. An Application Key is protected along with key policy and identifiers and wrapped by the SMK for protection during backup or restore.
Auditor	A user exporting the TOE audit data and reviewing the audit data with tools in the TOE environment.
Authentication data	Information used to verify the claimed identity of a user.
Backup	Backup 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.
CEN workshop agreement (CWA)	A consensus-based specification, drawn up in an open workshop environment of the European Committee for Standardization (CEN).
Certificate	An electronic attestation which links the SVD to a person and confirms the identity of that person.
Certification-service-provider (CSP)	An entity or a legal or natural person who issues certificates or provides other services related to electronic signatures.
CSP signature creation data (CSP-SCD)	SCD which is used by the CSP for the creation of advanced electronic signatures in qualified certificates or for signing certificate status information.

Term/Acronym	Definition
CSP signature verification data (CSP-SVD)	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.
Data to be signed (DTBS)	The complete electronic data to be signed, such as QC content data or certificate status information.
Digital signature	Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery 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 (Ref. [4]).
Hardware security module (HSM)	The cryptographic module used to generate the advanced signature in qualified certificates and which represents the TOE.
Internal Storage Master Key (ISMK)	The purpose of the ISMK is to protect the internal store.
Reference authentication data (RAD)	Data persistently stored by the TOE for verification of the authentication attempt as authorised user.
Restore	Action of importing of the backup data to recreate the state of the TOE at the time the backup was created.
Side-channel	Means illicit information flow in result of the physical behavior 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 behavior from outside.
Signature-creation data (SCD)	Unique data, such as codes or private cryptographic keys, which are used by the signatory to create an electronic signature.
Signature-verification data (SVD)	Data, such as codes or public cryptographic keys, which are used for the purpose of verifying an electronic signature.
Storage Master Key (SMK)	The purpose of the SMK is to protect the smart card store.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
User data	Data created by and for the user that does not affect the operation of the TSF.

1.6 ST overview

1.6.1 TOE overview

The TOE is a dedicated hardware product which provides secure digital signature services, cryptographic services and key management services to applications that reside on physically separate host computer systems. The TOE encompasses two models: AEP Keyper Enterprise 9720 and AEP Keyper Professional 9720. Both models share the same features and architecture (the only difference is performance), therefore both models shall be considered together. Two additional "High Availability" models also exist, however they are out of the scope of this evaluation.

The TOE communicates with applications on host computer systems through a standard network interface (100BASE-T Fast Ethernet). Host applications communicate with the TOE via an AEP produced 'Provider', which performs a similar function to a software driver. There are four 'Providers' (PKCS11 Provider, RSA Full Provider (MSCAPI), SChannel Provider (MSCAPI) and OpenSSL Provider) which may be selected based on host application requirements (note that the provider is not included in the scope of this evaluation).

The TOE is a secure module that is contained within an outer casing. The outer casing includes a Keypad, LCD screen, smart card reader and a number of external ports. The TOE is tamper reactive; and has been validated against the requirements for the FIPS PUB 140-2 at level 4. The device is intended for use in a dedicated network with devices and applications that make use of its cryptographic functions. The device should be provided appropriate physical and logical protections.

In addition to the Ethernet interface, the TOE provides interfaces for import/export of cryptographic keys via smartcards. Key management operations take place via a keypad/LCD and smart card reader. The TOE also implements a cryptographic policy that restricts specific roles to specific tasks associated with managing keys and the device.

1.6.2 TOE usage and major security features

The TOE is intended to be deployed as a core component of a critical enterprise cryptographic system where the generation, protection and management of cryptographic keys are all priorities. It can be deployed as part of any cryptographic system that uses digital keys. The TOE is intended to provide a high-level of assurance in protection of the digital keys, therefore the keys must be of high-value, that is, that there would be a significant negative impact if the keys were compromised.

Typically the TOE is used by certificate authorities (CAs) and registration authorities (RAs) in a PKI environment to generate, store, and manage keys. The banking sector, card payment systems and government agencies also make use of the TOE to provide the necessary protection for digital keys.

In the context of this ST the TOE is expected to provide the following major security features:

a) Secure generation, distribution and destruction of cryptographic keys.

b) Secure storage and management of keys throughout their lifecycle.

c) User authentication to facilitate controlled access to cryptographic key management and TOE management functions by trusted personnel only.

d) Security management to enable role-based management of the core functions of the TOE.

e) Access control for key management functions to ensure that only specified roles are permitted to perform defined tasks.

f) Auditing of security relevant events to provide suitable accountability.

g) Self-test of the core cryptographic functions and algorithms of the TOE.

h) Tamper protection to ensure that the TOE is adequately protected from unauthorised physical access.

1.6.3 TOE type

The TOE is a hardware security module (HSM)—a hardware-based and dedicated security device which generates, stores, protects and manages digital cryptographic keys. The TOE can be categorised as a ***key management system*** in accordance with the categories identified on the Common Criteria Portal that lists all certified products.

1.6.4 Supporting hardware, software and/or firmware

The TOE requires an AEP-provided 'Provider' (also called a 'driver') on the host computer to communicate cryptographic operations over the network. Firmware updates and audit extraction operations are also instigated by the host computer via AEP-provided HSM Management Centre (HMC) software. This supporting software is provided with the HSM.

1.7 TOE description

1.7.1 Physical scope of the TOE

The TOE is a hardware module with a tamper resistant casing. The TOE is held within a small standalone desktop unit (depicted in Figure 2, below). The physical TOE boundary is the tamper reactive mesh casing, which surrounds the secure module (Figure 1). The tamper-mesh encased TOE is physically segregated and lies within the outer desktop unit chassis. A single ribbon cable, which bundles all of the TOE's external interfaces, connects the secure module (TOE) to the outer desktop unit.

Figure 1: AEP Keyper Secure Module

The front panel of the desktop unit consists of a keypad that can be folded away into the main casing of the device, an LCD panel, LED indicators for system status and LAN traffic, a physical key switch, a restart button and a smartcard reader. Each device has a unique physical key which can be used to operate the key switch. When the key is turned to the off position, the keypad is locked and the menus cannot be accessed via the front panel. The rear panel of the desktop unit has a serial port for console access, an Ethernet port for LAN-based, a temperature sensor and a socket for external power.

The Ethernet interface is primarily used to manage application keys and request cryptographic services, and cannot be used to perform remote administrative functions. The main administrative interface is accessed via the front panel on the device. Authentication using multiple smart cards is required to access the administrative functions available to either an *Operator* or a *Security Officer*.

Figure 2 – AEP Keyper HSM

Internally the TOE is comprised of a number of different components that combine to deliver the core security functionality and capabilities of the device. Figure 3 below illustrates the core components of the TOE that reside within the tamper resistant casing.

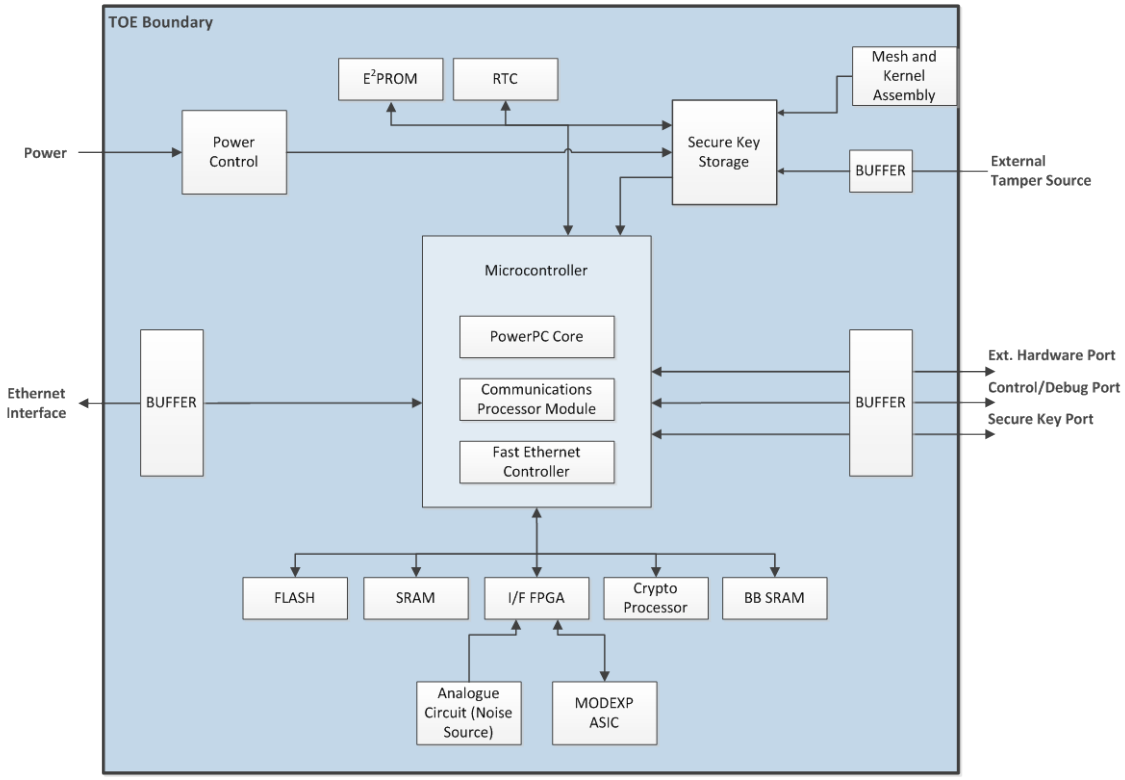

Figure 3 –High-Level Hardware Architecture

The key internal components of the TOE are described in the following table.

Table 1 – Key Hardware Components

Component	Description

Component	Description
Real Time Clock (RTC)	The real time clock is a battery backed device capable of providing a time and date stamp.
Secure Key Store (SKS)	The secure key store protects the most sensitive system keys (the keys used to wrap all other sensitive values and protect the integrity of the TOE).The secure key store also monitors all tamper sources. Upon detection of a tamper, the secure key store will erase its own store and generate a critical interrupt to the microcontroller, instructing it to erase all plaintext working keys in memory.
Microcontroller	An integrated communications microcontroller. The microcontroller includes the central processing unit of the TOE, a communications processor module and a fast Ethernet controller. The microcontroller is the central controller over the hardware devices and external interfaces, and facilitates communications between these devices.
FLASH	The FLASH is a bank of non-volatile memory which holds the TOE Firmware and the configuration data of the TOE. This consists of the boot code, the BIOS, the application and the encrypted FPGA configuration data.
Main Memory (SRAM)	The main memory is split into two sections: data memory and program memory. The data memory is always read/write, whereas the program memory is read only.
BBSRAM	Provides persistent storage for non-sensitive data, such as audit records, certificates, public keys and protected data. This memory is backed by a battery, and therefore can retain data once mains power is lost.
E^2PROM (EEPROM)	The E^2PROM holds several constant values that are unique to the TOE, which are set during manufacturing, such as the MAC address and serial number.
Crypto Processor	The Crypto Module provides a hardware implementation of symmetric encryption algorithms used by the TOE.
Noise Source	The noise source provides random values used to create seeds for the random number generator.
Modular Exponentiation ASIC (MODEXP ASIC)	A hardware circuit that can evaluate specific mathematical equations (modular exponentiation). It is used by the TOE to accelerate cryptographic functions.
I/F FPGA	The I/F FPGA provides an interface between the microcontroller, the analogue circuit (Noise Source) and MODEXP ASIC.
Power Control	The Power Control module regulates and supplies an operational voltage to the TOE's hardware components. This module also provides protection against side-channel attacks by attenuating any readable signals that may be output to the power inlet.

Component	Description
Mesh and Kernel Assembly	The physical barrier and associated circuitry to detect tamper events on the TOE. The mesh and kernel assembly forms the physical boundary of the TOE.
Buffer	Hardware circuitry that protects the components of the TOE that are involved in tamper response from damage by electrical attacks through the external interfaces.

1.7.2 Logical scope of the TOE

The logical scope of the TOE includes the Keyper firmware and configuration data. These components are stored in the TOE's FLASH memory.

The Keyper firmware includes the Keyper HSM Application, support libraries and other components, the operating environment, and a set of hardware drivers. The Keyper HSM Application (along with Keyper hardware components) implements and controls the security features which are the subject of this Security Target. Table 2 provides a brief overview of each of the security functions provided by the TOE.

The HSM Application is comprised of four primary processes: the Root Task, HSM Manager, HSM Crypto Manager and HSM UI Manager. Together these processes implement the TOE's user interfaces.

The Root Task's primary functions are to start and stop the three other processes and manages state transitions (e.g. from initialised state to operational state). The HSM UI Manager provides the main administrative interface, which is presented via a keypad/LCD and smartcard reader. The HSM UI Manager also provides key management functions, including backup/restore of application and system keys using smartcards. The HSM Manager is a TCP/IP server; it supports the HSM UI Manager, providing an interface for audit log extraction and firmware upload from a trusted network address. The HSM Crypto Manager process is also a TCP/IP server; it receives, processes, and returns requests for cryptographic services made from a trusted network address. The HSM Crypto Manager is the only interface that services cryptographic requests for application keys (other than backup/restore). The relationship between these processes is illustrated in Figure 4.

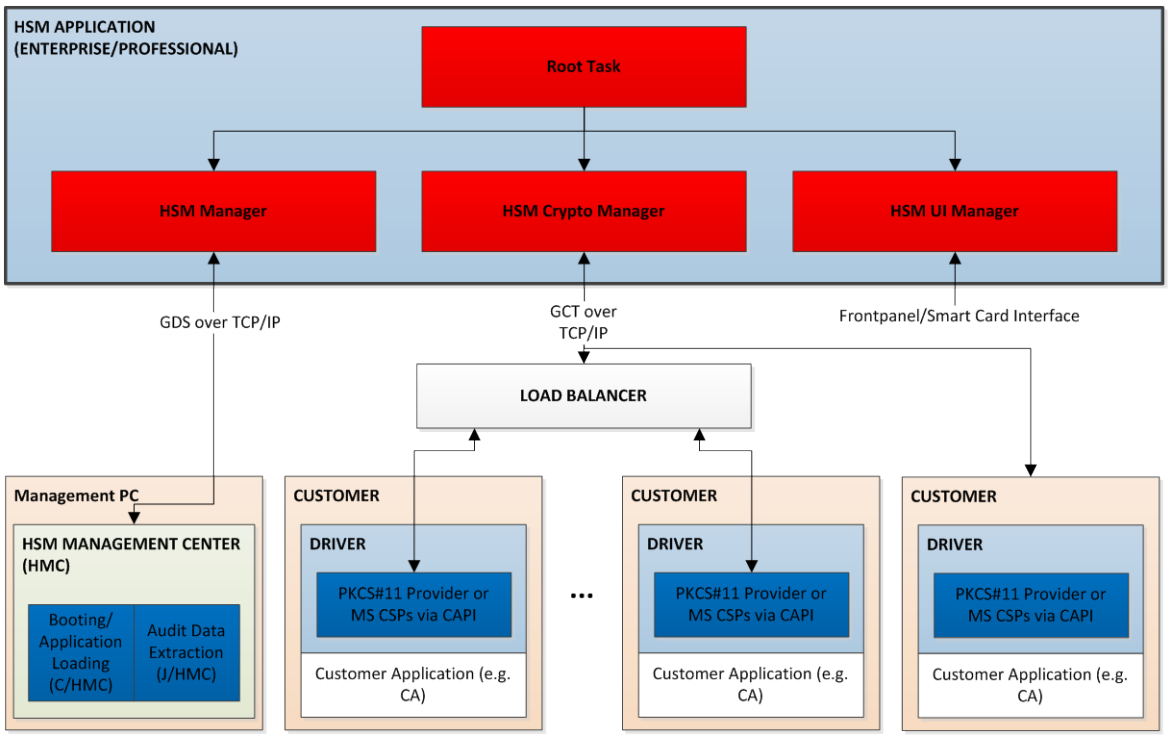

Figure 4 – Keyper HSM Application

A set of software components that support and provide functionality for the HSM Application are also included within the logical scope. These components include the boot loader and application loader, an audit module, a set of communications libraries used to transfer messages across a network and the crypto kernel. The crypto kernel implements a number of software cryptographic algorithms, and also provides access to the cryptographic hardware modules.

The software operating environment is the pSOS real time operating system (RTOS). The operating environment does not contribute to the implementation of the SFRs defined in section 5, other than to provide an underlying environment in which to execute the Keyper HSM Application.

The logical scope also includes a set of hardware drivers, which provide logical access to the hardware components outlined in section 1.7.1. The drivers are presented to the logical modules as APIs which provide a layer of abstraction that hides the lower-level logic required to interface with each hardware device.

Table 2 – TOE Security Function Overview

Security function	Description
Secure key management	The TOE provides the means to generate and manage cryptographic keys for use with its various cryptographic functions.
Secure key storage	The TOE provides the means to securely store sensitive cryptographic keys, using a dedicated hardware device and tamper mechanisms.

Security function	Description
Cryptographic operations	The TOE implements several cryptographic algorithms in hardware and software. These algorithms are used internally by the TOE and are also provided to users by the HSM Crypto Manager. The TOE implements asymmetric and symmetric encryption algorithms, key generation algorithms and cryptographic checksum algorithms and offers functions for data signing, encryption, secure storage and integrity checking.
User authentication	The TOE provides a mechanism for secure authentication using smart cards.
Security management	The TOE implements a set of functions and mechanisms to securely manage the TSF and TSF data.
Access control	The TOE implements two statically defined roles that are used primarily for segmenting access control. Each role has statically defined access to certain functions. Assumption of a role requires multiple smart card authentications.
Auditing	The TOE logs significant events to an internal audit log with at minimum a timestamp and error code.
Self-test	The TOE implements a set of self-tests that verify the TOE's hardware components, cryptographic algorithms, random number generator and firmware integrity.
Tamper protection	The TOE includes inbuilt tamper detection mechanisms that trigger tamper response mechanisms which wipe sensitive data and transition the TOE to a secure tamper state. An inbuilt battery allows the TOE to detect and react to tampers even when mains power is lost.

2 Conformance Claim (ASE_CCL)

The ST and TOE are conformant to version 3.1 (Rev 3) of the Common Criteria for Information Technology Security Evaluation.

The following specific conformance claims are made for this ST and TOE are:

a) **Part 2 extended.**

b) **Part 3 augmented, EAL4. + AVA_VAN.5**

While conformance cannot be claimed to a Protection Profile (PP) that has been developed and evaluated under the version 2.1 of the Common Criteria, this ST has been developed to align with all requirements specified in the *Cryptographic Module for CSP Signing Operations with Backup - Protection Profile* (Ref. [5]). This PP was issued by the issued by the European Committee for Standardization, Information Society Standardization System (CEN/ISSS) Electronic Signatures (E-SIGN) workshop.

The PP is for use by the European Commission in accordance with the procedure laid down in *Article 9 of the Directive 1999/93/EC of the European parliament and of the council of 13 December 1999 on a Community framework for electronic signatures* (Ref. [4]) as generally recognised standard for electronic-signature products in the Official Journal of the European Communities.

The PP is part of the CEN/ISSS workshop agreement (CWA) on trustworthy systems and is known as CWA 14167. This agreement comprises the following parts:

a) Part 1: System Security Requirements (Ref. [6]).

b) Part 2: Cryptographic Module for CSP Signing Operations with Backup – Protection Profile (Ref. [5]).

c) Part 3: Cryptographic Module for CSP Key Generation Services – Protection Profile (Ref. [7]).

d) Part 4: Cryptographic Module for CSP Signing Operations – Protection Profile (Ref. [8]).

3 Security problem definition (ASE_SPD)

3.1 Overview

This section describes the nature of the security problem that the TOE is designed to address. The security problem is described through:

a) a set of ***threats*** that the TOE must mitigate,

b) specific ***assumptions*** about the security aspects of the environment (both IT related and non-IT related elements) in which the TOE will operate, and

c) relevant ***organisational security policies*** that specify rules or guidelines that must be followed by the TOE and/or the operational environment.

3.2 Threats

In the context of this ST, the TOE has the following threat agents:

a) Individuals that have not been granted access to the application who attempt to gain access to information or functions provided by the TOE. This threat agent is considered an ***unauthorised individual***.

b) Individuals that are registered and have been explicitly granted access to the application who may attempt to access information or functions that they are not permitted to access. This threat agent is considered an ***authorised user***.

Identifier	Threat statement
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.CSP-SCD_Derive	***Deriving All or Parts of the CSP-SCD*** The most valuable asset the TOE has to protect is the CSP-SCD. The ability to derive all or parts of the CSP-SCD 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 using knowledge about the CSP-SCD generation and signing processes.

Identifier	Threat statement
T.CSP-SCD_Disclose	***Disclosing All or Part of the CSP-SCD*** Direct disclosure of the CSP-SCD or part of it presents a major threat to the TOE. This includes any way of disclosing all or part of the CSP-SCD over any physical or logical TOE interface.
T.CSP-SCD_Distortion	***Distortion of the CSP-SCD*** When the CSP-SCD is distorted, 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 Security Officer may also result in a compromise of the security of the TOE. The backup of user data and TSF data might be lost.

Identifier	Threat statement
T.Malfunction	***Malfunction of TOE*** Internal malfunction of TOE functions may result in the modification of DTBS-representation, misuse of TOE services, disclosure or distortion of CSP-SCD 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, the modification of DTBS-representation or the ability to misuse services of the TOE. Critical components might be: a) the central processing unit b) a coprocessor for accelerating cryptographic operations c) a physical random number generator d) storage devices used to store the CSP-SCD or the DTBS-representation e) 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, user management data, system data or TSF data.
T.Misuse_Sign	***Misuse of signature-creation function*** A user of the client application or of the TOE misuses the TOE service for signature-creation to sign with the SCP-SCD forged qualified certificates or forged certificate status information.

Identifier	Threat statement
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, to manipulate the 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.Signature_Forgery	***Forgery of digital signature*** An attacker exploits weaknesses in the cryptography and/or key management in the TOE in order to forge a CSP digital signature in a way that is not detectable by the verifier of the signature.

3.3 Organisational security policies

In the context of this ST, the following organisational security policies (OSPs) are used to provide the basis for security objectives that are most often desired by acquirers and users of the TOE.

Identifier	OSP statement
P.Algorithms	***Use of Approved Algorithms and Algorithm Parameter*** Only algorithms and algorithm parameter approved for being used for signature creation by trustworthy systems shall be used to perform cryptographic functions. A list of approved algorithms and parameters is given in the ESI standard Algorithms and Parameters for Secure Electronic Signatures (Ref. [9]). Where confidentiality protection is required such as for backup of CSP-SCD, only cryptographic strong algorithms and algorithm parameters shall be used.

3.4 Assumptions

The following assumptions provide the foundation for security objectives for the operational environment for the TOE.

Identifier	Assumption statement
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.

Identifier	Assumption statement
A.Correct_DTBS	***Correct DTBS Content Data*** DTBS-representation submitted to the TOE is assumed to be correct. This requires that the DTBS (e.g. the certificate content data) has been initialised correctly and maintains this correctness until it is passed to the TOE. This requires the DTBS to be correctly defined during the registration process, be transferred with integrity protection between the systems involved in the process (e.g. registration and certificate generation), be processed in a correct way by the client application, being hashed correctly (in the case the hashing is done by the client application and not by the TOE) and passed correctly to the TOE. The TOE environment will probably use its own mechanisms to ensure this correctness during processing and transmission. This will for example include mechanisms that can be used to verify the integrity and authenticity of user data when passed between different entities within the TOE environment. Specific instantiations of the TOE may have additional functions that can be used by the TOE environment to maintain the integrity of user data outside of the TOE, but those functions are not mandated by this Protection Profile
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 Operator role. Other users authorised for the TOE Operator 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 Operator services.

4 Security objectives (ASE_OBJ)

4.1 Overview

The security objectives are a concise statement of the intended response to the security problem defined in Section 3. There are security objectives for the TOE to address and additional objectives that provide specific direction for the intended environment in which the TOE is to operate.

4.2 Security objectives for the TOE

Identifier	Objective statements
O.AUDIT_CM	***Generation and Export of Audit Data*** The TOE shall audit the following events: a) TOE initialisation b) TOE start-up c) Unsuccessful authentication d) Modification of TOE management data e) Adding new users or roles f) Deleting users or roles g) Unsuccessful self-test operations h) Execution of the TSF self-tests i) Generation and export of backup data j) Import of backup keys k) Restore of backup data l) 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.
O.CSP-SCD_Secure	***Secure CSP-SCD Generation and Management*** The confidentiality and integrity of the CSP-SCD shall be ensured during their whole life time. The TOE shall ensure cryptographic secure CSP-SCD generation, use and management. This includes protection against disclosing completely or partly the CSP-SCD through any physical or logical TOE interface. The TOE implements secure cryptographic algorithms and parameters for the generation of CSP-SCD/CSP-SVD pairs chosen from [5].

Identifier	Objective statements
O.Check_Operation	***Check for Correct Operation*** The TOE shall perform 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.
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 Security 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 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.
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.Sign_Secure	***Secure advanced signature-creation*** The TOE creates signatures such as the advanced signature in qualified certificates that: a) do not reveal the CSP-SCD, and b) cannot be forged without knowledge of the CSP-SCD. The TOE implements secure cryptographic algorithms and parameters for the signing operation chosen from [5].
O.User_Authentication	***Authentication of Users interacting with the TOE*** The TOE shall be able to authenticate the users acting with a defined role, before allowing any access to TOE management operations.

4.3 Security objectives for the environment

Identifier	Objective statements
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 cannot 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_Human_Interface	***Reliable Human Interface*** If the client application provides a human interface and a communication path between human users and the TOE, the client application will ensure the confidentiality and integrity of the data transferred between the TOE and the human user.
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.

Identifier	Objective statements
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 the generation of signatures for qualified certificates or certificate status information. This includes the secure key generation / key import as well as the initial configuration of other TSF data like roles, users and user authentication information.
O.ENV_Secure_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.

4.4 TOE security objectives rationale

The following table demonstrates that all security objectives for the TOE trace back to the threats and OSPs in the security problem definition.

Threats/OSPs	Objectives	Rationale
P.Algorithms	O.CSP-SCD_Secure, O.Sign_Secure, O.Protect_Exported_Data	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. These properties are addressed in the objectives O.CSPSCD_Secure, O.Sign_Secure and O.Protect_Exported_Data.
T.Bad_SW	O.Check_Operation, O.Control_Services	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.
T.CSP-SCD_Derive	O.CSP-SCD_Secure, O.Sign_Secure, O.ENV_Protect_Access	T.CSP-SCD_Derive deals with the threat that the CSP-SCD 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 as well as the problem of timing channels or other signals of the CSP signing device that may carry information about the CSPSCD. Examples are power consumption or radiation. O.CSP-SCD_Secure is responsible to ensure that no information about the CSP-SCD is directly transmitted to any entity outside the TOE. O.Sign_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.

Threats/OSPs	Objectives	Rationale
T.CSP-SCD_Disclose	O.CSP-SCD_Secure, O.Check_Operation, O.Protect_Exported_Data, O.Sign_Secure, O.ENV_Protect_Access	T.CSP-SCD_Disclose deals with the threat of disclosing directly all or part of the CSP-SCD via the defined interfaces. This may happen 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 CSPSCD 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.Sign_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.CSP-SCD_Distortion	O.Check_Operation, O.Detect_Attack, O.Error_Secure, O.Protect_Exported_Data, O.ENV_Protect_Access	T.CSP-SCD_Distortion deals with the threat that the CSP-SCD 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 CSP-SCD. Therefore it has to be the objective to detect the distortion of the CSP-SCD, not only to prevent such a distortion. O.Check_Operation will ensure that the TOE will check the CSP-SCD regularly. O.Error_Secure will prevent the TOE to use distorted CSP-SCD after it has detected the distortion and O.Detect_Attack will prohibit the use of a distorted CSP-SCD after a physical attack (of course in the case of a physical attack the TOE will itself destroy the CSP-SCD 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 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 distortion by tampering.

Threats/OSPs	Objectives	Rationale
T.Data_Manipul	O.ENV_Application, O.ENV_Secure_Oper	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.

Threats/OSPs	Objectives	Rationale
T.Insecure_Init	O.Audit_CM, O.CSP-SCD_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_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.CSPSCD_Secure with respect to the secure CSP-SCD 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.

Threats/OSPs	Objectives	Rationale
T.Insecure_Oper	O.ENV_Personnel, O.ENV_Protect_Access, O.ENV_Secure_Oper	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. 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	O.Check_Operation, O.Error_Secure, O.ENV_Protect_Access, O.ENV_Recovery	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.

Threats/OSPs	Objectives	Rationale
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.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_Sign	O.Audit_CM, O.Control_Services, O.User_Authentication, O.ENV_Application	T.Misuse_Sign deals with the threat of misuse of the TOE to create a forged signature. This could be achieved, if an unauthorised user could invoke the signature function. O.Control_Services counters this threat by allowing network cryptographic services including signature services to be disabled and enabled by a TOE operator. O.User_Authentication ensures that application keys may only be directly exported by authenticated Security Officers. O.Audit_CM allows checking, if an authorised user has attempted to misuse the TOE by attempting to use functions he is not allowed to use. O.ENV_Application extends this protection to the end-users of the client application by their user authentication and access control.

Threats/OSPs	Objectives	Rationale
T.Phys_Manipul	O.Check_Operation, O.Detect_Attack, O.Error_Secure, O.ENV_Protect_Access	T.Phys_Manipul deals with physical manipulation of the TOE. An attacker may try to get access to the CSP-SCD 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.Signature_Forgery	O.Sign_Secure	T.Signature_Forgery deals with the threat that an attacker is able to generate a forged signature with the result that either a forged qualified signature or forged certificate status information is generated. While the threat of disclosing information about 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.Sign_Secure counters this threat by stating that it should not be possible to generate a valid signature without knowledge of the CSP-SCD.

4.5 Environment security objectives rationale

The following table demonstrates that all security objectives for the operational environment all trace back to assumptions or OSPs in the security problem definition.

Assumptions	Objectives	Rationale
A.Audit_Support	O.ENV_Audit, O.ENV_Personnel	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	O.ENV_Application, O.ENV_Secure_Oper	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	O.ENV_Recovery, O.ENV_Secure_Init, O.ENV_Secure_Oper	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	O.ENV_Application, O.ENV_Human_Interface	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 and O.ENV_Human_Interface address 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).

5 Security functional requirements (ASE_REQ)

5.1 Overview

This section defines the security requirements satisfied by the TOE. Each requirement has been extracted from version 3.1 (Rev 3) of the CC, part 2 providing functional requirements and part 3 providing assurance requirements.

Part 2 of the CC defines an approved set of operations that may be applied to security functional requirements. Following are the approved operations and the document conventions that are used within this ST to depict their application:

- **Assignment.** The assignment operation provides the ability to specify an identified parameter within a requirement. Assignments are depicted using bolded text and are surrounded by square brackets as follows [**assignment**].

- **Selection.** The selection operation allows the specification of one or more items from a list. Selections are depicted using bold italics text and are surrounded by square brackets as follows [***selection***].

- **Refinement.** The refinement operation allows the addition of extra detail to a requirement. Refinements are indicated using bolded text, for **additions**, and strike-through, for ~~deletions~~.

- **Iteration.** The iteration operation allows a component to be used more than once with varying operations. Iterations are depicted by placing a letter at the end of the component identifier as follows FDP_IFF.1a and FDP_IFF.1b.

The security functional requirements are expressed using the notation stated in Section 5.1 above and are identified in the table below.

Identifier	Title
Security audit (FAU)	
FAU_GEN.1	Audit data generation
FAU_GEN.2	User identity association
Cryptographic support (FCS)	
FCS_CKM.1	Cryptographic key generation
FCS_CKM.2	Cryptographic key distribution

Identifier	Title
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1a	Cryptographic operation (SIGN)
FCS_COP.1b	Cryptographic operation (ENCRYPTION)
FCS_COP.1c	Cryptographic operation (INTEGRITY)
FCS_RND.1	Quality metrics for random numbers
User data protection (FDP)	
FDP_ACC.1a	Subset access control (CRYPTO)
FDP_ACF.1a	Security attribute based access control (CRYPTO)
FDP_ACC.1b	Subset access control (BACKUP)
FDP_ACF.1b	Security attribute based access control (BACKUP)
FDP_ETC.1	Export of user data without security attributes
FDP_IFC.1a	Subset information flow control (BACKUP)
FDP_IFF.4a	Partial elimination of illicit information flows (BACKUP)
FDP_IFC.1b	Subset information flow control (CRYPTO)
FDP_IFF.4b	Partial elimination of illicit information flows (CRYPTO)
FDP_RIP.1	Subset residual information protection
FDP_SDI.2	Stored data integrity monitoring and action
Identification and authentication (FIA)	
FIA_SOS.1	Verification of secrets
FIA_UAU.1	Timing of authentication
FIA_UID.1	Timing of identification
Security management (FMT)	
FMT_MTD.1a	Management of TSF data
FMT_MTD.1b	Management of TSF data
FMT_SMF.1	Specification of management functions

Identifier	Title
FMT_SMR.1	Security roles
Protection of the TOE security functions (FPT)	
FPT_FLS.1	Failure with preservation of secure state
FPT_ITC.1	Inter-TSF confidentiality during transmission
FPT_ITI.1	Inter-TSF detection of modification
FPT_PHP.2	Notification of physical attack
FPT_PHP.3	Resistance to physical attack
FPT_RCV.1	Manual recovery
FPT_STM.1	Reliable time stamps
FPT_TST.1	TSF testing

5.2 Security audit (FAU)

5.2.1 FAU_GEN.1 Audit data generation

Hierarchical to:	No other components
Dependencies:	FPT_STM.1 Reliable time stamps
FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the [***not specified***] level of audit; and c) [**Initialisation of the TOE;** d) **Start-up after power up;** e) **Shutdown of the TOE;** f) **Creation of the audit log;** g) **Export of the audit log;** h) **TOE state changes;** i) **Execution of management functions (FMT_SMF.1);** j) **Cryptographic key destruction if performed via the keypad (FCS_CKM.4): CSP-SCD destruction, destruction of backup key(s);** k) **Backup and import : Use of the backup function;** l) **Smart card authentication failure: reaching of the threshold for the**

	unsuccessful authentication attempts and the actions; m) **Timing of authentication (FIA_UAU.1): all unsuccessful use of the authentication mechanism;** n) **Management of TSF data (FMT_MTD.1b): All modifications to the values of TSF data;** o) **Failure with preservation of secure state (FPT_FLS.1): Failure detection of the TSF and secure state;** p) **Inter-TSF detection of modification (FPT_ITI.1): The detection of modification of imported TSF data;** q) **Notification of physical attack (FPT_PHP.2): Detection of intrusion**].
FAU_GEN.1.2	The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**event number**].
Notes:	This SFR specifies those events that have a record captured in the audit trail. In addition, this requirement specifies the information that must be captured within an audit record, ensuring that the audit trail is of value.

5.2.2 FAU_GEN.2 User identity association

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation, FIA_UID.1 Timing of Identification
FAU_GEN.2.1	For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.
Notes:	Each TOE function that may be performed by a user is assigned to a single role only, therefore the TOE may associate each audit event that may result from the actions of an identified user with the role which caused that event. However, an audit entry is created when each individual smart card is authenticated containing the serial number of that smart card, therefore it is also possible to associate audit events with the individual smart cards that were used for authentication.

5.3 Cryptographic support (FCS)

5.3.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to:	No other components
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**a) Deterministic Random Bit Generator,** **b) RSA Key Pair Generation** **c) DSA Key Pair Generation** and specified cryptographic key sizes [**a) From Deterministic Random Bit Generator:** **a) 56 bit DES,** **b) 112 and 168 bit TDES, and** **c) 128, 192 or 256 bit AES** **b) From RSA Key Pair Generation:** **a) 512 to 4096 bit RSA (in 32 bit steps) with and without CRT. Public exponents of 3,17 and 65537** **c) From DSA Key Pair Generation:** **a) 512 to 1024 bit DSA modulus inclusive (in 64 bit steps)]** that meet the following: [**a) FIPS PUB 186-2, Digital Signature Standard (Ref. [13]), Appendix 3.1,** **b) ANSI X9.31-1998, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA), and** **c) FIPS PUB 186-2, Digital Signature Standard (Ref. [13]).**].
Notes:	None.

5.3.2 FCS_CKM.2 Cryptographic key distribution

Hierarchical to:	No other components
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.2.1	The TSF shall distribute cryptographic keys in accordance with a specified

	cryptographic key distribution method [**a) AAK Backup and Restore** **b) SMK Backup and Restore** **c) Application Keys Backup and Restore**] that meets the following **rules**: **[** **a) For AAK:** **i. backed up and restored in 2 of 2 through to 9 of 9 component form, and** **ii. all smartcards in the set must be presented to re-create the AAK key.** **b) For SMK:** **i. restored in 2 through 9 component form, and** **ii. a minimum of two (M) of four (N) smartcards to a maximum of nine (M) of nine (N) smartcards need to be present to rebuild the SMK.** **c) For Application Keys:** **i. backed up and restored via smartcard encrypted with the SMK** **ii. import and export via the network**].
Notes:	None.

5.3.3 FCS_CKM.4 Cryptographic key destruction

Hierarchical to:	No other components
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**overwrite**] that meets the following: **[positively overwrites the master keys and plain text keys]**.
Notes:	The TOE destroys the CSP-SCD and all other plaintext secret or private keys if the TOE enters the tamper state. Keys may also be destroyed by the *security officer* by initialising the TOE.

5.3.4 FCS_COP.1a Cryptographic operation (SIGN)

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

	FCS_CKM.4 Cryptographic key destruction
FCS_COP.1a.1	The TSF shall perform [**digital signature-creation**] in accordance with a specified cryptographic algorithm [**a) RSA** **b) DSA**] and cryptographic key sizes [**a) between 1024 and 4096 bit modulus (32 bit steps)** **b) between 512 to 1024 bit modulus (64 bit steps)**] that meet the following: [**a) ANSI X9.31-1998, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA).** **b) FIPS PUB 186-2, Digital Signature Standard (Ref. [13])**].
Notes:	None

5.3.5 FCS_COP.1b Cryptographic operation (ENCRYPTION)

Hierarchical to:	No other components
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1b.1	The TSF shall perform [**encryption and decryption**] in accordance with a specified cryptographic algorithm [**a) RSA,** **b) DES** **c) 3DES, and** **d) AES**] and cryptographic key sizes [**a) 1024 to 4096 bit (RSA)** **b) 56 bit (DES),** **c) 112 and 168 bits (3DES), and** **d) 128, 192 256 bits (AES)**] that meet the following: [**a) ANSI X9.31-1998, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA));** **b) FIPS Publication 46-3, Data Encryption Standard (Ref. [11])** **c) FIPS Publication 46-3, Data Encryption Standard (Ref. [11]); and** **d) FIPS Publication 197, Advanced Encryption Standard (Ref. [12])**].

Notes:	None

5.3.6 FCS_COP.1c Cryptographic operation (INTEGRITY)

Hierarchical to:	No other components
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1c.1	The TSF shall perform [**calculation and verification of cryptographic checksums**] in accordance with a specified cryptographic algorithm [**a) SHA-1, or** **b) SHA-2**] and cryptographic key sizes [**N/A**] that meet the following: [**a) FIPS Publication 180-1, Secure Hash Standard (Ref. [14])** **b) FIPS Publication 180-1, Secure Hash Standard (Ref. [14])**].
Notes:	The TOE provides a range of configurable cryptographic checksum algorithms.

5.3.7 FCS_RND.1 Quality metrics for random numbers

Hierarchical to:	No other components
Dependencies:	FPT_TST.1 TSF testing
FCS_RND.1.1	The TSF shall provide a mechanism for generating random numbers that meet [**FIPS PUB 186-2, Digital Signature Standard (Ref. [13]), Appendix 3.1**].
FCS_RND.1.2	The TSF shall be able to enforce the use of TSF-generated random numbers for [**FCS_CKM.1**].
Notes:	The TOE implements a random number generator that is certified as compliant to FIPS PUB 186-2 (Ref. [13]), this standard refers out to NIST SP 800-90 (Ref. [15]) which provides recommendations for random number generation using deterministic bit generators

5.4 User data protection (FDP)

5.4.1 FDP_ACC.1a Subset access control (CRYPTO)

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

FDP_ACC.1a.1	The TSF shall enforce the [**CRYPTO-SFP**] on [**a) Subjects:** **i. Role** **b) Objects:** **i. CSP-SCD** **ii. CSP-SVD** **iii. DTBS representation** **c) Operations:** **i. Generate CSP-SCD/CSP-SVD pair (FCS_CKM.1)** **ii. Destroy CSP-SCD and CSP-SVD (FCS_CKM.4)** **iii. Sign DTBS representation (FCS_COP.1a)** **iv. Export CSP-SVD (FCS_CKM.2)**].
Notes:	None

5.4.2 FDP_ACF.1a Security attribute based access control (CRYPTO)

Hierarchical to:	No other components
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1a.1	The TSF shall enforce the [**CRYPTO-SFP**] to objects based on the following:[**a) Subjects:** **i. Role**].
FDP_ACF.1a.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**a) An anonymous user with network access to the unit is permitted to create application specific keys.** **b) An anonymous user with network access to the unit is allowed to generate (FCS_CKM.1) the objects CSP-SCD and CSP-SVD.** **c) Role Security Officer is allowed to destroy (FCS_CKM.4) the objects CSP-SCD and CSP-SVD.** **d) An anonymous user with network access to the unit and a reference to a specific application security key is permitted to destroy the object.** **e) Role Security Officer is allowed to export CSP-SCD if key export not disabled during initialisation. An anonymous user who knows the reference of the key can also delete it via the network if and only if it has been allowed during the creation of the key and the API setting allows export.**

	f) Role Operator can enable or disable network cryptographic operations. **g) An anonymous user who knows the reference of the CSP-SCD can then use it to create a signature of the DTBS representation(FCS_COP.1a). An anonymous who knows the reference of the CSP-SVD can use it to verify the signature (if network operations are enabled).**
FDP_ACF.1a.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**none**].
FDP_ACF.1a.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**a) Role Operator is not permitted to destroy (FCS_CKM.4) the objects CSP-SCD and CSP-SVD**].
Notes:	There is no concept of individual users by the TOE; although individual smart card identifiers are logged during authentication. Authentication to the roles of Security Officer or Operator requires between 2 and 9 smart cards (depending on security policy). An anonymous user is any user of the TOE who is not identified or authenticated.

5.4.3 FDP_ACC.1b Subset access control (BACKUP)

Hierarchical to:	No other components
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1b.1	The TSF shall enforce the [**BACKUP-SFP**] on [**a) Subjects:** **i. Role** **b) Objects:** **i. CSP-SCD,** **ii. backup key(s),** **iii. backup data** **c) Operations:** **i. backup ,** **ii. restore ,** **iii. backup key entry (FCS_CKM.2)**].
Notes:	None

5.4.4 FDP_ACF.1b Security attribute based access control (BACKUP)

Hierarchical to:	No other components
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1b.1	The TSF shall enforce the [**BACKUP-SFP**] to objects based on the following:[**a) Subjects:** **i. Role**].
FDP_ACF.1b.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**a) Role Security Officer is permitted, under 2 to 9 person control, to backup CSP-SCD and CSP-SVD.** **b) Role Security Officer is permitted, under 2 to 9 person control, to import CSP-SCD and CSP-SVD.** **c) Role Security Officer is permitted, under 2 to 9 person control, to back up and import the SMK key (FCS_CKM.2)**]. **d) Role Security Officer is permitted, under 2 to 9 person control, to back up the AAK key (FCS_CKM.2)**]. **e) An anonymous user is permitted to import or replace the AAK when the TOE is in initialised state.**
FDP_ACF.1b.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**none**].
FDP_ACF.1b.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**a) Role Operator is not permitted to backup CSP-SCD,** **b) Role Operator is not permitted to import CSP-SCD ,** **c) Role Operator is not permitted to back up and import the SMK or AAK keys (FCS_CKM.2)**].
Notes:	There is no concept of individual users by the TOE; although individual smart card identifiers are logged during authentication. An anonymous user is any user of the TOE who is not identified or authenticated.

5.4.5 FDP_ETC.1 Export of user data without security attributes

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

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.
Notes:	None

5.4.6 FDP_IFC.1a Subset information flow control (BACKUP)

Hierarchical to:	No other components
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1a.1	The TSF shall enforce the [**Side-channel of backup-functions SFP**] on [**a) Subjects:** **i. All** **b) Information:** **i. CSP-SCD;** **ii. backup (FCS_COP.1b, FCS_COP.1c),** **iii. restore (FCS_COP.1b, FCS_COP.1c), and** **iv. key entry (FCS_CKM.2)**].
Notes:	None

5.4.7 FDP_IFF.4a Partial elimination of illicit information flows (BACKUP)

Hierarchical to:	FDP_IFF.3 Limited illicit information flows
Dependencies:	FDP_IFC.1 Subset information flow control
FDP_IFF.4a.1	The TSF shall enforce the [**Side-channel of backup-functions SFP**] to limit the capacity of [**covert channels information flow of:** **a) the backup function including encryption of the backup data ,** **b) the backup key(s) entry (FCS_CKM.2), and** **c) the encryption and decryption of the backup data (FCS_COP.1b)** **through physical behaviour of the TOE interfaces and emanation compromising information about the CSP-SCD**] to a [**limit of zero**].
FDP_IFF.4a.2	The TSF shall prevent [**side-channels information flow within the backup data about the CSP-SCD**].
Notes:	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.

5.4.8 FDP_IFC.1b Subset information flow control (CRYPTO)

Hierarchical to:	No other components
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1b.1	The TSF shall enforce the [**Side-channel of crypto-functions SFP**] on [**a) Subjects:** **i. All** **b) Information:** **i. CSP-SCD;** **ii. generation of CSP-SCD/SVD pair (FCS_CKM.1),** **iii. destruction of CSP-SCD (FCS_CKM.4),** **iv. signing DTBS representation (FCS_COP.1a)**].
Notes:	None

5.4.9 FDP_IFF.4b Partial elimination of illicit information flows (CRYPTO)

Hierarchical to:	FDP_IFF.3 Limited illicit information flows
Dependencies:	FDP_IFC.1 Subset information flow control
FDP_IFF.4b.1	The TSF shall enforce the [**Side-channel of crypto-functions SFP**] to limit the capacity of [**side-channels information flow of:** **a) the CSP-SCD/SVD generation (FCS_CKM.1),** **b) the signature-creation (FCS_COP.1a),** **through physical behaviour of the TOE interfaces and emanation compromising information about the CSP-SCD**] to a [**limit of zero**].
FDP_IFF.4b.2	The TSF shall prevent [**side-channels information flow within the data exported by the following TSF functions:** **a) generation of the CSP-SCD / SVD pair (FCS-CKM.1), and** **b) signature-creation function (FCS-COP.1a) for the CSP-SCD**].
Notes:	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.

5.4.10 FDP_RIP.1 Subset residual information protection

Hierarchical to:	No other components
Dependencies:	No dependencies
FDP_RIP.1.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the ***[de-allocation of the resource from]*** the following objects: [**any objects in the Red Store memory**].
Notes:	The Red Store is a managed segment of main memory that holds keys and other sensitive values when in use.

5.4.11 FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies
FDP_SDI.2.1	The TSF shall monitor user data stored within the TSC for [**integrity errors**] on ~~all objects~~ **application keys**, based on the following attributes: **[stored key metadata]**.
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF shall **[not use the application keys]**.
Notes:	Application keys are encrypted with the ISMK and stored with key metadata. When a key is to be used, it is decrypted and the stored metadata is compared against the key.

5.5 Identification and authentication (FIA)

5.5.1 FIA_SOS.1 Verification of secrets

Hierarchical to:	No other components
Dependencies:	No dependencies
FIA_SOS.1.1	The TSF shall provide a mechanism to **verify that** ~~secrets~~ **smartcard attributes** meet [**a) PIN Length is between four (4) and eight (8) characters** **b) Incorrect PIN entry limit is less than five (5)].**
Notes:	None

5.5.2 FIA_UAU.1 Timing of authentication

Hierarchical to:	No other components
Dependencies:	FIA_UID.1 Timing of identification
FIA_UAU.1.1	The TSF shall allow [**a) start-up,** **b) self-test (FPT_TST.1),** **c) issue security officer smart cards, import an AAK, enter operational state or set key export state when in initialised state,** **d) cryptographic key generation via network,** **e) cryptographic key destruction via network,** **f) cryptographic operation (SIGN) via network,** **g) cryptographic operation (ENCRYPTION) via network,** **h) cryptographic operation (INTEGRITY) via network,** **i) cryptographic key distribution via network,** **j) operational tamper event,** **k) positive tamper event,** **l) output of self-test results and tamper status via serial port,** **m) detection of the secure blocking state (FPT_FLS.1),** **n) detection of violation of physical integrity (FPT_PHP.2), and** **o) 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.
Notes:	The TOE must be set to *online mode* by an *operator* to enable access to network cryptographic operations.

5.5.3 FIA_UID.1 Timing of identification

Hierarchical to:	No other components
Dependencies:	No dependencies
FIA_UID.1.1	The TSF shall allow [**a) start-up,** **b) self-test (FPT_TST.1),** **c) issue security officer smart cards, import an AAK, enter operational state or set key export state when in initialised state,** **d) cryptographic key generation (FCS_CKM.1) via network,** **e) cryptographic key distribution (FCS_CKM.2) via network,** **f) cryptographic key destruction (FCS_CKM.4) via network,** **g) cryptographic operation (SIGN) (FCS_COP.1a) via network,** **h) cryptographic operation (ENCRYPTION) (FCS_COP.1b) via network,** **i) cryptographic operation (INTEGRITY) (FCS_COP.1c) via network,** **j) operational tamper event,** **k) positive tamper event,** **l) output of self-test results and tamper status via serial port,** **m) detection of the secure blocking state (FPT_FLS.1), and** **n) 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.
Notes:	The TOE must be set to *online mode* by an *operator* to enable access to network cryptographic operations.

5.6 Security management (FMT)

5.6.1 FMT_MTD.1a Management of TSF data

Hierarchical to:	No other components
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1a.1	The TSF shall restrict the ability to [***delete***] ~~the~~ [**all TSF data**] to [**Security Officer**].
Notes:	The unit can be set back to initialised state by the Security Officers which deletes all internal keys (including the security officers) and associated data. A new AAK may then be generated and new Security Officer smart cards can be issued.

5.6.2 FMT_MTD.1b Management of TSF data

Hierarchical to:	No other components
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1b.1	The TSF shall restrict the ability to [**[generate]**] the [**SMK**] to [**Security Officer**].
Notes:	None.

5.6.3 FMT_SMF.1 Specification of management functions

Hierarchical to:	No other components
Dependencies:	No dependencies
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: [**a) Issue Security Officer and Operator smart cards,,** **b) Delete all TSF data (FMT_MTD.1a),** **c) Import and export Application keys, AAK and SMK,** **d) Generate SMK (FMT_MTD.1b) and AAK,** **e) Set SMK algorithm,** **f) Set key export rules (when in initialised state),** **g) Manage online mode,** **h) Manage FIPS mode,** **i) Manage smart cards,**

	j) View key information, **k) Manage cryptographic service network availability,** **l) Manage network configuration,** **m) Change real time clock,** **n) Output TOE status,** **o) Import and export TOE configuration, and** **p) Recover from operational tamper state.**].
Notes:	None.

5.6.4 FMT_SMR.1 Security roles

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.1.1	The TSF shall maintain the roles [**a) Security Officer, and** **b) Operator]**
FMT_SMR.1.2	The TSF shall be able to associate users with roles.
Notes:	The TSF issues sets of smart cards that are associated with roles. Each smart card has a unique serial number.

5.7 Protection of the TOE security functions (FPT)

5.7.1 FPT_FLS.1 Failure with preservation of secure state

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: [**any failure detected by FPT_TST.1 or failure induced by physical conditions**].
Notes:	None.

5.7.2 FPT_ITC.1 Inter-TSF confidentiality during transmission

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITC.1.1	The TSF shall protect ~~all TSF data~~ Application Keys transmitted from the TSF to a remote trusted IT product from unauthorised disclosure during transmission.
Notes:	This SFR aims to address the protection of the TSF data exported as part of the backup process.

5.7.3 FPT_ITI.1 Inter-TSF detection of modification

Hierarchical to:	No other components.
Dependencies:	No dependencies.
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: **[** **a) Distributed SHA-1 hash for the AAK and SMK** **b) 3DES or AES encryption using the SMK for Application keys]**
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~~ [**refuse to import the TSF data and record an audit record**] if modifications are detected.
Notes:	This SFR addresses the integrity protection of the TSF data if they are imported as part of the backup data.

5.7.4 FPT_PHP.2 Notification of physical attack

Hierarchical to:	FPT_PHP.1 Passive detection of physical attack
Dependencies:	FMT_MOF.1 Management of security functions behaviour
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 [**the TOE casing, temperature sensor and power control module**], the TSF shall monitor the devices and elements and notify [**anyone**] when physical tampering with the TSF's devices or TSF's elements has occurred.
Notes:	The TSF shall detect physical tampering performed by opening the device or removal of a cover, application of extreme temperatures, total power failure or abnormal voltage levels.

5.7.5 FPT_PHP.3 Resistance to physical attack

Hierarchical to:	No other components
Dependencies:	No dependencies
FPT_PHP.3.1	The TSF shall ~~resist~~ **react to** [**a) External (mains) voltage outside of specified fatal range,** **b) External (mains) voltage outside of specified operational range,** **c) Storage temperature outside of normal operating range,** **d) Physical breach of the tamper casing,** **e) Operational temperature outside of normal operating range,** **f) External tamper switch triggered,** **g) Power fluctuation, or** **h) Total power failure (both internal battery and mains power).**] to the [**components of the TOE that:** **a) generates CSP-SCD (FCS_CKM.1),** **b) creates the signature with CSP-SCD (FCS_COP.1),** **c) stores CSP-SCD, and** **d) stores other secret or private keys]** by responding automatically such that the SFRs are always enforced.
Notes:	The TSF shall react to a tamper condition by destroying plaintext CSP-SCD and other confidential secret and private keys.

5.7.6 FPT_RCV.1 Manual recovery

Hierarchical to:	No other components
Dependencies:	AGD_OPE.1 Operational user guidance
FPT_RCV.1.1	After **[operational tamper or service discontinuity**] the TSF shall enter a ~~maintenance mode~~ **tamper state** where the ability to return to a secure state is provided.
Notes:	Service discontinuity relates to loss of power to the device. Operational tamper relates to the triggering of a non-fatal tamper event.

5.7.7 FPT_STM.1 Reliable time stamps

Hierarchical to:	No other components
Dependencies:	No dependencies
FPT_STM.1.1	The TSF shall be able to provide reliable time stamps.
Notes:	This requirement is included to support the generation of suitable audit records for the application.

5.7.8 FPT_TST.1 TSF testing

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TST.1.1	The TSF shall run **the following** ~~a~~ suite of self-tests**: [** **a) Installation:** **a) Software/firmware integrity test** **b) Power-up:** **a) Software/firmware integrity test** **b) Cryptographic algorithm test** **c) Random number generator test** **d) Critical functions test** **c) Conditional:** **a) Pair-wise consistency test for public and private keys** **b) Continuous random number generator test]** [***during initial start-up***] 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 [**no TSF data**].
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of [***stored TSF executable code***].
Notes:	The TSF performs self-tests according to FPT_TST.1 to ensure that the TOE is functioning properly. This SFR has been refined to specify the exact self-test activities that are performed by the TOE.

5.8 Dependency rationale

SFR	Dependencies	Rationale
FAU_GEN.1	FPT_STM.1 Reliable time stamps	Included
FAU_GEN.2	FAU_GEN.1 Audit data generation	Included
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Included Included Included
FCS_CKM.2	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	 Included Included
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	 Included
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	 Included Included
FCS_RND.1	FPT_TST.1 TSF testing	Included
FDP_ACC.1	FDP_ACF.1 Security attribute based access control	Included

SFR	Dependencies	Rationale
FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Included Not Included; the following rationale provides an explanation for this exclusion: The TOE has no concept of individual users, therefore there are no security attributes associated with a user to be initialised. Access control in the TOE is not based upon user security attributes, but upon two discrete, statically defined roles: the *security officer* and the *operator*. TOE functionality that is not assigned to either of these roles is accessible to anonymous (unidentified & unauthenticated) users. In order to access controlled functionality, TOE users are authenticated to one of these two roles through the use of smart cards. Sets of smart cards may be issued for each role (a minimum of two cards in a set are required to authenticate). Each smart card contains information identifying the set it belongs to and its type. Each *security officer* or *operator* smart card set is directly related to a single key: the Authorisation Key (AAK). This key is stored in the TOE, and may be generated or imported during initialisation of the TOE. The information on each smart card is protected by a key derived from the AAK, known as an Authorisation Sub Key (ASK). Each smart card stores its own ASK in a write-protected area. Multiple ASKs (stored in multiple cards) must be presented in order to authenticate against the AAK.
FDP_ETC.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Included Included

SFR	Dependencies	Rationale
FDP_IFC.1	FDP_IFF.1 Simple security attributes	Not Included. The information flow control SFPs specified by FDP_IFC.1 relate to flows of information in covert channels. Due to the nature of covert channels, the flow of information is not controlled, therefore FDP_IFF.1 is not applicable. FDP_IFF.4 describes the partial elimination of information flow on these channels.
FDP_IFF.4	FDP_IFC.1 Subset information flow control	Included
FDP_RIP.1	No dependencies	-
FDP_SDI.2	No dependencies	-
FIA_SOS.1	No dependencies	-
FIA_UAU.1	FIA_UID.1 Timing of identification	Included
FIA_UID.1	No dependencies	-
FMT_MTD.1	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Included Included
FMT_SMF.1	No dependencies	-
FMT_SMR.1	FIA_UID.1 Timing of identification	Included
FPT_FLS.1	No dependencies	-
FPT_ITC.1	No dependencies	-
FPT_ITI.1	No dependencies	-

SFR	Dependencies	Rationale
FPT_PHP.2	FMT_MOF.1 Management of security functions behaviour	Not Included. FPT_PHP.2 informs the local user about detected tampering attempt. No management of security functions behaviour is required.
FPT_PHP.3	No dependencies	-
FPT_RCV.1	AGD_OPE.1 Operational user guidance	Included
FPT_STM.1	No dependencies	-
FPT_TST.1	No dependencies	-

5.9 Mapping of SFRs to security objectives for the TOE

Security objective	Mapped SFRs	Rationale
O.Audit_CM	FAU_GEN.1, FAU_GEN.2,	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...
O.Protect_Exported_Data	FAU_GEN.1, FAU_GEN.2, FCS_CKM.2, FCS_COP.1b, FCS_COP.1c, FDP_ACC.1b, FDP_ACF.1b, FDP_ETC.1, FDP_IFC.1a, FDP_IFF.4a, FPT_ITC.1, FPT_ITI.1	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 functions require the confidentiality protection of backup data. The backup of CSP-SCD. The confidentiality of the TSF data as part of the backup data is implemented by the SFR FPT_ITC.1and SFR FPT_ITI.1.

Security objective	Mapped SFRs	Rationale
O.CSP-SCD_Secure	FCS_CKM.1, FCS_CKM.4, FCS_COP.1a, FCS_RND.1, FDP_ACC.1a, FDP_ACF.1a, FDP_IFC.1b, FDP_IFF.4b, FDP_RIP.1, FDP_SDI.2	O.SCP-SCD_Secure (secure CSP-SCD generation and management) addresses the confidentiality and integrity of the CSP-SCD which shall be ensured during their whole life time. The SFR ensure the cryptographic secure CSP-SCD generation by FCS_CKM.1 and FCS_RND.1 as well as operation by FCS_COP.1a according to the list of approved algorithms and parameters. The confidentiality and integrity of the CSP-SCD 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 after operational life time. The all CSP-SCD management and operation is under access control of the SFR FDP_ACC.1aand FDP_ACF.1a. The TOE shall protect CSP-SCD against side-channels by the SFR FDP_IFC.1b and FDP_IFF.4b.
O.Check_Operation	FAU_GEN.1, FPT_TST.1	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 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 Security Officer) about the performed self-tests and their results. The FPT_TST.1 includes checks of the executable code.
O.Control_Services	FDP_ACC.1a, FDP_ACC.1b, FDP_ACF.1a, , FDP_ACF.1b, FMT_MTD.1a, FMT_SMF.1, FMT_SMR.1, FPT_TST.1	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.1a and FDP_ACF.1a for the cryptographic functions (Crypto-SFP), b) FDP_ACC.1a and FDP_ACF.1a for the audit function (Audit-SFP), c) FDP_ACC.1ba and FDP_ACF.1b for the backup function (Backup-SFP with the roles Auditor, Security Officer and Operator as defined by the SFR FMT_SMR.1. The SFR FMT_MTD.1a, FMT_MTD.1b and FMT_SMF.1 assign the management functions for the cryptographic to the Security Officer and audit functions to the Auditor. FMT_SMR.1 is to helps define the users of the TOEs and FPT_TST.1 is to ensure that the TOE can perform self-test functions.

Security objective	Mapped SFRs	Rationale
O.Detect_Attack	FPT_PHP.2, FPT_PHP.3	O.Detect_Attack (detection of physical attacks) addresses the detection of physical tampering attempts and the secure destruction of the CSP-SCD 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.
O.Error_Secure	FPT_FLS.1, FPT_RCV.1, FPT_TST.1	O.Error_secure (secure state in case of error) addresses a secure state and protection of CSP-SCD confidentiality whenever the TOE detects an error. The SFR 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.
O.Sign_Secure	FCS_COP.1a, FDP_IFC.1b, FDP_IFF.4b	O.Sign_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.1a with reference to the list of approved algorithms and parameters. The SFR FDP_IFC.1b and FDP_IFF.4b requires TSF to prevent illicit information flow about the CSP-SCD through side-channels in the signatures.
O.User_Authentication	FIA_SOS.1, FIA_UAU.1, FIA_UID.1,FMT_MTD.1b, FMT_SMF.1	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_SOS.1 ensures the verification of the quality of the secret used for authentication. SFRs FMT_MTD.1b and FMT_SMF.1 provide management functions for identification.

6 Security assurance requirements (ASE_REQ)

This ST implements the Security Assurance Requirements (SARs) of the EAL4 package and augments this package with the inclusion of the following assurance requirements:

a) AVA_VAN.5 Advanced methodical vulnerability analysis

EAL4 provides assurance by a full security target and an analysis of the SFRs in that ST, using a functional and complete interface specification, guidance documentation and a description of the modular design of the TOE. The full implementation is also provided to the evaluator so that analysis can be conducted of an evaluator-selected subset, so that security behaviour can be understood and potential vulnerabilities identified.

The analysis is supported by independent testing of the TSF, which can be based on evidence of developer testing of the functions of the TOE. In addition, the evaluators will conduct a vulnerability analysis using all provided inputs and ensure that the TOE is resistant to penetration attackers with an ***high*** attack potential due to the augmentation of EAL4 with AVA_VAN.5. EAL4 also provides assurance through the use of development environment controls and additional TOE configuration management including automation, and evidence of secure delivery procedures.

The selected set of SARs is appropriate due to the intended enterprise operating environment and customer base that this product is intended for. EAL4 provides evaluators with access to the implementation details for the TOE and enables deep analysis to identify potential vulnerabilities and exposures which is relevant and expected of an enterprise-grade software product.

EAL4 provides the right balance with understanding and documenting the modular structure of the TOE and the implementation detail, and providing sufficient assurance through independent functional and penetration testing. The following table highlights the assurance requirements of the EAL4 assurance package. The following sections provide specific developer and evaluator notes for applying the identified assurance requirements to web applications.

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 threatens 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. As such, The TOE protecting the CSP-SCD as most valuable asset shall be shown to be highly resistant to penetration attacks. Therefore AVA_VAN.5 was chosen as this demonstrates that the TOE is resistant to attackers with a high attack potential.

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

7 TOE summary specification

7.1 Overview

This chapter provides the TOE summary specification, a high-level description of how the TOE implements the claimed security functional requirements. The TOE implements the following security functions that suitable address the claimed set of requirements:

a) **Secure key management.** Generation and management of digital cryptographic keys for use within the TOE and on behalf of client applications.

b) **Cryptographic operations.** Cryptographic services and the algorithms are provided by the TOE for client applications and internal use.

c) **Secure storage.** Securely protection of keys and other sensitive values that may be stored within the TOE.

d) **User authentication.** The mechanisms used to ensure that only trusted individuals are permitted to access the functions of the TOE.

e) **Security management.** Role-based access to security relevant management functions and data.

f) **Access control.** The mechanisms used to control access to the various security management functions and sensitive data stored in the TOE.

g) **Audit.** Recording of all security relevant events.

h) **Self-test.** A suite of internal tests of the core cryptographic functions and algorithms of the TOE.

i) **Tamper protection.** Mechanisms designed to ensure that the TOE is physically protected against any attempts to directly access the components that comprise the TOE.

7.2 Secure key management

Keyper manages the following three categories of cryptographic keys:

- **Authorisation Key.** The Adapter Authorisation Key (AAK) protects the TOE from unauthorised access by providing the means to authenticate *security officer* and *operator* smart cards. There is only one AAK for each device, which is always generated in the TOE

during the initialisation phase. Alternatively, it may be imported from a smartcard during initialisation.

- **Storage Keys.** Keys generated as a result of a key generation request from an application or keys imported or exported from smart card are called *application keys*. Application keys are protected by a master key called an internal storage master key (ISMK) when stored internally. When exported to be stored externally, they are protected by the currently selected storage master key (SMK). When the SMK is destroyed (during tamper) the ISMK is also destroyed.

- **Application Keys.** These keys are generated by a request for cryptographic services from an external system or application. The TOE can store and manage multiple Application Keys. When a key is generated the external application provides the TOE with the required algorithm, key length, a key label and key usage policy. The generated application key and some data associated with it (key label, usage policy, algorithm, key length) are protected under the ISMK and stored within the HSM. A reference to the key is returned to the external application. This consists of the id of the ISMK and the same associated data that is held with the key (key label, usage policy, algorithm, key length).

Keyper also makes use of a number of system keys, however these cannot be managed.

Keyper provides a range of key management functions that are available to authenticated users (*security officer* and *operator*), which can be accessed via the front panel interface (EL_HUIM). Permitted roles are capable of generating system keys, such as the AAK and SMK (FCS_CKM.1), securely destroying (FCS_CKM.4) and importing and exporting keys to and from smart cards (FCS_CKM.2).

Backup and restore of all keys (AAK, SMK and application) must be authorised by two Security Officers but the mechanism used to produce the backup data is different for each key type.

- AAK – The AAK is split over n components (where n is between 2 and 9 inclusive), all of which are required to import the AAK. One component is written to each smartcard. n-1 components are generated randomly and the final component is calculated such that when all components are xor-ed together the key is reformed.

- SMK – The SMK is split across at least 4 (max. 9) components of which at least 2 (max. 9) are required to import the SMK. One component is written to each smartcard. An “m of n” algorithm is used to calculate the components on export and to reform the SMK on import.

- Application keys – Inside the HSM the application keys are wrapped with the ISMK. When exported through the front panel the keys are decrypted with the ISMK then encrypted with the SMK before being written to the backup media. The reverse happens when they are imported. Anyone with the key reference can export an application key from the TOE (via

the network interface) encrypted under a transport key if and only if the HSM settings and the application key's policy allow it..

All keys (AAK, SMK and application) are destroyed when the HSM is set back to initialised state (which is authorised by the Security Officer – requiring authentication with two users). All keys (AAK, SMK and application) are destroyed when an external or fatal tamper is detected. An external tamper can be done by anyone with physical access to the HSM. In addition, application keys can be destroyed by anyone with the key reference via the network interface or through the front panel (when authorised by the Security Officer – authenticated by two users).

These controls ensure that the confidentiality of the application keys is managed throughout their lifecycle, including when being exported via methods such as smartcards (FPT_ITC.1 and FPT_ITI.1).When the AAK and SMK are exported a key checksum and component checksum are written with each component. The checksum for each component is verified as it is imported and the whole key checksum is verified when the key is reformed. The checksum is the first 4 bytes of the DES ECB encryption of an 8 byte buffer containing all 0x00 using the key/component as the encryption key. Exported application keys are encrypted using the TOE's internal SMK, therefore they may only be imported to a Keyper that contains the same SMK.

As demonstrated, the ***secure key management*** function implements the following SFRs: ***FCS_CKM.2, FCS_CKM.4, FPT_ITC.1, FPT_ITI.1, and FDP_SDI.2.***

7.3 Cryptographic operations

The TOE provides cryptographic data signing/verification, key derivation encryption/decryption, MAC and hashing services to remote uses over TCP/IP. Remote users may also generate and store keys (known as Application Keys) within the TOE. Cryptographic services may be used in conjunction with these keys.

These services are accessed by making specific requests from a network location. This network location must be configured in the TOE; the TOE will not accept requests from any other IP address. The manufacturer provides a "driver" which, when installed on the remote machine, will allow remote users to request services from the TOE using well-known cryptographic APIs: PKCS#11 or the Microsoft CAPI (RSA Full and SChannel). This driver is outside of the scope of this evaluation.

The TOE offers the following cryptographic services to remote users:

a) Generate or import keys (DSA, RSA, DES, TDES or AES) (FCS_CKM.1, FCS_CKM.2)

b) Delete keys (FCS_CKM.4)

c) Signature and signature verification (RSA or DSA) (FCS_COP.1a)

d) MACing and MAC verification (AES, DES or TDES) (FCS_COP.1b)

e) Encryption/decryption (AES, DES or TDES) (FCS_COP.1b)

f) Extract (encrypted) keys (FCS_CKM.2)

g) Hashing (SHA or MD5) (FCS_COP.1c)

h) Random number generation (FCS_RND.1)

i) Storage and retrieval of objects

These services are available to anonymous users (no identification or authentication required), provided that the TOE is in Operational state, has been set *online* by an Operator (or *auto-online* is configured), and the requests originate from the configured IP address.

As demonstrated, the ***cryptographic operations*** function implements the following SFRs: ***FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1a, FCS_COP.1b, FCS_COP.1c and FCS_RND.1.***

7.4 Secure storage

The TOE provides a multilayered secure storage solution to hold cryptographic keys and other sensitive values and objects.

The most sensitive cryptographic values (including the IMK, ISMK & SMK) are kept within the Secure Key Store, a dedicated hardware module. The IMK is generated inside the HSM, before it leaves the manufacturer; if it is erased, the TOE must be returned to the manufacturer for recovery.

All application keys and sensitive cryptographic values are held in a portion of main memory known as the Black Store, which is encrypted with the internal storage master key (ISMK). All system keys (including the AAK) are also encrypted with the internal master key (IMK). Keys which are decrypted for use are held within a separate memory location called the Red Store. No plain-text private or secret keys are permitted to pass outside the boundary of the TOE except in component form and then only if the controlling entity (the TOE) is authorised to make the transfer.

Keyper protects each of the key stores from access through unintended means (such as direct physical access) by implementing a series of tamper detection mechanisms. Tamper events may trigger either a *positive* tamper condition or *operational* tamper condition (depending on the event). An *external* tamper condition (with the effects of either a *positive* or *operational* tamper) may also be triggered manually. When a tamper condition is activated, Keyper positively destroys all sensitive data and securely wipes all keys (FDP_RIP.1). The IMK and other primary sensitive values are only wiped during a *positive* tamper; therefore it is only possible to manually recover from an *operational* tamper. See section 7.10 (Tamper protection) for further details on Keyper's tamper mechanisms.

Keyper's advanced power control module also provides protection against side channel attacks during operation (FDP_IFC.1a, FDP_IFF.4a, FDP_IFC.1b, FDP_IFF.4b).

As demonstrated, the ***secure key storage*** function implements the following SFRs: ***FDP_RIP.1, FDP_IFC.1a, FDP_IFF.4a, FDP_IFC.1b, FDP_IFF.4b.***

7.5 User authentication

The TOE has no internal concept of individual users; although authentication of TOE users is required in order to access restricted functions. Access control in the TOE is not based upon user security attributes, but upon two discrete, statically defined roles: *security officer* and *operator*. TOE functionality that is not assigned to either of these roles is accessible to anonymous (unidentified & unauthenticated) users (FIA_UAU.1, FIA_UID.1).

In order to access controlled functionality, TOE users are authenticated to one of these two roles through the use of smart cards. Sets of smart cards may be issued for each role. The number of *security officer* or *operator* smartcards that are provisioned is configurable with a minimum of two (M) of four (N) smartcards to a maximum of nine (M) of nine (N) smartcards.

Each smartcards contain information that links it to a set, that is, each set has a unique tag. Each smartcard has a unique serial number. The serial numbers are not related to the human users inside the TOE but this can be done outside the TOE. When the *security officer* and *operator* cards are issued or used the serial number of the smart card is noted in the audit log.

Each smart card has an associated personal identification number (PIN), which is must be input as part of the authentication process. When provisioning smart cards, the TOE configures the smart card's internal usage policy, which determines the length of the PIN, and the number of incorrect PIN input attempts that are tolerated (FIA_SOS.1). Once provisioned, this policy is enforced by the smart card, not by the TOE.

Each *security officer* or *operator* smart card set is directly related to a single key: the Authorisation Key (AAK). This key is stored in the TOE, and may be generated or imported during initialisation of the TOE. The information on each smart card is protected by a key derived from the AAK, known as an Authorisation Sub Key (ASK). Each smart card stores its own ASK in a write-protected area. Multiple ASKs (stored in multiple cards) must be presented in order to authenticate against the AAK. As an AAK may be imported, smart card sets may be used for authentication to multiple instances of the TOE, provided each contains the same AAK (FIA_UID.1).

As demonstrated, the ***user authentication*** function implements the following SFRs: ***FIA_SOS.1***, ***FIA_UAU.1*** and ***FIA_UID.1***.

7.6 Security management

The TOE is delivered in "controlled" Initialised State. In this state an Authorisation Key (AAK) can be imported and a set of *security officer* smart cards issued. If an AAK is not imported, one is automatically generated when the *security officer* smart cards are issued. If the Keyper is powered

down when in Initialised State, any stored AAK will be destroyed, rendering any smart cards that have been generated with it useless. When in Initialised State, the TOE cannot accept any requests for cryptographic services. Likewise, SMK or Application keys cannot be imported. In order for the Keyper to become usable (i.e. provide core cryptographic functions and key services), it must be set to Operational State, which requires authentication with a (sub)set of *security officer* smart cards. Note that key export rules and the SMK algorithm type may only be specified in Initialised State.

Once the controlled initialisation is complete and the TOE is transitioned to Operational State, the *security officer* may be authenticated using a (sub)set of *security officer* smart cards and may access restricted *security officer* management functions. The *security officer* may access the following management functions (FMT_SMF.1, FMT_SMR.1):

a) HSM Management

a. Enable auto-online

b. Change the RTC

c. Import & export configuration

d. Modify or view the network IP address, netmask and port number

e. Issue *operator* smart cards

f. Clear smart cards

g. View smart card properties

h. View the FIPS mode, software version, or output the current status

i. Enable or disable selected API operations

j. Enable algorithm suites

k. Set FIPS mode

l. Go to Initialised State

b) Key Management

a. Generate the SMK (FMT_MTD.1b)

b. restore or backup the SMK using smart cards

c. Erase selected Application keys

d. Backup all/selected, or import Application Keys using smart cards

e. Backup the AAK to smart cards

f. Output key information, key summary or key details

The *security officer* may wipe all TSF data, including the AAK, SMK and Application Keys, by returning the TOE to Initialised State (FMT_MTD.1a).

The *operator* is able to enable or disable *online* mode, which must be enabled for the TOE to accept and respond to requests for cryptographic services (FMT_SMF.1 and FMT_SMR.1).

A card holder may make use of the TOE to change a PIN code.

If the TOE's tamper detection mechanisms are activated the TOE could transition to one of two tamper states. Recovery from an Operational tamper is possible if the SMK and Application Keys have been backed up. The TOE will output a message indicating the type of tamper that occurred, and a prompt to clear the tamper status. A *security officer* may then import the backed up SMK and Application Keys. Self-recovery from a Positive tamper event is not possible; the TOE must be returned to the manufacturer for recovery.

As demonstrated, the ***secure management*** function implements the following SFRs: ***FMT_MTD.1a***, ***FMT_MTD.1b***, ***FMT_SMF.1***, and ***FMT_SMR.1.***

7.7 Access control

The TOE implements role based access control mechanisms based on to ensure that users authenticated with their smartcard and PIN are only permitted to perform allocated functions. The TOE implements the following roles that are used to restrict access to the TOE's management tasks (FMT_SMR.1):

a) Operator

b) Security Officer

c) Anonymous user

Each function provided by the TOE may only be performed by a user acting in one of these roles; any one function is not provided to multiple roles (other than smart card PIN change, which may be performed by any smart card issued by the TOE). Only the *security officer* is permitted to perform core security management functions that relate to the management of secure configuration and TSF data. The *operator* is only permitted to enable and disable the availability of the TOE's network services. Section 7.6 (Security management) describes these management functions. Anonymous users (neither identified nor authenticated) may access all other functions provided by the TOE, including requesting cryptographic and key management services from a trusted remote network location.

The *operator* and *security officer* roles are statically defined, therefore there is no security policy or set of security attributes which may affect any change on these roles. A user may assume either of these roles by using the smart card authentication mechanism described in section 7.5 (User authentication). Assumption of a role requires at least two smart card holders to authenticate (the total number depends on the number of smart cards provisioned in a set, and the minimum number specified when the set was created).

As demonstrated, the ***access control*** function implements the following SFRs: **FMT_SMR.1, *FDP_ACC.1, FDP_ACF.1,*** and ***FDP_ETC.1***.

7.8 Audit

All operations performed via the front panel are logged. Key deletion via the network, self-test errors, hardware errors and tamper events are also logged. Event data is captured in the audit log; no sensitive information is included. Each audit log entry includes an event code, a time/date stamp, a number indicating the event type, and a relevant parameter (FAU_GEN.1).

Each TOE management function that may be performed by a user is assigned to a single role, therefore the TOE may associate each audit event that may result from the actions of an identified user with the role which caused that event. However, an audit entry is created when each individual smart card is authenticated containing the serial number of that smart card, therefore it is also possible to associate audit events with the individual smart cards that were used for authentication (FAU_GEN.2).

Keyper includes a real time clock; a battery backed device capable of providing a time and date stamp for the audit records. Access to the device is only possible using trusted methods (FPT_STM.1).

As demonstrated, the ***audit*** function implements the following SFRs: **FAU_GEN.1, FAU_GEN.2 and FPT_STM.1**.

7.9 Self-test

Keyper includes a set of built-in self-tests (BIST). There are three categories of tests: power on self-tests (POST), continuous tests and the firmware load test.

Self-tests are conducted each time Keyper is powered on or reset. Known answer tests are carried out on all FIPS-approved algorithms (i.e. DES, 3DES, SHA-1 and DSA). These tests involve the execution of a series of operations and verification of the outcomes against known answers. A firmware integrity check is also performed, as well as a series of verification tests for all hardware components necessary for correct operation.

In addition to power-up tests, continuous tests are carried out when appropriate and on an ongoing basis. The following continuous tests are implemented by Keyper: random noise source test, random number generator (RNG) conditional test, DSA pair wise consistency test and RSA pair wise consistency test. The pseudo RNG (which is seeded by a hardware noise source) is subjected to a statistical test which is applied to 20 000 bits of output to confirm a sufficiently “flat” random output.

A failure of the POST or continuous self-tests will cause Keyper to inhibit the use of all cryptographic operations (FPT_FLS.1).

Firmware load tests are carried out prior to a firmware upgrade using the HSM Manager interface. Firmware updates are digitally signed using the RSA algorithm and verified by a public key which is built into the module during factory commissioning. The loading of any firmware that is not FIPS 140-2 validated renders the unit a non-FIPS validated unit.

Should a fatal failure occur a hexadecimal number is output via the serial port. If a non-fatal error occurs an audit event is written to the audit log and output via the serial port. The Ready LED will flash when self-tests are in progress or if a self-test has failed. In the event of a self-test failure, the TOE will not be usable and should be returned to the manufacturer for repair.

As demonstrated, the ***self-test*** function implements the following SFRs: ***FPT_TST.1*** and ***FPT_FLS.1***.

7.10 Tamper protection

The TOE includes several tamper protection mechanisms, which meet the tamper protection requirements of FIPS 140-2 level 4. The tamper mechanisms include both internal and external sources of tamper.

The following are considered tamper events and are detected by the TOE (FPT_PHP.2 and FPT_PHP.3):

a) External (mains) voltage outside of specified fatal range,

b) External (mains) voltage outside of specified operational range,

c) Storage temperature outside of normal operating range,

d) Physical breach of the tamper casing (mesh),

e) Operational temperature outside of normal operating range,

f) External tamper switch triggered,

g) Power fluctuation, or

h) Total power failure (both internal battery and mains power).

The internal subsystems of the TOE react to the sources of tamper by notifying the Microcontroller. Once the Microcontroller has been notified of the tamper the TOE will react depending on which tamper event was detected (FPT_PHP.3 and FPT_RCV.1).

If a Positive tamper is triggered, the secure key store (containing the IMK, ISMK and SMK) is actively erased. The Red memory area containing any plain text keys is also actively erased. This behaviour is in accordance with the requirements established in FIPS PUB 140-2 Level 4 (Ref. [16]). When the HSM is next powered up, the Black Store (holding wrapped System Keys and wrapped Application Keys) (which has already been rendered useless by the destruction of the IMK and SMKs) and the application software and download software is positively destroyed. The unit cannot be recovered on site, and must be returned to the manufacturer.

If an Operational tamper is triggered (for example, temperature or power out of range), the Red memory area containing any plaintext keys and the ISMK and SMK (but not the IMK) in the secure key store are actively erased. This renders the Application keys stored in the Black Store useless, as the ISMK protecting it is overwritten. The HSM can be recovered on site by re-importing the SMK and restoring the application keys from smart card. A new ISMK will then be generated automatically.

An External tamper is triggered by a pin-hole switch on the rear panel of the HSM unit. If an External tamper is triggered, the behaviour of the HSM depends on whether or not it has power. If the unit is not currently powered the effects of an Operational tamper are triggered (as long as the external tamper trigger has gone prior to power being applied to the HSM). If the unit is either powered up or currently powered then the effects of a Positive tamper occur, and the unit will need to be returned to the manufacturer for recovery.

As demonstrated, the ***tamper protection*** function implements the following SFRs: **FPT_RCV.1**, **FPT_PHP.2** and **FPT_PHP.3**.

Annex A – Extended components definition (ASE_ECD)

A.1 FCS_RND Generation of random numbers

Justification

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

Family behaviour

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

Component levelling

This family contains only one component.

Management: FCS_RND.1

There are no management activities foreseen.

Audit: FCS_RND.1

There are no actions defined to be auditable.

FCS_RND.1 Quality metrics for random numbers

Hierarchical to: No other components

Dependencies: FPT_TST.1 TSF testing

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*].**