THE
DATA
PROTECTION
COMPANY

LUNA® PCI-E Cryptographic Module

SECURITY TARGET

CR-3524 – REV 18, MAY 30TH 2017.

USED AS A STANDALONE DEVICE OR AS AN EMBEDDED DEVICE IN LUNA® SA

EVALUATION AS ACCORDING TO COMMON CRITERIA EAL4+

CONTENTS

GLOSSARY OF TERMS

Audit data

Data recorded related to security relevant activities (i.e. activities controlled by the TSF). Such audit records can be examined to determine which security relevant activities took place and whom (which user) is responsible for them.

Authentication interface/port

Data interface respective port used for input of authentication data.

Authentication data

General term for data used for authentication of data or of the identity of an entity (e.g., a key or a password).

Authorized user

An individual or remote IT product for which a user account exists in the TOE.

Black data

Cryptographically protected user data representing user information. If this information needs protection in confidentiality the data shall be encrypted. If this information needs protection in integrity a cryptographic MAC or digital signature shall be associated with this data to detect modification.

Bypass

Bypass means that cryptographic processing is not used where it usually would be applied.

Compromise

Unauthorized disclosure, modification, substitution, or use of sensitive data (including plaintext cryptographic keys and other CSPs).

Confidentiality

Property that sensitive information is not disclosed to unauthorized individuals, entities, or processes.

Control input interface/port

Interface respective port intended for all input commands, signals and control data (including function calls and manual controls such as switches, buttons, and keyboards) used to control the operation of a cryptographic module shall enter via the “control input” interface.

Critical security parameter (CSP)

Security-related information (e.g., secret and private cryptographic keys, and TSF data like authentication data) whose disclosure or modification can compromise the security of a cryptographic module.

Critical TSF

TSF that, upon failure, could lead to (i) the disclosure of secret keys, private keys, or CSPs or (ii) modification of public keys. Examples of critical functionality include but are not limited to random number generation, operation of the cryptographic algorithm, and cryptographic bypass.

Crypto Officer

Authorized user who has been granted the authority to perform cryptographic initialization and management functions (including key management) on cryptographically unprotected data in the red area of the IT system. These users are expected to use this authority only in the manner prescribed by the guidance given to them.

Crypto User

Individual or process (subject) acting on behalf of an individual that accesses a cryptographic module in order to obtain cryptographic services.

Cryptographic Algorithm

Well-defined computational procedure that takes variable inputs that usually includes a cryptographic key and produces an output, e.g. encryption, decryption, a private or a public operation in a dynamic authentication, signature creation, signature verification, generation of hash value.

Cryptographic Boundary

Explicitly defined continuous perimeter that established the physical bounds of a cryptographic module and contains all the hardware, software, and/or firmware components of a cryptographic module.

Cryptographic Functions

TSF implementing cryptographic algorithms and/or protocols for:

- Encryption and decryption,
- Signature creation or verification,
- Calculation of Message Authentication Code,
- Entity authentication,
- Key Management.

Cryptographic Key

Parameter used in conjunction with a cryptographic algorithm that determines:

- the transformation of plaintext data into ciphertext data,
- the transformation of ciphertext data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data,
- a Message Authentication Code computed from data,
- a proof of the knowledge of a secret,
- a verification of the knowledge of a secret or
- an exchange agreement of a shared secret.

Cryptographic key component (key component)

Parameter used in conjunction with other key components in an Endorsed security function to form a plaintext cryptographic key by a secret sharing algorithm (e.g. the cryptographic plaintext key is the xor-sum of two key components).

Cryptographic module

Set of hardware, software, and/or firmware that implements Endorsed security functions (including cryptographic algorithms and key generation) contained within the cryptographic boundary.

Cryptographic protocol

Cryptographic algorithm including interaction with an external entity (e.g. key exchange).

Data input interface/port

Interface respective port intended for all data (except control data entered via the control input interface) that is input to and processed by the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and status information from another entities).

Data output interface/port

Interface respective port intended for all data (except status data output via the status output interface) that is output from the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and control information for another entity).

Data path

Physical or logical route over which data passes; a physical data path may be shared by multiple logical data paths.

Decryption algorithm

Algorithm of decoding a ciphertext into the plaintext using a decryption key. The decryption algorithm reproduces the plaintext which was used to calculate the ciphertext with the corresponding encryption algorithm and the corresponding encryption key.

Digital signature

Result of an asymmetric cryptographic transformation of data which, when properly implemented, provides the services of 1. origin authentication, 2. data integrity and 3. signer non-repudiation.

Encryption algorithm

Algorithm of processing a plaintext into a ciphertext using an encryption key in a way that decoding of the ciphertext into the plaintext without knowledge of the corresponding decryption key is computationally infeasible.

Endorsed mode of operation

For this ST, an operational mode of the cryptographic module that employs only Endorsed security functions (e.g. installation, startup, normal operation, maintenance; not to be confused with a specific mode of an Endorsed security function, e.g., AES CBC mode).

Endorsed security function

For this ST, a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is Endorsed by the Bundesamt für Sicherheit in der Informationstechnik. (First hints about Endorsed security functions can be found in BSI TR-02101 [6], but the final decision whether a security function is deemed Endorsed is made when the contents of a Security Target is agreed.)

End-user

Individual or process (subject) acting on behalf of an individual that accesses a cryptographic module in order to obtain cryptographic services.

Error detection code (EDC)

Code computed from data and comprised of redundant bits of information designed to detect, but not correct, unintentional changes in the data.

Error mode

Mode of operation when the cryptographic module has encountered an error condition as defined in FPT_FLS.1 (term is used for description of the Mode transition SFP).

Error state

State related to the Error mode in the finite state model.

Finite state model

In this ST the term finite state model denotes to a finite state machine model, which includes a description of all states of the module, of all transitions between these states (including initial state, destination state, input signal causing the transition, and output signal caused by the transition), and corresponding state transition diagram(s). The term semiformal finite state model is used when the finite state model is using semiformal notation for all of its content.

Firmware

Programs and data components of a cryptographic module that are stored in hardware (e.g., ROM, PROM, EPROM, EEPROM or FLASH) and cannot be dynamically written or modified during execution.

Hardware

Physical equipment used to process programs and data.

Hash-based message authentication code (HMAC)

Message authentication code that utilizes a keyed hash.

Initialization vector (IV)

Vector used in defining the starting point of an encryption process within a cryptographic algorithm.

Input data

Information that is entered into a cryptographic module for the purposes of transformation or computation using an Endorsed security function.

Integrity

Property that sensitive data has not been modified or deleted in an unauthorised and undetected manner.

Internal secrets

Confidential data inside the cryptographic boundary not intended for export (e.g. secret or private plaintext keys, reference authentication data).

IT system

an IT system using the TOE to protect user data during transmission over or storage on media to which unauthorised users have access to.

Key establishment

Process by which cryptographic keys are securely distributed among cryptographic modules using manual transport methods (e.g., key loaders), automated methods (e.g., key transport and/or key agreement protocols), or a combination of automated and manual methods (consists of key transport plus key agreement).

Key interface/port

Data interface respective port used for the input and output of plaintext cryptographic key components and CSPs.

Key loader

Self-contained unit that is capable of storing at least one plaintext or encrypted cryptographic key or key component that can be transferred, upon request, into a cryptographic module.

Key management

Activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.

Key Management mode

Mode for handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.

Key transport

Secure transport of cryptographic keys from one cryptographic module to another module.

Key usage type
Type of cryptographic algorithm a key can be used for (e.g. AES encryption, CMAC calculation, RSA signature creation).

Keying material
Any confidential data directly related to keys or to generation or establishment of keys, i.e. key value itself, key components, intermediate values from key generation (e.g., random numbers, prime numbers) and intermediate values from key agreement/derivation (e.g., shared secret).

Maintenance mode
Mode of operation for maintaining and servicing a cryptographic module, including physical and logical maintenance testing.

Maintenance state
State related to the Maintenance mode in the Finite state model.

Manual key entry
Entry of cryptographic keys into a cryptographic module, using devices such as a keyboard.

Manual key transport
Non-electronic means of transporting cryptographic keys.

Message authentication with appendix
Digital signature scheme which requires the message as input to the verification algorithm. The signature is attached to the message.

Message authentication with message recovery
Digital signature scheme with message recovery is a digital signature scheme for which a priori knowledge of the message is not required for the verification algorithm.

Microcode
Elementary processor instructions that correspond to an executable program instruction.

Operating conditions
Conditions including but not limited to voltage of external power supply and ambient temperature. (Any environmental condition being accidental or induced outside of the normal range intended for the TOE may affect the correct operation or compromise of confidential information.)

Operational CSP
CSP used for protection of the confidentiality or integrity of data by cryptographic operation.

Output data
Data containing information that is produced from a cryptographic module.

Password
String of characters (letters, numbers, and other symbols) used to authenticate an identity or to verify access authorization.

Personal identification number (PIN)
Alphanumeric code or password used to authenticate an identity.

Physical protection
Safeguarding of a cryptographic module, cryptographic keys, or CSPs using physical means.

Plaintext key
Unencrypted cryptographic key.

Port

Physical input or output interface of a cryptographic module that provides access to the module for physical signals, represented by logical information flows. Physically separated ports do not share the same physical pin or wire. In the CC terminology a port is a physical external interface of the TOE (direct or indirect interface to the TSF or interface to the non-TSF portion of the TOE, cf. CEM paragraph 529 for details).

Power interface/port

Interface respective port providing all external electrical power supply.

Power on/off modes

Modes for primary, secondary, or backup power. These states may distinguish between power sources being applied to a cryptographic module.

Private key

Cryptographic key, used with a public key cryptographic algorithm, that is uniquely associated with an entity and is not made public.

Protection Profile

Implementation-independent set of security requirements for a category of Targets of Evaluation (TOEs) that meet specific consumer needs.

Public key

Cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that may be made public.

Public key (asymmetric) cryptographic algorithm

Cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible.

Public key certificate

Set of data that uniquely identifies an entity, contains the entity's public key, and is digitally signed by a trusted party, thereby binding the public key to the entity.

Random Number Generator

Random Number Generators (RNGs) used for cryptographic applications produce a sequence of zero and one bits that may be combined into sub-sequences or blocks of random numbers. There are three basic classes physical true RNG, non-physical true RNG, and deterministic RNG. A physical true RNG produces output that depends on some physical random source inside the TOE boundary only. A non-deterministic true RNG gets its entropy from sources from outside the TOE boundary (e.g. by system data like RAM data or system time of a PC, output of API functions etc. or human interaction like key strokes, mouse movement etc.). A deterministic RNG consists of an algorithm that produces a sequence of bits from an initial random value (seed).

Reference authentication data

Data known for the claimed identity and used by the TOE to verify the verification authentication data provided by an entity in an authentication attempt to prove their identity.

Red data

Cryptographically unprotected user data representing user information which need protection in confidentiality and/or integrity.

Reset

Action to clear any pending errors or events and to bring a system to normal condition or initial state (e.g. after power-up).

Secret key

Cryptographic key, used with a secret key cryptographic algorithm that is uniquely associated with one or more entities and should not be made public.

Secret key (symmetric) cryptographic algorithm)

Cryptographic algorithm which keys for both encryption and decryption respective MAC calculation and MAC verification are the same or can easily be derived from each other and therefore must be kept secret.

Security Officer

Authorized user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given to them.

Self-test mode

Mode of operation in which the cryptographic module performs initial start-up self-test, self-test at power-up, self-test at the request of the authorised user and may perform other self-tests identified in FPT_TST.2.6.

Self-test state

State related to the Self-test mode in the Finite state model.

Shutdown

Shutdown of the TOE initiated by the user (may not include reset after detection of error or power-off due to loss of power supply).

Signature-creation key

Private key for the creation of digital signatures.

Signature-verification key

Public key for the verification of digital signatures.

Software

Programs and data components, usually stored on erasable media (e.g., disk), that can be dynamically written and modified during execution.

Split knowledge

Process by which a cryptographic key is split into multiple key components, individually sharing no knowledge of the original key, that can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.

Status data/information

Data/information that is output from a cryptographic module for the purposes of indicating certain operational characteristics or modes of the module.

Status output interface/port

Interface respective port intended for all output signals, indicators, and status data (including return codes and physical indicators such as Light Emitting Diodes and displays) used to indicate the status of a cryptographic module.

System software

Special software within the cryptographic boundary (e.g., operating system, compilers or utility programs) designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system, and associated programs, and data.

Target of Evaluation (TOE)	Information technology product or system and associated preparative and operational user guidance documentation that is the subject of an evaluation.
TOE Security Functionality (TSF)	Set of the TOE consisting of all hardware, software, and firmware that must be relied upon for the correct enforcement of the TOE Security Policy.
TOE Security Functionality Interface (TSFI)	Set of interfaces, whether interactive (man-machine interface) or machine (machine-machine interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.
Trusted channel	Means by which a TSF and a remote trusted IT product can communicate with necessary confidence.
Trusted path	Means by which a user and a TSF can communicate with necessary confidence.
TSF data	Data for the operation of the TOE upon which the enforcement of the SFR relies.
Unauthenticated user	Identified user not being authenticated and having rights as identified in the component FIA_UAU.1.
Unauthorized user	User who may obtain access only to system provided public objects if any exist.
Unidentified user	User not being identified and having rights as identified in the component FIA_UID.1.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE (includes both authorized and unauthorized entities).
User data	Data for the user that does not affect the operation of the TSF.
Crypto User mode	In a Crypto User mode the data interfaces are open for encryption/decryption of user data with operational keys but key management functions are blocked.
Verification authentication data	Data provided by an entity in an authentication attempt to prove their identity to the TOE.

LIST OF ACRONYMS AND ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
ANSI	American National Standards Institute
API	Application Programming Interface
BSI	Bundesamt für Sicherheit in der Informationstechnik
CA	Certificate Authority
CC	Common Criteria
CIMS	Certificate Issuing and Management System
CKE	Cryptographic Key Export
Challenge Secret	A variable 7 to 16 character ASCII string used during authentication of the Crypto Officer and Crypto User roles generated at the time of role creation and required for authentication to the TOE
CLI	Command Line Interface
CO	Crypto Officer
COTS	Commercial Off-the-Shelf
CSP	Critical Security Parameter
CU	Crypto User
DES	Data Encryption Standard
DLL	Dynamic Linked Library
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
HSM	Hardware Security Module
IT	Information Technology
MTK	Master Tamper Key
NIST	National Institute of Science and Technology
PED	PIN Entry Device (trusted IT device)
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
PP	Protection Profile
RAM	Random Access Memory
RNG	Random Number Generator (Generation)

RSA	Asymmetric algorithm developed by Rivest, Shamir and Adleman
SAR	Security Assurance Requirements
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirements
SHA	Secure Hash Algorithm
SO	Security Officer
ST	Security Target
TAC	Token Access Control (Luna Terminology).
TDES	Triple DES
TOE	Target of Evaluation
TSA	Time Stamp Authority
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
UAV	User Authorization Vector

1 ST Introduction

1.1 ST Identification

Title:	LUNA® PCI-E Cryptographic Module - Security Target
Version:	Rev 18
ST Reference:	CR-3524
Assurance level:	EAL 4 augmented by ALC_FLR.2 and AVA_VAN.4.

Keywords: Commercial-off-the-shelf (COTS), hardware security module, certification authority, certification service provider, key management, cryptographic services, key generation, key protection, digital certificate management, public-key infrastructure, digital signature, encryption, confidentiality, integrity, networked information systems, baseline information protection.

1.2 TOE Reference

Vendor Name	SafeNet, Inc.
TOE Name	LUNA® PCI-E Cryptographic Module.

1.3 TOE Overview

The Target of Evaluation (TOE) described by this Security Target (ST) is inspired but non-conformant to the Protection Profile – BSI-CC-PP-0078 [Cryptographic Modules, Security Level “Enhanced Basic”], version 2.0.2 dated December 6, 2012, published by the Bundesamt für Sicherheit in der Informationstechnik (BSI).

The TOE described in the Security Target is modified from that described in [Cryptographic Modules, Security Level “Enhanced Basic”], in that it must be deployed in a secure environment and relies on trusted software components outside the scope of the TOE to protect against the misuse of commands by authenticated users and to ensure proof of origin for commands received from remote clients. Attack Potential attributed to attacks for non-physical attacks has been separately increased to ‘moderate’ from ‘enhanced basic’ throughout. A full description of differences is included in Section 2.2.

In real-world deployments the use of trusted software is either met by using the services of the Cryptographic Module when configured as an embedded module inside Luna SA or as a stand-alone card when used with supplied host software (non-TOE).

All TOE Security Functions are met by the Luna® PCI-E cryptographic module when installed in a suitable environment and configured as per the supplied user guidance.

The Luna® PCI-E cryptographic module is a Hardware Security Module (HSM) in the form of a PCI-E card that typically resides within a custom computing or secure communications appliance that is operated in a controlled environment.

The TOE is ideally suited as a Hardware Security Module (HSM) for use in digital identity and data protection applications. The Luna® PCI-E cryptographic module features true hardware key management to maintain the confidentiality and integrity of digital signature and encryption keys. Key material is generated, stored, and used exclusively within the secure confines of the Luna® PCI-E cryptographic module to prevent compromise.

The Luna® PCI-E cryptographic module also provides advanced features like direct HSM-to-HSM backup, split user role administration, and multi-person authentication.

1.3.1 TOE Components

The TOE encompasses the following components:

- Luna® PCI-E Cryptographic Module:
    - Firmware Version: 6.10.9
    - Hardware Versions[1] [2]**:** 808-00015-001, 808-00015-003, 808-000052-001, 808-000053-001.
    - Guidance Documentation: CR-4119, Luna PCI-E Cryptographic Module, Common Criteria User Guidance, Revision 2.0 [24].

1.3.2 TOE Usage and Major Security Features

The TOE provides a physically and logically protected component for the performance of cryptographic functions such as:

- key generation; symmetric (e.g. TDES, AES) keys and asymmetric key pairs (e.g. RSA, ECDSA),
- key storage,
- encryption and decryption using both symmetric and asymmetric cryptography, and
- digital signature generation and verification using RSA and ECDSA key pairs.

The TOE is comprised of processors, read-only and random-access memory, and firmware packages in a tamper-resistant form. Access to TOE services is provided using supplied host software (non-TOE) that interfaces to the Crypto Module using its PCI-E interface. Authentication is provided by interfacing to a trusted PIN entry device, such as the Luna® PED II.

[1] the hardware version listed identifies the Luna® PCI-E hardware alongside the non-reconfigurable firmware loaded during production. Non-reconfigurable firmware includes: (i) the bootstrap for the crypto-module and (ii) the configuration files for any one-time soft programmable discrete components configured during manufacture. It is not possible for the user to identify the version of the bootstrap and one-time programmable elements post manufacture independent of the hardware part number. Any modification to these elements made by Safenet will result in a new part number.

[2] a single HW part number (i.e. 808-XXXXX-XXX or 808-XXXXXX-XXX) identifies the TOE with 4 variants being covered by this Security Target where each unique variant replaces 'XXXXX-XXX' with its unique variant identifier e.g. '00015-002'. Hardware revisions associated with the part numbers reflect non-security relevant changes to the product BOM such as work instructions or other non-security enforcing components related to the construction of the TOE.

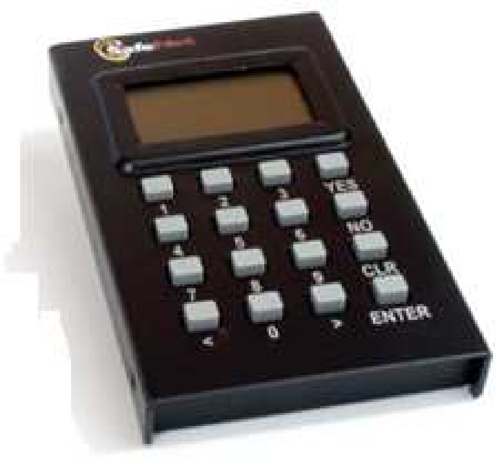

Figure 1 – Example of Trusted PIN Entry Device (Luna® PED II) that can be used with the TOE

Before the TOE can be used to perform any cryptographic or key management functions, it must first be initialised. Initialisation causes the cryptographic module's contents (if any) to be erased and creates the Security Officer (SO) for the cryptographic module. The SO must then set the configurable policies at the cryptographic module level and create at least one partition, with a corresponding user in the Crypto Officer role (creating a user in the Crypto User role is optional), to make the cryptographic module ready for use. The SO may also be required to make policy settings at the partition level to conform to the organization's security requirements.

In operation, the TOE requires users in any of the three roles to be identified and authenticated before they are authorised to perform any cryptographic or key management operations. Authentication is performed using inputs from a trusted PIN entry device. Authentication mechanisms are enforced by the TOE for all Security Officer, Crypto Officer and Crypto User roles.

1.3.3 TOE Type

The TOE is a host-attached hardware cryptographic module or HSM.

1.4 TOE Description

The TOE provides a physically and logically protected component for the performance of cryptographic functions for key generation, key storage, encryption and decryption, key wrapping, secure key transport, key establishment, digital signature generation and verification used by application systems that provide cryptographic support functions such as a Certificate Authority/Certification Service Provider (CA) or Time Stamp Authority (TSA). It includes processors, read-only and random-access memory, and firmware enclosed in a tamper-resistant package.

Figure 2 shows the TOE and Figure 3 shows the TOE when configured as an embedded module inside Luna® SA.

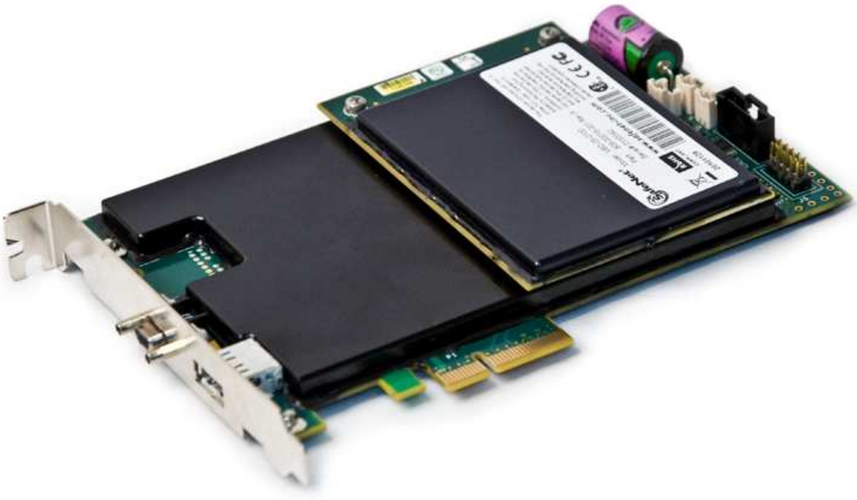

Figure 2 – Luna® PCI-E Cryptographic Module (TOE)

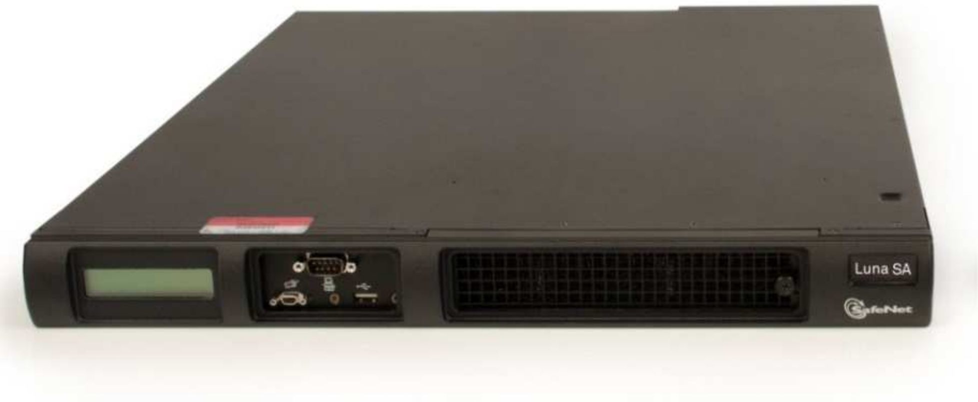

Figure 3 - Luna® SA (TOE Embedded)[3]

Luna SA is the most common host for the Luna PCI-E Cryptographic Module and when acting as the host system, it allows clients to authenticate to the TOE to access cryptographic services and in addition provides the following capabilities outside the scope of the TOE:

- provision of secure channels between the Luna SA appliance and clients;
- management of sessions in multi-partition deployments to strictly enforce the use of allocated session identifiers to authenticated clients;
- collection and management of audit data.

[3] tamper label designs on Luna SA appliances may vary with the label shown being an example.

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

- The Luna® PCI-E cryptographic module – a printed circuit board in PCI-E card format enclosed within tamper-evident metal covers.

TOE Security Functions are contained within the Luna® PCI-E cryptographic module.

The Luna PCI-E cryptographic module includes Linux Kernel version 2.6.28 at its core.

The TOE supports cloning cryptographic objects using the cloning feature which implements cryptographic protocols and mechanisms to protect the confidentiality of user data when transmitted between distinct TOEs or between the TOE and another compatible Trusted IT Product.

1.4.1 TOE Roles

The following authenticated roles are supported by the TOE:

- **Security Officer (SO)** – authorised to install and configure the TOE, set and maintain security policies, and to create and delete users (Crypto Officer and Crypto User roles). The TOE can have only one Security Officer.
- **Crypto Officer (CO)** – authorised to create, use, destroy and backup/restore cryptographic objects. The TOE can have only one Crypto Officer per partition (see below).
- **Crypto User** – authorised to use cryptographic objects (e.g., sign, encrypt/decrypt).

The Crypto Officer and Crypto User can communicate with the TOE for cryptographic operations using PKCS #11 which is part of the IT Environment. The Security Officer uses a separate Command Line Interface (CLI), which is part of the IT Environment, to send command data to perform hsm configuration, to make security policy setting changes and to perform user creation/deletion. The CLI can also be used by the Crypto Officer to initiate the cloning of cryptographic objects between distinct TOEs or between the TOE and a compatible, Trusted IT Product via a Trusted Channel. The network channels used to provide access to the CLI are a requirement of the IT environment.

The TOE allows for the creation of multiple users in the Crypto Officer and Crypto User roles. Each user is created within a cryptographically separated partition in the Luna® PCI-E cryptographic module. Each partition *must* have a user assigned to the Crypto Officer role (a maximum of one user assigned to the Crypto Officer role is permitted). A partition *may* optionally assign a distinct user to the Crypto User role (a maximum of one user assigned to the Crypto User role is permitted).

In Table 1-1 the roles supported by the TOE are compared to the roles defined in the PP [Cryptographic Modules, Security Level “Enhanced Basic”] and PKCS#11.

Function	PP Role	TOE Role	PKCS#11 Role
Initialisation, configuration	Administrator	Security Officer	Security Officer
Key Management	Crypto Officer	Crypto Officer	User
Use	End-User	Crypto User	N/A

Table 1-1 – Protection Profile to TOE Role Comparison

1.4.2 Cryptographic Services

The TOE provides the full range of cryptographic and key management functions. The major functions supported by the TOE are listed below.

- Random Number Generation;
- Asymmetric (Public/Private) Key Pair Generation;
- Symmetric (Secret) Key Generation;
- Secure Key Material Storage and Access;
- Compute Digital Signatures and Verify Digital Signatures;
- Data Encryption / Decryption;
- Secret and Private Key Unwrapping (Import);
- Secret and Private Key Wrapping[4] (Export);
- Key Cloning between HSMs (Backup);
- Key Derivation;
- Message Authentication Code generation.

1.4.3 Non-cryptographic Security Services

The TOE provides the following security services to support the protection of key material and cryptographic services:

- User authentication;
- Access control for security administration functions;
- Access control for the creation and destruction of keys;
- Access control for usage of keys with cryptographic functions;
- Security audit;
- Self-test of the TOE.

1.4.4 Authentication

User authentication data is provided by means of a separate port and data path on the TOE. Trusted PIN Entry Devices, such as the a Luna® PIN Entry Device II (Luna® PED II) can be connected directly to this

[4] private key export is only available when CKE configuration option is enabled and the SO enables the 'allow private key export' policy setting. CKE configuration can by enabled by loading a signed Configuration User File (CUF) that enables CKE (in preference to Cloning).

port. The trusted PIN entry device and software required to operate the trusted PIN entry device are the responsibility of the IT Environment.

1.4.5 Configurable Policy Settings

The Luna® firmware is designed with the flexibility needed to support a number of product variants. The main method used to control the behaviour of different products is a fixed set of “capabilities” set at the factory by loading a signed Configuration Update File (CUF). The settings that are possible to make for the TOE configuration are shown in sections 7.2.1 and 7.2.2. For each of the capabilities, a corresponding policy element exists. The TOE provides security management functions by giving the SO the ability to establish the policy that will govern the cryptographic module’s operation, according to the requirements of the customer organization, by enabling/disabling or refining the corresponding policy elements to equate to or to be more restrictive than the pre-assigned capabilities.

Policy set elements can only refine capability set elements to more restrictive values. Specifically, if a capability is set to allow, the corresponding policy element may be set to either enable or disable. However, if a capability is set to disallow, the corresponding policy element is set to disabled and is not SO-configurable. Thus, an SO cannot use policy configuration to lift a restriction set in a capability definition (i.e. using a CUF).

There are also several elements of the cryptographic module’s behaviour that are truly fixed for all product variants and, therefore, are never subject to configuration by the SO. These fixed elements are the following:

- Non-sensitive secret keys are not allowed;
- Non-sensitive private keys are not allowed;
- Non-private (Public) secret keys are not allowed;
- Non-private (Public) private keys are not allowed;
- Creation of secret keys and private keys through the PKCS #11 create object interface is not allowed. That is, the API cannot be used to create keys by passing in known plaintext values.

The TOE is supplied with a fixed set of capabilities that are selected by the customer on purchase but can be reconfigured once deployed by loading a new CUF which can be supplied by the HSM vendor and is tailored for an individual HSM.

1.4.6 Cryptographic Module Capabilities

Refer to section 7.2.1 for a list of capabilities supported at the cryptographic module level.

1.4.7 Partition Capabilities

Refer to section 7.2.2 for a list of capabilities supported at the partition level.

1.4.8 Backup and Restoration

In order to support the backup and transparent recovery of the cryptographic keys and supporting data stored within the Luna® PCI-E cryptographic module, the cloning capability can be employed. Each Luna® PCI-E cryptographic module may have its cryptographic objects backed up using the Luna® Key Cloning protocol to either a second Luna® PCI-E or a Luna® HSM Backup device. Conversely, the cryptographic objects stored on a backup module may be restored to a properly initialised Luna® PCI-E cryptographic module when bringing it into service.

1.4.9 Guidance Documentation

Guidance on the preparation and operation of the TOE in a manner compliant with common criteria and the requirements in this Security Target is provided in CR-4119, Luna PCI-E Cryptographic Module, Common Criteria User Guidance [24].

1.4.10 Environment

The TOE is used as the cryptographic module for a customer application that typically interfaces with the cryptographic module using supplied (non-TOE) software.

For all deployments, the end user is responsible for meeting a number of obligations on both the IT and physical environments originating for the ‘Security Objectives for the Operational Environment’ section of this Security Target (Section 4.2) and outlined in greater depth in the Common Criteria User Guidance document [24].

2 CONFORMANCE CLAIMS

2.1 CC Conformance Claim

Version: Common Criteria Version 3.1 R4 Part 1, Part 2, and Part 3

ST conformance:

CC Part 2- extended. The following non-Part 2 Security Functional Requirements are included to meet specific requirements of the TOE:

- FCS_RNG.1 (Random Number Generation),
- FPT_EMS.1 (TOE emanation),
- FPT_TST.2 (Self-Testing).

CC Part 3 conformant to EAL 4 augmented. The EAL 4 package has been augmented by the addition of the following Part 3 requirements:

- ALC_FLR.2 (Flaw Reporting Procedures),
- AVA_VAN.4 (Methodical vulnerability analysis).

2.2 Protection Profile Conformance Claim

This ST does not conform to any published Protection Profile but is written to be consistent with the Security Problem Definition, Security Objectives and Security Functional Requirements of Protection Profile [Cryptographic Modules, Security Level “Enhanced Basic”] with the following amendments:

- A.Env - Protected operating environment is added to protect against certain classes of physical attacks on the TOE once deployed.
- A.Client_Management – this is added to ensure the environment takes responsibility for the management of client commands to ensure only clients authorised to use certain identities (issued during stateful interaction) attempt to access services using the identity.
- OE.Env and OE.Client_Management – have been added as objectives to embody the principals introduced in the corresponding assumptions above.
- OSP.Resist_EnhBasic – has been renamed OSP.Resist_Moderate and updated to increase the attack potential for attacks from ‘enhanced basic’ to ‘moderate’.
- O.Physical_Protect, FPT_PHP.1 and FPT_PHP.3 have been removed as this objective (and corresponding SFR) are now covered by OE.Env.
- O.Key_Import and O.Key_Export have been updated to remove a requirement for integrity protection of keys during import and export which is now covered by OE.Client_Management.
- O.Prevent_Inf_Leakage has been amended to only apply to attacks that can be performed from outside the secure environment provided by OE.Env.
- Addition of AVA_VAN.4 instead of AVA_VAN.3 as an additional augmentation to EAL4 which leads to the modification of a number[5] of Threats, Objectives and Operating Security Policy to

[5] The following items have been modified to reference ‘Moderate’ rather than ‘Enhanced Basic’: T.Compro_CSP, T.Modif_CSP, T.Abuse_Func, T.Inf_Leakage, T.Malfunction, T.Physical_Tamper, T.Masquerade, OSP.Resist_Moderate, O.Prevent_Inf_Leakage, OE.Assurance.

reference ‘Moderate’ Attack Potential rather than ‘Enhanced Basic’. This brings the ST in line with higher attack potential in [Cryptographic Modules, Security Level “Standard”].

- O.Red-Black Separation has been removed as a redundant objective in light of the addition of A.Client_Management and the SFR supported by it being independently justified by O.Key_Import, O.Key_Export and O.Key_Management.
- FDP_UCT.1 and FDP_UIT.1 have been removed as redundant functions given:
  - Red-Black SFP is undefined in the original PP which makes these functions technically unenforceable.
  - The TOE environment provides confidential and integrity protection to user data and integrity for PKCS #11 commands used to import and export keys through OE.Client_Management.
  - FCS_CKM.2/Private_Key_Import, FCS_CKM.2/ Secret_Symmetric_Key_Import and FCS_CKM.2/ Private_Key_Export and FCS_CKM.2/ Secret_Symmetric_Key_Export already provide additional confidentiality protection for the import and export of secret and private keys.
- Roles throughout the ST have been updated to Luna equivalent names to those used in the PP to tie in with User Documentation - Administrator has become Security Officer and End-User, the Crypto User.
- Reference to the optional Maintenance Role has been dropped from the ST as it isn’t required by the TOE[6].
- PP refinement to FCS_CKM.2 iterations for import and export of keys using the key wrapping has been updated to remove the requirement for integrity protection of keys which is now provided by OE.Client_Management for keys imported using this protocol[7]. This is consistent with changes to O.Key_Import and O,Key_Export covered above.
- Updates have been made to O.Key_Import, O.Key_Export, OE.Personnel and the refinements present in all iterations of FCS.CKM.2 to remove reference to manual or split key entry and export which are not supported by the TOE.
- ‘Key/CSP entry mode’ as used in FDP_ACF.1.4 has been updated to ‘Key Management Mode’ to reflect the wider range of key life-cycle operations[8] permissible in this mode.
- ‘time validity’ has been removed as a mandated TOE enforced attribute for objects resulting in changes from the PP:
  - Section 3.1.3 has been updated to remove ‘time validity’ as a mandated attribute;
  - FDP_ACF.1.1/Key_Man and FDP_ACF.1.2/Oper have been updated to remove ‘time validity’ as an attribute used for access control.

[6] with the exception of FDP_ACF.1/Key_Man which is solely focused on the maintenance role. This SFR has been retained to keep consistency with BSI-CC-PP-0078 but an application note has been added to state the maintenance role and associated keys and CSP are not supported by the TOE.

[7] the requirement for integrity protection is retained as a refinement for FCS_CKM.2/cloning_import and FCS_CKM.2/cloning export that use the ‘Cloning Protocol’ which is used to transfer keys between HSM and is not covered by OE.Client_Management.

[8] Key Management mode includes the range of operations defined under Key Management in the glossary including: ‘*handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction*’.

- FDP_ITC.2.5, FDP_ETC.2.4 and FPT_TDC.1 have been updated to remove ‘time validity’ as a mandated attribute to transfer and to maintain consistency off, during user data import and export.
- FMT_MSA.1/Key_Man_1, FMT_MSA.1/Key_Man_2, FMT_MSA.2 have been updated to remove ‘key validity time period’ as an attribute to control.
- FPT_STM.1 has been removed as it is was only required as a dependency to support enforcement of ‘time validity’ as an object attribute.
- FMT_MSA.1/Key_Man_1 and FMT_MSA.1/Key_Man_2 have been re-written to correct mistakes in the original PP and to align them with the Key Management SFP and roles as defined for the module.

- Pairwise consistency test on generation of asymmetric keys as covered in the refinement to FPT_TST.2 in the PP has been simplified to a ‘sign/verify’ omitting the ‘encrypt/decrypt’ test covered in the PP.

New assumptions and objectives for the environment that have been introduced or SFR that have been added or significantly altered have been marked with a (*) in their title to clearly identify them to the reader.

3 SECURITY PROBLEM DEFINITION

3.1 Introduction

This Security Target (ST) describes the security problem for cryptographic modules, which may provide a wide range of cryptographic security functions depending on the intended protection of the user data. This intended protection of primary assets is addressed by organisational security policies. The TOE protects the user data in confidentiality and integrity. The use of cryptographic methods implies specific threats, which are common for all TOEs as cryptographic modules.

3.1.1 Assets

The cryptographic module is intended to protect, as primary assets, User Data in the form of wrapped secret and private keys as input. User Data is only exported from the TOE as wrapped secret keys when leaving the TOE boundary.

The cryptographic keys need protection as the primary assets they protect and depending on the cryptographic technique they are used for:

- **Secret keys** of symmetric cryptographic algorithms and protocols need protection in confidentiality and integrity.
- **Private keys** of asymmetric cryptographic algorithms and protocols need protection in confidentiality and integrity.
- **Public keys** of asymmetric cryptographic algorithms and protocols need protection in integrity and authenticity.

The cryptographic module also accepts Red Data in the form of:

- **Plaintext data** provided by the IT Environment containing information, which need protection in confidentiality.
- **Original data** containing information, which need protection in integrity or a proof of origin and authenticity to third parties.
- **ICD Command Data,** including client-application identifying data

The use of cryptographic algorithms and protocols requires the protection of the **cryptographic keys** as primary assets.

Where the need of confidentiality of secret and private keys follows directly from the cryptographic technique the integrity protection for these keys prevents indirect attacks (e.g. substitution of an unknown secret key by a known key compromise the subsequent encryption of plaintext data, an undetected modification of a private key may enable attacks against this key).

The CC deals with cryptographic keys as user data and as TSF data depending on their specific use by the TSF. Cryptographic keys are user data in the terminology of CC if they are used to protect cryptographically the confidentiality or integrity of data provided by the IT system, or to transform “cryptographically protected (encrypted) data” into “cryptographically unprotected (decrypted) data” by cryptographic functions. Encryption and decryption keys are examples of such keys. Cryptographic keys are TSF data in the terminology of CC if their information is used by the TSF in making security-relevant decisions. Root public keys are examples of cryptographic keys as TSF because they are used to verify the authenticity of all other public keys of the public key infrastructure, which may be provided by any user. Public keys may be used as reference authentication data for external entities as user of the TOE.

3.1.2 Roles

The section has been updated from [Cryptographic Modules, Security Level "Enhanced Basic"] to map the roles as defined in the PP to those used on Luna PCI-E and covered in Table 1-1. The re-defined roles are used throughout subsequent sections of the document. This ST retains all mandated roles as listed in the PP.

Roles	Description
Security Officer[9]	An authorised user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given to them.
Crypto Officer	An authorised user who has been granted the authority to perform cryptographic initialization and management functions. These users are expected to use this authority only in the manner prescribed by the guidance given to them.
Crypto User[10]	An authorised user assumed to use general security services, including cryptographic operations and other Endorsed security functions.
Unidentified User	A user not being identified.
Unauthenticated User	An identified user not being authenticated.

Table 3-1 – Roles

The term "user" is used to include both authorised and unauthorised users. Authorized users are known to the TOE and their security attributes are maintained by the TOE as prerequisite for their identification and authentication. Unauthorised users are unknown to the TOE. An authenticated authorised user is an authorised user that has been successfully authenticated for one or more of the following roles: Security Officer, Crypto Officer or Crypto User role.

A user in the Crypto User role may be a human user or an IT system communicating with the TOE.

The TOE maintains at least the following security attributes of authorised users:

(1) **User identity** that uniquely identifies the user,

(2) **Role** for which the user is authorised.

and TSF data

(3) Reference Authentication Data for users.

The TOE maintains at least the following security attributes of subjects:

(1) **Identity** of the user bound to this subject,

(2) **Role** for which this user is currently authenticated.

[9] equivalent to the Administrator role in the PP.
[10] equivalent to the End User role in the PP.

3.1.3 Objects

The following objects are defined in [Cryptographic Modules, Security Level “Enhanced Basic”]. The full list of objects has been retained in this Security Target to cover of range of operations performed by the TOE.

Object	Description
Plaintext data	Red-Data encoded in a public known way which will be transformed by an encryption algorithm implemented in the TOE into ciphertext data (i.e. plaintext input data) or which is the result of decryption of the corresponding ciphertext data by the TOE (i.e. plaintext output data).
Ciphertext data	Black-data as result of the application of an encryption algorithm to plaintext data and an encryption key. The knowledge of ciphertext data by an attacker does not compromise the confidential information represented by the corresponding plaintext.
Cryptographic keys	Parameters used in conjunction with a cryptographic algorithm that determines the transformation of plaintext data into ciphertext data, the transformation of ciphertext data into plaintext data, a digital signature computed from data, the verification of a digital signature computed from data, a message authentication code computed from data, a proof of the knowledge of a secret, a verification of the knowledge of a secret or an exchange agreement of a shared secret.
Cryptographic key component	Parameters used in split knowledge procedures for manual key export methods and manual key import methods.
Critical security parameters	Security-related information (e.g. secret and private cryptographic keys, and TSF data like authentication data) whose disclosure or modification can compromise the security of a cryptographic module.
Digital signature	The result of an (asymmetric) signature-creation algorithm applied to the original data using a signature-creation key. The digital signature may contain or be appended to the original data.
Message authentication code (MAC)	The result of a (symmetric) message authentication algorithm applied to the original data using a message authentication key. The MAC will be appended to the original data.

Table 3-2 – Objects

Critical security parameters (CSP) have at least the security attributes:

(1) **Identity of the CSP** that uniquely identifies the CSP,
(2) **CSP usage type** identifying the purpose and methods of use of the CSP,
(3) **CSP access control rules.**

The CSP access control rules may restrict the access for operation like import or export of the key.

Cryptographic keys have at least the security attributes:

(1) **Identity of the key** that uniquely identifies the key,
(2) **Key entity**, i.e. the identity of the entity this key is assigned to,
(3) **Key type**, i.e. secret key, private key, public key,
(4) **Key usage type**, i.e. the cryptographic algorithms a key can be used for

(5) **Key access control rules**.

The security attribute “key usage type” shall identify the cryptographic algorithm the key is intended to be used and may contain information about the rank of this key in a key hierarchy, and other information. The security attribute “Key access control rules” restricts the access for operation like import or export of the key.

Cryptographic key components have at least the security attributes

(1) **Key component** identity that uniquely identifies the key component,
(2) **Key entity**, i.e. the identity of the key the key component belongs to,
(3) **Key entry method**, i.e. the method the key component is used for.

Furthermore cryptographic keys, key components and CSP may be distinguished as

- Operational if they are used to protect user data,
- Maintenance if they are used for maintenance of the TOE by maintenance personnel only[11].

Please take note that data used internally by known answer self-test of the TOE instead of cryptographic keys are seen neither as operational nor as maintenance keys (CSP).

3.1.4 Operations

The following objects are defined in [Cryptographic Modules, Security Level “Enhanced Basic”]. The full list of objects has been retained in this Security Target to cover of range of operations performed by the TOE.

Operation	Description
Decryption	Processes a decryption algorithm to the ciphertext data using the decryption key and returns the corresponding plaintext data.
Encryption	Processes an encryption algorithm to the plaintext data using the encryption key and returns the corresponding ciphertext data.
Export of key	Output of cryptographic keys in protected form.
Export of protected data	Output of user data with or without security attributes to the black area of the IT system protected in confidentiality or integrity or both by cryptographic security functions of the TOE.
Export of unprotected data	Output of user data with or without security attributes to the red area of the IT system cryptographically protected by cryptographic security functions of the TOE.
Import of key	Input of cryptographic keys in protected form.
Import of protected data	Input of user data with or without security attributes from the black area of the IT system where the cryptographic security functions of the TOE support the protected in confidentiality by decryption or in integrity by detection modification or verification of data origin.

[11] The TOE does not support a maintenance role or the concepts of maintenance cryptographic keys, key components and CSP. These concepts are retained in this section for consistency with Protection Profile – BSI-CC-PP-0078 only.

Operation	Description
Import of unprotected data	Input of user data with or without security attributes to the red area of the IT system cryptographically unprotected by cryptographic security functions of the TOE.
MAC calculation	Processes a (symmetric) MAC algorithm to the original data using the secret message authentication key and returns the corresponding Message Authentication Code.
MAC verification	Processes a (symmetric) MAC algorithm to the presented user data and MAC using the secret message authentication key and returns the result of checking whether the user data, the MAC and the key fit together (integrity confirmed) or not (integrity not confirmed).
Signature-creation	Processes a (asymmetric) signature-creation algorithm to the original data using the private signature-creation key of the signatory and returns the corresponding digital signature.
Signature-verification	Processes a (asymmetric) signature-verification algorithm to the signed data and the digital signature using the public key and returns the result of checking whether the original data, the electronic signature and the public key fit together (integrity confirmed) or not (integrity not confirmed).
Use of key	Use of the cryptographic key by a cryptographic algorithm as key parameter.

Table 3-3 – Operations

3.2 Threats

Threat statements are taken directly from [Cryptographic Modules, Security Level “Enhanced Basic”], Section 3.2 with the only modification being the increase of attack potential from ‘Enhanced Basic’ to ‘Moderate’.

3.2.1 T.Compro_CSP – Compromise of confidential CSP

An attacker owing moderate attack potential may compromise confidential CSP like secret keys, private keys or authentication data, which enables attacks against the confidentiality or integrity of user data protected by these CSPs or the TSF using these CSPs as TSF data.

3.2.2 T.Modif_CSP – Modification of integrity-sensitive CSP

An attacker owing moderate attack potential may modify integrity-sensitive CSP like permanent stored public keys and therefore compromise the confidentiality or integrity of user data protected by these CSPs or the TSF using these CSPs as TSF data.

3.2.3 T.Abuse_Func – Abuse of function

An attacker owing moderate attack potential may use TOE functions intended for installation, configuration or maintenance of the TOE which shall not be used for operational cryptographic keys or user data in order (i) to disclose or manipulate operational CSP or user data, or (ii) to enable attacks against the integrity or confidentiality of operational CSP or user data by (ii.a) manipulating (explore, bypass, deactivate or change) security features or functions of the TOE or (ii.b) disclosing or manipulating TSF Data.

3.2.4 T.Inf_Leakage – Information leakage

An attacker owing moderate attack potential may observe and analyse any energy consumed or emitted through the cryptographic boundary (i.e. including the external interfaces) of the TOE to get internal secrets (especially secret or private cryptographic keys) or confidential user data not intended for export. The information leakage may be inherent in the normal operation or caused by the attacker.

3.2.5 T.Malfunction – Malfunction of TSF

An attacker owing moderate attack potential may use a malfunction of the hardware or software, which is accidental or deliberated by applying environmental stress or perturbation, in order to deactivate, modify, or circumvent security functions of the TOE to enable attacks against the integrity or confidentiality of the User data or the CSP.

3.2.6 T.Physical_Tamper – Physical tampering

An attacker owing moderate attack potential may tamper the cryptographic module to get secrets, to modify data on whose integrity the TSF relies, or to corrupt or de-activate the TSF inside the cryptographic boundary to violate the integrity or confidentiality of the User data, the CSP or the TSF data.

3.2.7 T.Masquerade – Masquerade authorized data source or receiver

An attacker owing moderate attack potential may masquerade as an authorized data source or receiver to perform operations that will be attributed to the authorized user or may gain undetected access to cryptographic module causing potential violations of integrity or confidentiality of the User data, the CSP or the TSF data.

3.3 Organisational Security Policies

Organisational Security Policy statements are taken directly from [Cryptographic Modules, Security Level “Enhanced Basic”], Section 3.3. with the only modifications being the increase of attack potential from ‘Enhanced Basic’ to ‘Moderate’, minor rewording of roles to match those used on Luna.

3.3.1 OSP.User_Data_Prot – Protection of user data by cryptographic functions

The cryptographic module will be used to protect the confidentiality and/or integrity of information represented by user data which may be get known or modified by an attacker. The IT system will ensure the availability of the user data and the cryptographic keys outside the cryptographic module.

3.3.2 OSP.Resist_Moderate – Resistance against moderate attack potential (*)

The TOE shall resist attacks all attacks on the module with moderate attack potential.

3.3.3 OSP.I&A – Identification and authentication of users

All users shall be identified and authenticated prior to accessing any controlled resources with the exception of read access to public objects and cryptographic operations with public keys.

3.3.4 OSP.Access – Access control of TOE functions

The TOE shall limit the extent of each user's abilities to use the TOE functions in accordance with the security function policies.

3.3.5 OSP.Roles – Roles

The authorized Security Officer, Crypto Officer and Crypto User shall have separate and distinct roles associated with them.

3.3.6 OSP.Endorsed_Crypto – Endorsed cryptographic functions only

The TOE shall implement Endorsed cryptographic algorithms and Endorsed cryptographic protocols for the protection of the confidentiality and/or the integrity of the user data according to the organizational security policy OSP.User_Data_Prot and for the cryptographic key management according to the organizational security policy OSP.Key_Man. The cryptographic module must not provide any non-Endorsed cryptographic function.

3.3.7 OSP.Key_Man – Cryptographic key management

The CSP, cryptographic keys and cryptographic key components are assigned to cryptographic algorithms and protocols they are intended to be used with and the entities, which are allowed to use them.

3.3.8 OSP.Key_Personnel – Personnel security for cryptographic keys

The cryptographic keys shall be managed in such a way that their integrity and confidentiality cannot be compromised by a single person.

3.4 Assumptions

Existing assumptions from [Cryptographic Modules, Security Level "Enhanced Basic"], Section 3.4 of the referenced PP are retained but have been be expanded to include two new assumptions (A.Env and A.Client_Management) as already covered in Section 2.2.

3.4.1 A.User_Data – Protection of user data by the IT system

The TOE environment uses the TOE for cryptographic protection of user data for transmission over channels or storage in media, which are not protected against access by unauthorised users. The TOE environment provides cryptographically unprotected user data to the TOE and identifies protection in confidentiality and/or integrity to be provided by the TOE.

3.4.2 A.Data_Sep – Separation of cryptographically protected and unprotected data

The TOE environment separates the cryptographically unprotected data from the cryptographically protected user data in the IT system.

3.4.3 A.Key_Generation – Key generation and import to the cryptographic module

Cryptographic keys generated by the IT environment and imported into the TOE are cryptographically strong for the intended key usage and have secure security attributes.

3.4.4 A.Availability – Availability of keys

The TOE environment ensures the availability of cryptographic keys, key components, CSP and keying material.

3.4.5 A.Env – Protected operating environment (*)

The TOE operates in a protected environment that limits physical access to the TOE to authorised individuals and protects against all physical attacks that could lead to compromise of the objectives of the TOE.

3.4.6 A.Client_Management – Correct data supplied by client applications (*)

Any host supporting the TOE shall:

- manage messages exchanged between the TOE and client to ensure only clients authorised to use certain identities (issued during stateful interaction) attempt to access services using the identity.
- ensure proof of origin for messages passed to the host from clients prior to these being passed to the TOE.
- ensure the confidentiality and integrity of User Data in transit between clients and the TOE[12].

[12] This is explicitly noted as not including confidentiality protection of cryptographic keys which are already provided confidentiality protection under O.Key_Import and O.Key_Export.

4 Security Objectives

This section identifies and defines the security objectives for the TOE and its environment. Security objectives reflect the stated intent and counter the identified threats, as well as comply with the identified organisational security policies and assumptions. Section 4.1 and Section 4.2 are taken from [Cryptographic Modules, Security Level “Enhanced Basic”] with the following changes:

- new Security Objectives for the Environment to address A.Env and A.Client_Management have been added.
- Role titles have been updated to tie in with those used by Luna but without a change in definition from those used in the PP.
- O.Key_Import and O.Key_Export have been updated to remove requirement to ensure integrity of keys which is now covered by OE.Client_Management.
- O.Prevent_Inf_Leakage has been updated to only apply to attacks that can be performed from outside the secure environment. This avoids a conflict with OE.Env that already protects the TOE from direct attacks.
- OE.Personnel has been updated to remove reference to manual key or split key component loading which is not supported by the TOE.

4.1 Security Objectives for the TOE

4.1.1 O.Endorsed_Crypto – Endorsed cryptographic functions

The TOE shall provide Endorsed cryptographic functions and Endorsed cryptographic protocols to protect the user data as required by OSP.User_Data_Prot and for key management.

4.1.2 O.I&A – Identification and authentication of users

The TOE shall uniquely identify users and verify the claimed User identity before providing access to any controlled resources with the exception of read access to public objects.

4.1.3 O.Roles – Roles known to TOE

The TOE shall provide at least the Security Officer, the Crypto Officer, and the Crypto User roles.

4.1.4 O.Control_Services – Access control for 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.

4.1.5 O.Control_Keys – Access control for cryptographic keys

The TOE shall restrict the access to the keys, key components and other CSP according to their security attributes. Cryptographic keys intended for the use with Endorsed cryptographic functions must not be used by any non-endorsed functions.

4.1.6 O.Key_Export – Export of cryptographic keys

The TOE shall export cryptographic keys with their security attributes. The TOE shall ensure the confidentiality of secret and private keys exporting them in encrypted form to authorized entities only.

4.1.7 O.Key_Generation – Generation of cryptographic keys by the TOE

The TOE shall generate cryptographic strong keys using Endorsed cryptographic key generation algorithms.

4.1.8 O.Key_Import – Import of cryptographic keys

The TOE shall import keys with security attributes. The TOE shall import secret or private keys in encrypted form.

4.1.9 O.Key_Management – Management of cryptographic keys

The TOE shall securely manage cryptographic keys, cryptographic key components and CSP. The TOE shall associate security attributes of the entity the key is assigned to and of the intended cryptographic use of the key. Assignment of the security attributes to the cryptographic keys, cryptographic key components and CSP shall be either done by explicit action of a Crypto Officer or by default.

4.1.10 O.Key_Destruction – Destruction of cryptographic keys

The TOE shall destruct in a secure way the keys cryptographic key components and other CSP on demand of authorised users or when they will not be used any more that no information about these keys is left in the resources storing or handling these objects before destruction.

4.1.11 O.Check_Operation – Check for correct operation

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

4.1.12 O.Prevent_Inf_Leakage – Prevent leakage of confidential information

The TOE shall prevent information leakage about secret and private keys and confidential TSF data outside the cryptographic boundary and unintended output confidential user information. The TOE shall resist attacks with moderate attack potential, which are based on information leakage and that can be performed from outside the secure environment.

4.2 Security Objectives for the Operational Environment

4.2.1 OE.Assurance – Assurance Security Measures in Development and Manufacturing Environment

The developer and manufacturer ensure that the TOE is designed and fabricated so that it requires a combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through attack. The developer provides necessary evaluation evidence that the TOE fulfils its security objectives and is resistant against attack with moderate attack potential.

4.2.2 OE.Key_Generation – Key generation by IT environment

The IT environment shall ensure the cryptographic strength, the confidentiality and integrity of secret and private keys, the integrity and authenticity of public keys and correct security attributes if they are generated outside the TOE and imported into the TOE.

4.2.3 OE.Red-Black-Sep – Separation of red and black area of the IT system

The TOE environment protects the user data in the red area of the IT system and controls the exchange data between the red and black area of the IT system according to the IT security policy. It provides the red data with their security attributes for cryptographic protection to the TOE and receives red data with their security attributes from the TOE.

4.2.4 OE.Personnel – Personnel security

The Security Officer, Crypto Officer, Crypto User roles, shall be assigned with distinct authorised persons.

4.2.5 OE.Key_Availability – Availability of cryptographic key and key material

The IT environment shall ensure the availability of the user data, cryptographic keys key components, CSP and keying material.

4.2.6 OE.Env – Protected operating environment (*)

The TOE shall operate in a protected environment that limits physical access to the TOE to authorised individuals. The TOE software and hardware environment shall be maintained in a secure state that mitigates the specific risks applicable to the deployment environment.

4.2.7 OE.Client_Management (*)

Any host supporting the TOE shall:

- manage messages exchanged between the TOE and client to ensure only clients authorised to use certain identities (issued during stateful interaction) attempt to access services using the identity.
- ensure proof of origin for messages passed to the host from clients prior to these being passed to the TOE.
- ensure the confidentiality and integrity of User Data in transit between clients and the TOE[13].

4.3 Security Objectives Rationale

The organisational security policy **OSP.User_Data_Prot** "Protection of user data by cryptographic functions" addresses the protection of the confidentiality and/or integrity of information represented by user data of the IT-system to be provided by the cryptographic module and the protection of availability of user data by the IT system. The security objective O.Endorsed_Crypto ensures that TOE provides Endorsed cryptographic functions to protect the user data as required by OSP.User_Data_Prot. The security objective for the IT environment OE.Key_Availability ensures that IT system protects the availability of the user data and the cryptographic keys outside the cryptographic module.

The organisational security policy **OSP.Resist_Moderate** "Resistance against moderate attack potential" requires the TOE to resist attacks with moderate attack potential. This is ensured by the security objective for the development environment OE.Assurance (cf. to last sentence) and OE.Env relating to it being deployed in a protected operating environment. The security objectives O.I&A, O.Prevent_Inf_Leakage address directly the resistance against attacks with moderate attack potential that can be performed from outside the controlled environment.

The organisational security policy **OSP.I&A** "Identification and authentication of users" addresses identification and authentication of all users prior to accessing any controlled resources with the exception of public objects. This is directly ensured by the security objective O.I&A.

The organisational security policy **OSP.Access** "Access control of TOE functions" addresses the limitation of the extent of each user's abilities to use the TOE functions in accordance with the security function policies. The security objective O.Control_Services requires that the TOE shall restrict the access to its services, depending on the user role, to those services explicitly assigned to this role which are provided according to the security objective O.Roles. The security objective O.Control_Keys limits user's ability to use the TOE functions to ensure the cryptographic security as part of the security function policies. The environmental objective OE.Client_Management adds a secondary layer of to provide stateful management of users.

The organisational security policy **OSP.Roles** "Roles" addresses separate and distinct roles for authorised Security Officer, Crypto Officer and Crypto User. The security objective O.Roles requires the TOE to implement them and the security objective OE.Personnel requires the IT environment to use them.

The organisational security policy **OSP.Endorsed_Crypto** "Endorsed cryptographic functions" addresses the implementation of Endorsed cryptographic algorithms and Endorsed cryptographic protocols for the protection of the confidentiality and/or the integrity of the user data according to the organizational

[13] This is explicitly noted as not including confidentiality protection of cryptographic keys which are already provided confidentiality protection under O.Key_Import and O.Key_Export.

security policy OSP.User_Data_Prot and for the key management. This is ensured generally by the security objective O.Endorsed_Crypto.

The security objective **OSP.Key_Man** “Cryptographic key management” requires to manage and use the cryptographic keys as they are assigned to the entities, cryptographic algorithms and protocols. This OSP is implemented generally by the security objectives for the TOE O.Key_Management for secure key management and specifically for critical processes over the key life cycle by the security objectives O.Key_Generation, O.Key_Import, O.Key_Export and O.Key_Destruction. OE.Key_Generation ensures the cryptographic strength, the confidentiality and integrity of secret and private keys, the integrity and authenticity of public keys and correct security attributes if they are generated outside the TOE and imported into the TOE. OE.Client_Management provides integrity protection for keys (and any associated attributes during import and export of wrapped packages).

The organisational security policy **OSP.Key_Personnel** “Personnel security for cryptographic keys” addresses key management in a way that the integrity and confidentiality of key cannot be compromised by a single person. This OSP is implemented generally by the security objectives O.Key_Management and O.Control_Keys for secure key management and use. Furthermore for critical processes, the security objectives O.Key_Import, O.Key_Export and O.Control_Keys enforce secure key import, key export and key usage. O.I&A ensures that the TOE uniquely identifies users and verifies the claimed User identity before providing access. OE.Personnel requires assignment of roles to distinct authorised persons and that for manual key import at least two different authorised persons are assigned to Crypto Officer role. OE.Client_Management ensures stateful management of commands sent between the TOE and clients.

The threat **T.Compro_CSP** “Compromise of CSP” addresses the compromise of confidential CSP which enables attacks against the confidentiality or integrity of user data and TSF data protected by these CSPs. The security objective O.Control_Keys requires the TOE to restrict the access to the keys, key components and CSP according to their security attributes. The security objective O.Key_Management ensures these security attributes are managed securely. The security objective O.Key_Export and O.Key_Import require the protection of secret or private keys in encrypted. The security objective O.Key_Generation requires the TOE and OE.Key_Generation requires the environment to generate cryptographic strong keys. O.Key_Destruction requires the secure destruction on demand of user. The security objective O.Prevent_Inf_Leakage requires the TOE to prevent information leakage about secret and private keys and confidential TSF data that can be intercepted from outside the secure environment provided by OE.Env. Physical tampering attempts aiming at compromising CSPs are countered by the environment as addressed by OE.Env. OE.Client_Management provides protection against any attempts to hijack an authenticated and active session.

The threat **T.Modif_CSP** “Modification of integrity-sensitive CSP” addresses the modification of the integrity-sensitive CSP which enables attacks against the confidentiality or integrity of user data or the TSF protected by these CSPs. The security objective O.Control_Keys requires the TOE to restrict the access to the keys, key components and CSP according to their security attributes. The security objective O.Key_Management ensures these security attributes are managed securely. The security objective O.Check_Operation requires verification the integrity of CSP. OE.Env provides protection against direct attacks on the cryptographic module looking to modify or replace CSP. OE.Client_Management provides protection against the hijack of authenticated sessions that could lead to the deletion of CSP and integrity protection for wrapped keys during import and export as part of messages exchanges between the TOE and client.

The threat **T.Abuse_Func** “Abuse of function” addresses the misuse of TOE functions intended for installation, configuration or maintenance which shall not be used for operational cryptographic keys or user data. This is ensured by the security objective O.Control_Services that restricts the access to TOE services, depending on the user role, to those services explicitly assigned to this role. The security objective O.Roles requires the TOE to provide at least the Security Officer, the Crypto Officer, the Crypto

User roles. The Security Officer, Crypto Officer, Crypto User roles will be assigned to authorised distinct persons according to the security objective for the IT environment OE.Personnel. OE.Client_Management provides protection against the hijack of authenticated sessions.

The threat **T.Inf_Leakage** “Information leakage” describes that an attacker may observe and analyse any energy consumed or emitted through the cryptographic boundary (i.e. including the external interfaces) of the TOE to get internal secrets or confidential user data not intended for export. The protection against this threat is directly required by the security objective O.Prevent_Inf_Leakage and supported by OE.Env to defend against direct attacks on the cryptographic module.

The threat **T.Malfunction** “Malfunction of TSF” describes the use of a malfunction of the hardware or software in order to deactivate, modify, or circumvent security functions of the TOE to enable attacks against the integrity or confidentiality of the User data or the CSP. The security objective O.Check_Operation prevents this threat by regular checks verifying that TOE components operate correctly.

The threat **T.Physical_Tamper** “Physical tampering” describes tampering the cryptographic module to get secrets, to modify data on whose integrity the TSF relies, or to corrupt or deactivate the TSF inside the cryptographic boundary, which is directly addressed by the inclusion of OE.Env to ensure the environment protects against direct physical tampering of the TOE.

The threat **T.Masquerade** “Masquerade authorised data source or receiver” describes that an attacker may masquerade as an authorised data source or receiver to perform operations that will be attributed to the authorised user or gains undetected access to cryptographic module causing potential violations of integrity, or confidentiality. The security objective O.I&A requires the TOE to identify and authenticate the user before providing access to any controlled resources with the exception of public objects. The security objective O.Control_Keys restricts the access to the keys, key components and other CSP according to their security attributes (including Key entity). OE.Client_Management provides protection against any attempts to hijack an authenticated and active session.

The assumptions **A.User_Data** “Protection of user data by the IT system” and **A.Data_Sep** “Separation of cryptographically protected and unprotected data “ are covered by the security objective for the IT environment OE.Red-Black-Sep “Separation of red and black area of the IT system” dealing with protection of the user data in the red area of the IT system, their security attributes for cryptographic protection to the TOE and the control the exchange data between the red and black area of the IT system according to the IT security policy.

The assumption **A.Key_Generation** “Key generation and import to the cryptographic module” deals with the cryptographic strength and secure security attributes of cryptographic keys generated by the IT environment and imported into the TOE. This assumption is directly and completely covered by the security objective for the IT environment OE.Key_Generation.

The assumption **A.Availability** “Availability of keys” describes that the IT environment ensures the availability of cryptographic keys and keying material as ensured by the security objective for the IT environment OE.Key_Availability.

The assumption **A.Env** “Protected Operating Environment” requires the environment protect the TOE against direct physical attacks which is ensured by the security objective for the environment OE.Env.

The assumption **A.Client_Management** requires the stateful management of commands sent and received by the TOE alongside integrity and confidentiality protection of message from Users authenticated to the TOE which is ensured by the security objective for the environment OE.Client_Management.

Refer to Section 8 for detailed mappings and descriptions of the rationale for the Security Objectives, IT Security Requirements and Dependencies, Assurance Measures, and Security Functional Requirements.

Table 4-1 provides an overview for security objectives rationale.

	OSP.User_Data_Prot	OSP.Resist_Moderate(*)	OSP.I&A	OSP.Access	OSP.Roles	OSP.Endorsed_Crypto	OSP.Key_Man	OSP.Key_Personnel	T.Compro_CSP	T.Modif_CSP	T.Abuse_Func	T.Inf.Leakage	T.Malfunction	T.Physical_Tamper	T.Masquerade	A.User_Data	A.Data_Sep	A.Key_Generation	A.Availability	A.Env(*)	A.Client_Management(*)
O.I&A		X	X					X							X						
O.Control_Services				X							X										
O.Control_Keys				X				X	X	X					X						
O.Roles				X	X						X										
O.Key_Export							X	X	X												
O.Key_Generation						X	X		X												
O.Key_Import							X	X	X												
O.Key_Management							X	X	X	X											
O.Key_Destruction							X		X												
O.Check_Operation										X			X								
O.Endorsed_Crypto	X					X															
O.Prevent_Inf_Leakage(*)		X							X			X									
OE.Assurance		X																			
OE.Key_Generation							X		X									X			
OE.Red-Black-Sep																X	X				
OE.Personnel					X			X			X										
OE.Key_Availability	X																		X		
OE.Env(*)		X							X	X	X	X		X						X	
OE.Client_Management(*)							X	X	X	X	X				X						X

Table 4-1 – Security Objectives Rationale

5 Extended Components Definition

The following non Part 2 Security Functional Extended Component is included to meet specific requirements of the TOE.

5.1 Definition of the Family FCS_RNG

Family behaviour

This section describes the functional requirements for the generation of random numbers, which may be used as secrets for cryptographic purposes or authentication. The IT security functional requirements for a TOE are defined in an additional family (FCS_RNG) of the Class FCS (Cryptographic support). The requirements address the type of the random number generator as defined in AIS 20 [9] and AIS 31 [10] and quality of the random numbers.

Component levelling

FCS_RNG.1 Generation of random numbers requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1	**Random number generation**
Hierarchical to**:**	No other components.
Dependencies:	No dependencies

FCS_RNG.1.1 The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic] random number generator that implements: [assignment: list of security capabilities].

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

5.2 Definition of the Family FPT_EMS

Family behaviour

This family defines requirements to mitigate intelligible emanations.

Component levelling

FPT_EMS.1 TOE Emanation

Management: FPT_EMS.1

There are no management activities foreseen**.**

Audit: FPT_EMS.1

There are no actions defined to be auditable.

FPT_EMS.1	**TOE Emanation**
Hierarchical to:	No other components.
Dependencies:	No dependencies

FPT_EMS.1.1 The TOE shall not emit [assignment: types of emissions] in excess of [assignment: specified limits] enabling access to [assignment: list of types of TSF data] and [assignment: list of types of user data].

FPT_EMS.1.2 The TSF shall ensure [assignment: type of users] are unable to use [assignment: types of interfaces/ports] to gain access to [assignment: list of types of TSF data] and [assignment: list of types of user data].

5.3 Definition of the Security Functional Component FPT_TST.2

The following additions are made to “TSF self-test (FPT_TST)” in Common Criteria, Part 2 to require the self-testing of TSF and of the integrity of the TSF-data and TSF-executable code. FPT_TST.2 requires the behaviour of TSF during self-testing and the actions to be performed by TSF in dependency of the results of the self-testing. This kind of requirements lies beyond FPT_TST.1 defined in Common Criteria, Part 2.

Family behaviour

The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. Examples are interfaces to enforcement functions, and sample arithmetical operations on critical parts of the TOE. These tests can be carried out at start-up, periodically, at the request of the authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self-testing are defined in other families.

The requirements of this family are also needed to detect the corruption of TSF executable code (i.e. TSF software) and TSF data by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TSF due to inadequate logical and/or physical protection.

Component levelling

FPT_TST.1 TSF testing provides the ability to test the TSF's correct operation. These tests may be performed at start-up, periodically, at the request of the authorised user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and executable code.

FPT_TST.2 TSF self-testing requires self-testing capabilities of the TSF correct operation. These tests must be performed at start-up. Conditional and on-demand by a user self-testing may be required. Particular TSF behaviour during self-testing and TSF-actions after self-testing is required.

Management: FPT_TST.2

There are no management activities foreseen.

Audit: FPT_TST.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Execution of the TSF self-tests and the results of the tests.

FPT_TST.2	**TSF self-testing**
Hierarchical to:	No other components.
Dependencies:	FPT_FLS.1 Failure with preservation of secure state.

FPT_TST.2.1 The TSF shall perform self-testing at power-up to verify the correctness of [*assignment: list of cryptographic algorithms*] and of [*assignment: list of critical TSF*], and to verify the integrity of the TSF-software/firmware.

FPT_TST.2.2 The TSF shall perform self-testing at the conditions [*assignment: list of conditions*] to verify the correctness of [*assignment: list of critical cryptographic algorithms*].

FPT_TST.2.3 The TSF shall perform self-testing at the conditions [*assignment: list of conditions*] to verify the correctness of [*assignment: list of critical TSF*], and to verify the integrity of [*assignment: list of TSF data*].

FPT_TST.2.4 The TSF shall perform self-testing at the conditions [*assignment: list of conditions*] to verify the integrity of [*assignment: list of TSF-objects*].

FPT_TST.2.5 The TSF shall provide [*assignment: list of users*] with the capability to invoke the following self-tests [*assignment: list of self-tests*].

FPT_TST.2.6 During [assignment: list of self-tests] the TSF shall [assignment: list of actions to be performed].

FPT_TST.2.7 After completion of self-testing the TSF shall [*assignment: list of actions to be performed*].

FPT_TST.2.8 If the self-testing result is fail the TSF shall [*assignment: list of actions to be performed*].

6 Security Requirements

This chapter gives the security functional requirements (SFRs) and the security assurance requirements (SARs) for the TOE. These are taken directly from [Cryptographic Modules, Security Level “Enhanced Basic”] without modification except FPT_PHP.1 and FPT_PHP.3 have been removed which corresponds to the removal of O.Physical_Protect as a result of the addition of OE.Env and independently, FDP_UCT and FDP_UIT have been removed as surplus requirements in light of the absence of the Red-Black SFP and protection to be provided by the environment under OE.Client_Management.

Security functional requirements components given in section 6.1 are drawn from the Common Criteria (ISO 15408), Version 3.1, Part 2 [2]. In addition, extended security functional requirements are also used; these are defined in Section 5. Operations for assignment, selection and refinement have been made as needed.

TOE Security Assurance Requirements, section 6.2, are drawn from the security assurance components from Common Criteria, Version 3.1, Part 3 [3].

The conventions used which are carried through from [Cryptographic Modules, Security Level “Enhanced Basic”] are also used in this Security Target to indicate operations that have been performed on the CC functional components:

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is (i) denoted by the word “refinement” in **bold** text and the added/changed words are in bold text, or (ii) directly included in the requirement text as **bold** text. In cases where words from a CC requirement were deleted, a separate attachment indicates the words that were removed.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the PP authors are denoted as *italic* text and the original text of the component is given by a footnote. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made, [selection:], and are *italicized*.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the PP authors are denoted by showing as *italic* text and the original text of the component is given by a footnote. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:], and are *italicized*.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash “/” and the iteration indicator after the component identifier.

6.1 TOE Security Functional Requirements

6.1.1 Cryptographic operation and key management

6.1.1.1 FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with **the**[14] specified cryptographic key generation **algorithms**[15] [assignment*: listed below*] and specified cryptographic key sizes [assignment*: specified for each algorithm below*] that meets the following: [assignment*: see list below*

(1) *RSA 1024, 2048, 3072 and 4096 bits key pairs in accordance with FIPS PUB 186-3 and ANSI X9.31.*

(2) *TDES 112, 168 bits (security strength) keys in accordance with NIST SP 800-67.*

(3) *AES 128, 192, 256 bits keys in accordance with FIPS PUB 197.*

(4) *DSA 1024, 2048 and 3072 bits key pairs in accordance with FIPS PUB 186-3.*

(5) *Elliptic Curve key pairs P-192, P-224, P-256, P-384, P-521, K-163, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409, B-571 in accordance with FIPS PUB 186-3 and ANSI X9.62.*

(6) *EC Diffie-Hellman Key Agreement curves P-192, P-224, P-256, P-384, P-521, K-163, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409 and B-571 in accordance with NIST SP-800-56A.*]

6.1.1.2 FCS_CKM.2/Private_Key_Import Cryptographic key distribution

FCS_CKM.2.1/ Private_Key_Import The TSF shall distribute **private key**[16] in accordance with a specified cryptographic key distribution method *key entry*[17] that meets the following: [assignment*: see list below:*

(1) *Encrypted Private Key Info Syntax format from PKCS #8: Private-Key Information Syntax Standard using TDES (112, 168 bit keys) in CBC mode*

(2) *Encrypted Private Key Info Syntax format from PKCS #8: Private-Key Information Syntax Standard using AES (128, 192 and 256 bit keys) in CBC mode*].

Refinement:

The key entry shall be performed using electronic methods.

All secret key (symmetric cryptographic algorithm) or private keys that are imported into the TOE in encrypted form shall be encrypted using an Endorsed cryptographic algorithm.

Application note**:** Implemented using the key wrapping protocol for symmetric keys.

Application note: SFR renamed from FCS_CKM.2/Import for clarity and consistency.

[14] a

[15] algorithm

[16] cryptographic keys

[17] *[assignment: cryptographic key distribution method]*

6.1.1.3 FCS_CKM.2/Secret_Symmetric_Key_Import Cryptographic key distribution

FCS_CKM.2.1/ Secret_Symmetric_Key_Import The TSF shall distribute **secret key (symmetric cryptographic algorithm)**[18] in accordance with a specified cryptographic key distribution method *key entry*[19] that meets the following: [assignment*: see list below:*

(1) *RSAES-OAEP from PKCS #1 v2.1 with RSA Modulus sizes of 1024-4096 bits;*

(2) *TDES from NIST SP-800-67 in CBC mode with 112 and 168 bit keys;*

(3) *TDES from NIST SP-800-67 in ECB mode with 112 and 168 bit keys;*

(4) *AES from FIPS PUB 197 in CBC mode with 128, 192 and 256 bit keys;*

(5) *AES from FIPS PUB 197 in ECB mode with 128, 192 and 256 bit keys*].

Refinement:

The key entry shall be performed using electronic methods.

All secret or private keys that are imported into the TOE in encrypted form shall be encrypted using an Endorsed cryptographic algorithm.

Application note**:** Implemented using the key wrapping protocol for asymmetric keys.

Application note: SFR renamed from FCS_CKM.2/Import for clarity and consistency.

6.1.1.4 FCS_CKM.2/Cloning_Import Cryptographic key distribution

FCS_CKM.2.1/Cloning_Import The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method *key entry*[20] that meets the following: [assignment: *Safenet Cloning Protocol using AES-256 in CBC mode for confidentiality and SHA-256 for Integrity*].

Refinement:

The key entry shall be performed using electronic methods.

All secret or private keys that are imported into the TOE in encrypted form shall be encrypted and integrity protected using an Endorsed cryptographic algorithm.

Application Note: Implemented using the Luna Cloning Protocol.

Application Note: SFR renamed from FCS_CKM.2/Import for clarity and consistency.

6.1.1.5 FCS_CKM.2/Private_Key_Export Cryptographic key distribution

FCS_CKM.2.1/Private_Key_Export The TSF shall distribute **private key** [21] in accordance with a specified cryptographic key distribution method *key export*[22] that meets the following: [assignment*: see list below:*

[18] cryptographic keys
[19] *[assignment: cryptographic key distribution method]*
[20] *[assignment: cryptographic key distribution method]*
[21] cryptographic keys
[22] *[assignment: cryptographic key distribution method]*

(1) *Encrypted Private Key Info Syntax format from PKCS #8: Private-Key Information Syntax Standard using TDES (112, 168 bit keys) in CBC mode;*

(2) *Encrypted Private Key Info Syntax format from PKCS #8: Private-Key Information Syntax Standard using AES (128, 192 and 256 bit keys) in CBC mode;*]

Refinement:

The key export shall be performed using electronic key export methods.

All secret or private keys exported in encrypted form by the TOE shall be encrypted using an Endorsed cryptographic algorithm.

Application Note: Implemented using the key wrapping protocol for symmetric keys.

Application note: SFR renamed from FCS_CKM.2/Export for clarity and consistency.

6.1.1.6 FCS_CKM.2/Secret_Symmetric_Key_Export Cryptographic key distribution

FCS_CKM.2.1/ Secret_Symmetric_Key_Export The TSF shall distribute **secret key (symmetric cryptographic algorithm)**[23] in accordance with a specified cryptographic key distribution method *key export*[24] that meets the following: [assignment*: see list below:*

(1) *RSAES-OAEP from PKCS #1 v2.1 with RSA Modulus sizes of 1024-4096 bits;*

(2) *TDES from NIST SP-800-67 in CBC mode with 112 and 168 bit keys;*

(3) *TDES from NIST SP-800-67 in ECB mode with 112 and 168 bit keys;*

(4) *AES from FIPS PUB 197 in CBC mode with 128, 192 and 256 bit keys;*

(5) *AES from FIPS PUB 197 in ECB mode with 128, 192 and 256 bit keys*].

Refinement:

The key export shall be performed using electronic key export methods.

All secret or private keys exported in encrypted form by the TOE shall be encrypted using an Endorsed cryptographic algorithm.

Application Note: Implemented using the key wrapping protocol for asymmetric keys.

Application note: SFR renamed from FCS_CKM.2/Export for clarity and consistency.

6.1.1.7 FCS_CKM.2/Cloning_Export Cryptographic key distribution

FCS_CKM.2.1/Cloning_Export The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method *key export*[25] that meets the following: [assignment: *Safenet Cloning Protocol using AES-256 in CBC mode for confidentiality and SHA-256 for Integrity*].

Refinement:

The key export shall be performed using electronic key export methods.

[23] cryptographic keys

[24] *[assignment: cryptographic key distribution method]*

[25] *[assignment: cryptographic key distribution method]*

All secret or private keys exported in encrypted form by the TOE shall be encrypted and integrity protected using an Endorsed cryptographic algorithm. All public keys exported for electronic key entry method shall be integrity protected using an Endorsed cryptographic algorithm.

Application Note: Implemented by means of the Luna Cloning Protocol.

Application Note: SFR renamed from FCS_CKM.2/Export for clarity and consistency.

6.1.1.8 FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *overwrite*] that meets the following: [assignment: *overwrite of stored key values in memory*].

6.1.1.9 FTP_ITC.1/Cloning Inter-TSF trusted channel

FTP_ITC.1.1/Cloning The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/Cloning The TSF shall permit [selection*: another trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC.1.3/Cloning The TSF shall initiate communication via the trusted channel for *electronic key distribution according to* ***FCS_CKM.2/Cloning_Import and FCS_CKM.2/Cloning_Export***[26]**.**

6.1.1.10 FCS_COP.1/Sign Cryptographic operation - Digital signature

FCS_COP.1.1/Sign The TSF shall perform [assignment: *digital signature generation and verification*] in accordance with a specified cryptographic algorithm [assignment: *from the list below*] and cryptographic key sizes [assignment: *as specified for each algorithm listed below*] that meet the following: [assignment:

(1) *RSA 1024-4096 bits with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (RSASSA-PSS and RSASSA-PKCS1-v1_5 from PKCS #1),*

(2) *RSA 1024 -4096 bits with SHA-1, SHA-224, SHA-256, SHA-384, SHA 512 (FIPS PUB 186-3 and ANSI X9.31),*

(3) *DSA 1024, 2048 and 3072 bits with SHA-1, SHA-256, SHA-384, SHA 512 (FIPS PUB 186-3),*

(4) *ECDSA curves P-192 P-224 P-256 P-384 P-521 K-163 K-233 K-283 K-409 K-571 B-163 B-233 B-283 B-409 B-571 with SHA-1, SHA-256, SHA-384, SHA 512 (FIPS PUB 186-3)*].

6.1.1.11 FCS_COP.1/Digest Cryptographic operation - Message digest

FCS_COP.1.1/Digest The TSF shall perform [assignment: *message digest*] in accordance with a specified cryptographic algorithm [assignment: *from the list below*] and cryptographic key sizes [assignment: *as specified for each hash function below*] that meet the following: [assignment:

(1) *SHA-1 / 160 bits / FIPS PUB 180-4*

[26] [assignment: *list of functions for which a trusted channel is required*]

(2) *SHA-224 / 224 bits / FIPS PUB 180-4,*

(3) *SHA-256 / 256 bits / FIPS PUB 180-4,*

(4) *SHA-384 / 384 bits / FIPS PUB 180-4,*

(5) *SHA-512 / 512 bits / FIPS PUB 180-4*].

6.1.1.12 FCS_COP.1/RSA_Enc_Dec Cryptographic operation - RSA Encrypt/Decrypt

FCS_COP.1.1/RSA_Enc_Dec The TSF shall perform [assignment: *asymmetric encryption and decryption*] in accordance with a specified cryptographic algorithm [assignment: *RSA*] and cryptographic key sizes [assignment: *1024 – 4096 bits*] that meet the following: [assignment: *RSAES-OAEP from PKCS #1 v2.1*].

6.1.1.13 FCS_COP.1/TDES_Enc_Dec Cryptographic operation - TDES Encrypt/Decrypt

FCS_COP.1.1/TDES_Enc_Dec The TSF shall perform [assignment: *symmetric encryption and decryption*] in accordance with a specified cryptographic algorithm [assignment: *Triple DES (ECB, CBC and CFB-8 mode)*] and cryptographic key sizes [assignment: *112 and 168 bits*] that meet the following: [assignment: *NIST SP 800-67*].

6.1.1.14 FCS_COP.1/AES_Enc_Dec Cryptographic operation - AES Encrypt/Decrypt

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

6.1.1.15 FCS_COP.1/Key_Derive Cryptographic operation – Key Derivation

FCS_COP.1.1/Key_Derive The TSF shall perform [assignment: *key derivation*] in accordance with a specified cryptographic algorithm [assignment: *Counter Mode KDF with AES or TDES)*] and cryptographic key sizes [assignment: *112 -256 bits*] that meet the following: [assignment: Counter Mode KDF from *FIPS SP 800-108*].

6.1.1.16 FCS_COP.1/MAC Cryptographic operation – MAC

FCS_COP.1.1/MAC The TSF shall perform [assignment: Message Authentication Codes] in accordance with a specified cryptographic algorithm [assignment: *from the list below*] and cryptographic key sizes [assignment: *as specified for each function below*] that meet the following: [assignment:

(1) *HMAC with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (FIPS PUB 198-1),*

(2) *TDES MAC with 168 bit keys (FIPS PUB 113),*

(3) *AES CMAC with 128, 192 and 256 bit keys (NIST SP 800-38B),*

(4) *TDES CMAC with 112 and 168 bit keys (NIST SP 800-38B),*

(5) *AES GMAC with 128, 192 and 256 bit keys (NIST SP 800-38D].*

6.1.1.17 FCS_RNG.1 Random number generation

FCS_RNG.1.1 The TSF shall provide a [selection: *deterministic*] random number generator that implements [assignment: *a Deterministic Random Bit Generator (CTR_DRBG with AES-256) conformant to NIST SP 800-90A, seeded by an internal Hardware Non-deterministic Random Bit Generator]*.

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: in*dependent bits with Shannon entropy of greater than 7.9999 bits per octet based on a* 2^{24} *bit data-set*].

6.1.2 User Identification and authentication

6.1.2.1 FIA_ATD.1 User attribute definition

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

(1) *User identity,*

(2) *Role,*

(3) *Reference authentication data,*

(4) [assignment*: User failed login count, User "locked" flag*].

6.1.2.2 FIA_UID.1 Timing of identification

FIA_UID.1.1 The TSF shall allow

(1) *Self-test according to FPT_TST.2,*

(2) [assignment: *none*[27]],

(3) [assignment:

a. *query HSM status, authenticated identity, configuration and licenses,*

b. *query container configuration,*

c. *query container object identify (from known OUID or object handle),*

d. *session management functions (i.e. open, close, close all, clean access),*

e. *MTK re-generation and unlock,*

f. *HSM deactivation,*

g. *PED configuration and communication requests;*

h. *Host-to-HSM communication channel tests;*

i. *query log status and submit external log messages for addition to secure audit log,*

[27] [assignment: list of cryptographic operations with public keys]

j. *Host-to-Luna PCI-E external USB interface communication tunneling]* [28]

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.

Application Note: MTK re-generation is a factory operation performed during re-manufacture of an HSM where an existing MTK is zeroised and re-generated.

6.1.2.3 FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow

(1) *Self-test according to FPT_TST.2*,

(2) [assignment: *none*] [29]*,*

(3) Identification according to FIA_UID.1,

(4) Selection of [selection: *a role*],

(5) [assignment:

a. *query HSM status, authenticated identity, configuration and licenses,*

b. *query container configuration,*

c. *query container object identify (from known OUID or object handle),*

d. *session management functions (i.e. open, close, close all, clean access),*

e. *MTK re-generation and unlock,*

f. *HSM deactivation,*

g. *PED configuration and communication requests;*

h. *Host-to-HSM communication channel tests;*

i. *query secure audit log status and submit external log messages for storage,*

j. *Host-to-Luna PCI-E external USB interface communication tunneling,*

k. *request login against a role or login to a role on an HA paired container,*

l. *create, modify, destroy and get attributes of public partition objects,*

m. *request entropy,*

n. *session management function (*close, get session info),

o. *request cryptographic digest of data,*

p. *request a challenge for challenge-response authentication secondary authentication mechanism,*

q. *zeroize the HSM.*] [30]

[28] [assignment: list of TSF mediated actions]
[29] [assignment: list of cryptographic operations with public keys]
[30] [assignment: list of other TSF mediated actions]

on behalf of the user to be performed before the user is authenticated.

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

6.1.2.4 FIA_UAU.6 Re-authenticating

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions:

(1) Changing to a role not selected for the current valid authentication session,

(2) power on or reset,

(3) [assignment: *No other conditions*][31].

6.1.2.5 FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only [assignment: *status messages to trusted PIN entry device*] to the user while the authentication is in progress.

Application Note: Status messages alert the trusted PIN Entry device that the TOE is ready to receive authentication data. The trusted PIN entry device is responsible for the content of the status message data and feedback to the user.

6.1.2.6 FIA_USB.1 User-subject binding

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

(1) User identity,

(2) Role,

(3) [assignment: *Challenge Secret*]

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: *the initial role of the user is Unidentified User*.

Application note: Unidentified User is established by checking the following parameters:

- User ID is Public (unidentified)
- User Challenge Secret is Nil.
- User role is Nil.

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

(1) the subject attribute Role shall be changed from Unidentified user to Unauthenticated user after successful identification;

[31] [assignment: list of other conditions under which re-authentication is required]

(2) *after successful authentication the subject attribute Role shall be changed from Unauthenticated User to a role that the user has selected for the authentication session if the user is authorised for this role;*

(3) *after successful re-authentication of the user the subject attribute Role shall be changed to a role that the user has selected for the authentication session if the user is authorised for this role,*

(4) [assignment*: no additional rules*[32]]

6.1.2.7 FIA_AFL.1/CO Authentication failure handling

FIA_AFL.1.1/CO The TSF shall detect when [selection: *[assignment: an* ***SO configurable***[33] *positive integer within the range of three (3) to ten (10)]* unsuccessful authentication attempts occur related to [assignment: *Crypto Officer authentication*]**.**

FIA_AFL.1.2/CO When the defined number of unsuccessful authentication attempts has been [selection: *met*], the TSF shall [assignment: *block the identity for authentication*]*.*

Application Note: The TOE blocks the identity for authentication by terminating the session establishment and, according to the SO configurable policy by:

(1) removing the Crypto Officer and clearing the Crypto Officers memory space and permanent storage, or

(2) disabling the Crypto Officer account by setting the locked flag in the Crypto Officer's attributes (FIA_ATD.1).

6.1.2.8 FIA_AFL.1/SO Authentication failure handling

FIA_AFL.1.1/SO The TSF shall detect when [selection: [assignment: *three (3)*]] unsuccessful authentication attempts occur related to [assignment: *Security Officer authentication*]*.*

FIA_AFL.1.2/SO When the defined number of unsuccessful authentication attempts has been [selection: *met*], the TSF shall [assignment: *change the state of the TOE to require re-initialization*]*.*

6.1.2.9 FIA_AFL.1/User Authentication failure handling

FIA_AFL.1.1/User The TSF shall detect when [selection: [*assignment: an* ***SO configurable***[34] *positive integer within the range of three (3) to ten (10)*] unsuccessful authentication attempts occur related to [assignment: *User authentication*]**.**

FIA_AFL.1.2/User When the defined number of unsuccessful authentication attempts has been [section: *met*], the TSF shall [assignment: *block the identity for authentication*].

Application Note: The TOE blocks the identity for authentication by terminating the session establishment and disabling the User account by setting the User locked flag in the User's attributes (FIA_ATD.1).

[32] [assignment: additional rules for the changing of attributes]
[33] *administrator-configurable*
[34] *administrator-configurable*

6.1.3 Protection of user data

6.1.3.1 FDP_ACC.2/Key_Man Complete access control

FDP_ACC.2.1/Key_Man The TSF shall enforce the *Key Management SFP*[35] on:

(1) *all cryptographic keys, key components, CSP;*

(2) *all subjects acting on behalf of users*[36].

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/Key_Man The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

6.1.3.2 FDP_ACF.1/Key_Man Security attribute based access control

FDP_ACF.1.1/Key_Man The TSF shall enforce the *Key Management SFP*[37] to objects based on the following:

(1) *Subjects with security attributes: User identity the subject is bound to, Role of this user;*

(2) *Objects*

a. *Cryptographic keys with security attributes: Key identity, Key entity, Key type, Key usage type, Key access control rules;*

b. *Key components with security attributes: Key component identity, Key entity, Key entry method,*

c. *CSP with security attributes: CSP Identity, CSP usage type, CSP access control rules*[38].

FDP_ACF.1.2/Key_Man The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) *Subject in Crypto Officer role is allowed to import encrypted secret and private keys if the security attribute Key access control rules of the key allows import;*

(2) *Subject in Crypto Officer role is allowed to import one key component of a key with the key entry method assigned to the key component;*

(3) *Subject in Crypto Officer role is allowed to import CSP,*

(4) *Subject in Crypto Officer role is allowed to export encrypted secret or private keys if the security attribute Key access control rules of the key allows export;*

(5) *Subject in Crypto Officer role is allowed to export one key component of a key with the key entry method assigned to the key component;*

(6) *Subject in Crypto Officer role is allowed to export CSP if the security attribute CSP access control rules of the CSP allows export;*

[35] [assignment: *access control SFP*]
[36] [assignment: *list of subjects and objects*]
[37] [assignment: access control SFP]
[38] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or names groups of SFP-relevant security attributed*]

(7) Subject in Crypto Officer role is allowed to destruct cryptographic keys, cryptographic key components and CSP.

(8) [assignment: *None*][39].

FDP_ACF.1.3/Key_Man The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

(1) Subjects in Maintenance role are allowed to import and destruct maintenance cryptographic keys, key components and CSP;

(2) [*assignment: no other rules*][40]*.*

Application Note: The TOE does not support a maintenance role or the concept of maintenance cryptographic keys, key components and CSP. This TOE Security functional requirement is retained for consistency with Protection Profile – BSI-CC-PP-0078 only.

FDP_ACF.1.4/Key_Man The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

(1) Subject in Crypto Officer role is not allowed to import a key component if the same subject or another subject with the same User identity already input a key component with a different Key component identity and the same Key entity;

(2) Subject in Crypto Officer role is not allowed to export a key component if the same subject or another subject with the same Identity of the user already export a key component with a different Identity and the same Key entity;

(3) Subjects with other roles than Crypto Officer role are not allowed to input operational public root key;

(4) Subjects with other roles than Crypto Officer role are not allowed to input permanent stored operational secret keys, private keys, key components and CSP;

(5) No subject is allowed to import or export secret key or private keys in plaintext;

(6) No subject is allowed to use keys by operation other than identified in Key usage type and the Key access control rules;

(7) [assignment: *no other rules*][41]*.*

6.1.3.3 FDP_ACC.2/Oper Complete access control

FDP_ACC.2.1/Oper The TSF shall enforce the *Cryptographic Operation SFP*[42] on:

(1) operational cryptographic keys, CSP,

(2) plaintext data, ciphertext data, original data;

[39] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[40] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[41] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]
[42] [assignment: *access control SFP*].

(3) all subjects acting on behalf of users[43]

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/Oper The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

6.1.3.4 FDP_ACF.1/Oper Security attribute based access control

FDP_ACF.1.1/Oper The TSF shall enforce the *Cryptographic Operation SFP*[44] to objects based on the following:

(1) Subjects with security attributes: User Identity the subject is bound to, Role of this user;

(2) Objects

a. Operational cryptographic keys with security attributes: Key identity, Key entity, Key type, Key usage type, Key access control rules;

b. Operational CSP with security attributes: CSP identity, CSP usage type, CSP access control rules,

c. plaintext data, ciphertext data, original data.[45]

FDP_ACF.1.2/Oper The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed[46]:

(1) Subject in ***Crypto User***[47] *role is allowed to perform cryptographic operation with public, private, and secret keys in accordance with the security attributes of the used cryptographic keys and CSP;*

(2) [assignment: *Subject in Crypto Officer or Crypto User role is allowed to perform cryptographic operation in accordance with the security attributes of the used cryptographic keys and CSP*][48]

Application Note: The following rules has been removed from FDP_ACF.1.2/Oper in this Security Target from the original PP: “Subject in Unauthenticated user role is allowed to perform cryptographic operation with public keys in accordance with the security attributes of the used cryptographic keys and CSP”. No support is provided in the TOE to allow this operation and hence it’s been removed as an assurance claim.

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

FDP_ACF.1.4/Oper The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

[43] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

[44] [assignment: *access control SFP*].

[45] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

[46] The Luna TAC policy contains additional security policy rules, governing access to and usage of keys and other objects, enforced by the TSF.

[47] End-User

[48] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

(1) No subject is allowed to use cryptographic keys by cryptographic operation other than identified in the security attributes Key usage type and the Key access control rules;

(2) No subject is allowed to use CSP by cryptographic operation other than identified in the security attributes CSP usage type and the CSP access control rules;

(3) [assignment: *no other rules*][49]*.*

6.1.3.5 FDP_ACC.2/Mode_Trans Complete access control

FDP_ACC.2.1/Mode_Trans The TSF shall enforce the *Mode transition SFP*[50] on *all subjects acting on behalf of users and the mode variable*[51] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/Mode_Trans The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

6.1.3.6 FDP_ACF.1/Mode_Trans Security attribute based access control

FDP_ACF.1.1/Mode_Trans The TSF shall enforce the *Mode transition SFP*[52] to objects based on the following: *all subjects acting on behalf of users and the mode variable*[53].

FDP_ACF.1.2/Mode_Trans The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) the subject in Crypto Officer role is allowed to change the mode variable to a ***Key Management mode, Crypto User mode***[54] *or a Self-test mode;*

(2) the subject in ***Crypto User***[55] *role is allowed to change the mode variable to* ***Crypto User***[56] *mode*[57]*.*

FDP_ACF.1.3/Mode_Trans The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

(1) the TOE shall enter automatically an Error mode from any mode of operation except a Power-off mode or a Maintenance mode, when failure listed in FPT_FLS.1 occur,

(2) [assignment: *no other rules*][58].

FDP_ACF.1.4/Mode_Trans The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

[49] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]
[50] [assignment: *access control SFP*]
[51] [assignment: *list of subjects and objects*].
[52] [assignment: *access control SFP*].
[53] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or names groups of SFP-relevant security attributes*]
[54] *Key/CSP entry mode, Crypto User mode*
[55] User
[56] User
[57] [assignment: *rules governing access among controlled subjects and controlled objects using operations on controlled objects*]
[58] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

(1) *Subjects in other roles than the Crypto Officer are not allowed to change the mode variable to a Crypto Officer mode or a* ***Key Management mode***[59]*;*

(2) [assignment: *no other rules*][60]

Application note: The TOE does not provide any maintenance functionality and the Maintenance modes, therefore, do not exist and the Maintenance Personnel Role is superfluous (cf. to FMT_SMR.2). In this case the rules (2) and (3) in FDP_ACF.1.2/Mode_Trans, (1) in FDP_ACF.1.2/Mode_Trans and (2) in FDP_ACF.1.4/Mode_Trans have been removed from this ST but were present in [Cryptographic Modules, Security Level "Enhanced Basic"] as they do not apply to the TOE.

6.1.3.7 FDP_ITC.2 Import of user data with security attributes

FDP_ITC.2.1 The TSF shall enforce the *Key Management SFP*[61] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE

(1) *keys shall be imported with the security attributes Key identity, Key entity, Key type, Key usage type;*

(2) *key components shall be imported with the security attributes Identity of the Key, Key entity, Key entry method;*

(3) *CSP shall be imported with security attributes Identity of the CSP and CSP usage type;*

(4) *all secret and private keys imported into the TOE shall be encrypted or entered using split knowledge procedures using an Endorsed algorithm*[62]*.*

Application Note**:** All secret and private keys entered into the TOE and used by an Endorsed function shall be imported in encrypted form (cf. FCS_CKM.2.1/Private_Key_Import and FCS_CKM.2.1/ Secret_Symmetric_Key_Import, FCS_CKM.2.1/Cloning_Import[63]).

Application Note: The TOE does not support key import using split knowledge procedures.

6.1.3.8 FDP_ETC.2 Export of user data with security attributes

[59] *Key/CSP entry mode*

[60] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[61] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[62] [assignment: *additional importation control rules*]

[63] FCS_CKM.2.1/Import

FDP_ETC.2.1 The TSF shall enforce the *Key Management SFP*[64] when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2 The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE:

(1) keys shall be exported with the security attributes Key identity, Key entity, Key type, Key usage type;

(2) secret and private keys exported in encrypted form shall be exported with additional security attribute: Identity of the key encryption key under which they are encrypted;

(3) key components shall be exported with the security attributes Identity of the Key component, Key entity, Key entry method;

(4) CSP shall be exported with security attributes Identity of the CSP and CSP usage type;

(5) all secret and private keys exported from the TOE shall be encrypted or protected by split-knowledge procedure using an Endorsed algorithm[65].

Application Note**:** All secret and private keys exported from the TOE shall be exported in encrypted form (cf. FCS_CKM.2.1/Private_Key_Export and FCS_CKM.2.1/ Secret_Symmetric_Key_Export, FCS_CKM.2.1/Cloning_Export[66]).

Application Note: The TOE does not support key export using split knowledge procedures.

6.1.3.9 FDP_RIP.2 Full residual information protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to*] all objects.

6.1.4 Management of TSF and protection of TSF data

6.1.4.1 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

*(1) management of security functions behaviour (**FMT_MOF.1/SO**[67]),*

(2) management of Reference Authentication Data (FMT_MTD.1/Admin, FMT_MTD.1/User,

[64] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[65] [assignment: *additional exportation control rules*]
[66] FCS_CKM.2.1/Import
[67] *FMT_MOF.1/CO*

(3) management of security attributes of cryptographic keys, cryptographic key components and CSP (FMT_MSA.1/Key_Man_1, FMT_MSA.1/Key_Man_2, FMT_MSA.2, FMT_MSA.3).

(4) [assignment: *no other functions*][68].

6.1.4.2 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1 The TSF shall maintain the roles: ***Security Officer***[69]***, Crypto Officer, Crypto User***[70], *Unidentified User, Unauthenticated User,* [assignment: *no other roles*][71].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

(1) Any user identity assigned to the ***Security Officer***[72] *role must not simultaneously be assigned to the* ***Crypto User***[73] *role or the Crypto Officer role,*

(2) Any user identity assigned to the Crypto Officer role must not simultaneously be assigned to the Crypto User role or the ***Security Officer***[74] *role.*

(3) [assignment: *no other conditions*][75].

are satisfied.

6.1.4.3 FMT_MOF.1/CO Management of security functions behaviour

FMT_MOF.1.1/CO The TSF shall restrict the ability to [selection: see individual items from assigned list][76]the functions [assignment:

(1) Activate partition

(2) Enable the Crypto User role]

to *Crypto Officer]*[77].

Application Note: The Crypto Officer can only perform the aforementioned functions on partitions specifically assigned to them by the Security Officer (SO).

6.1.4.4 FMT_MOF.1/SO Management of security functions behaviour

FMT_MOF.1.1/SO The TSF shall restrict the ability to [selection: see individual items from assigned list][78] the functions [assignment:

[68] [assignment: *list of management functions to be provided by the TSF*].
[69] Administrator
[70] *End-User*
[71] [assignment: authorised identified roles]
[72] *Administrator*
[73] End-User
[74] *Administrator*
[75] [assignment: conditions for the different roles]
[76] [selection: determine the behaviour of, disable, enable, modify the behaviour of]
[77] [assignment: *the authorised identified roles*]
[78] [selection: determine the behaviour of, disable, enable, modify the behaviour of]

HSM Level

(1) HSM Cloning – SO may enable and disable / assign cloning domain)

(2) Remote Authentication – SO may enable and disable.

(3) Network Replication – SO may enable and disable.

(4) Force change of User authentication data – SO may enable and disable.

(5) Number of failed User login attempts allowed – SO may set.

Partition Level

(6) Partition reset – SO may enable and disable.

(7) Partition activation – SO may enable and disable.

(8) Partition auto-activation – SO may enable and disable.

(9) High Availability – SO may enable and disable.

(10) Multi-purpose keys – SO may enable and disable.

(11) Changing key attributes once a key has been created – SO may enable and disable.

(12) Operation without RSA blinding – SO may enable and disable.

(13) Signing operations with non-local keys – SO may enable and disable.

(14) Performing raw RSA operations – SO may enable and disable.

(15) Private key wrapping – SO may enable and disable

(16) Private key unwrapping – SO may enable and disable.

(17) Secret key wrapping – SO may enable and disable.

(18) Secret key unwrapping – SO may enable and disable.

(19) User key management capability – SO may enable and disable.

(20) Increment failed login attempt counter on failed challenge response validation – SO may enable and disable.

(21) RSA signing without confirmation – SO may enable and disable.]

to ***Security Officer***[79]:

Refinement:

If bypass mode is supported by the TOE then the TSF shall indicate through the status output interface/port when the TOE is in bypass mode.

Application Note:

The TOE does not support bypass mode. The ST author has introduced an iteration of the SFR to better describe management functions and role restrictions as implemented by the TOE. The module and partitions must be configured by the Security Officer before a Crypto Officer can manage objects on the designated partitions. The Crypto Officer can only create, use, destroy and backup/restore cryptographic objects on the partition and cannot modify any configuration parameters other than its own password).

[79] *Crypto Officer*

6.1.4.5 FMT_MTD.1/Admin Management of TSF data

FMT_MTD.1.1/Admin The TSF shall restrict the ability to *create*[80], *clear and delete*[81] the *Reference Authentication Data*[82] of all authorised users to ***Security Officer***[83].

6.1.4.6 FMT_MTD.1/User Management of TSF data

FMT_MTD.1.1/User The TSF shall restrict the ability to *modify*[84] the *Reference Authentication Data*[85] to **the corresponding authorised user**[86]**.**

6.1.4.7 FMT_MSA.1/Key_Man_1 Management of security attributes

FMT_MSA.1.1/Key_Man_1 The TSF shall enforce the *Key Management SFP*[87] to restrict the ability to *query*[88] the security attributes *Key identity, Key entity, Key type of the key, Key usage type, Key access control rules, Key entry method*[89] to *Crypto Officer or Crypto User*[90]*.*

6.1.4.8 FMT_MSA.1/Key_Man_2 Management of security attributes

FMT_MSA.1.1/Key_Man_2 The TSF shall enforce the *Key Management SFP*[91] to restrict the ability to *modify*[92] the security attributes *Key identity, Key entity, Key type, Key usage type, Key access control rules*[93] to *Crypto Officer*[94].

6.1.4.9 FMT_MSA.2 Secure security attributes

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for [assignment: *Key identity, Key entity, Key type, Key usage type, Key access control rules, CSP identity, CSP usage type, CSP access control rules*].

6.1.4.10 FMT_MSA.3 Static attribute initialisation

[80] "create" denotes initial creation and setting a new value in case a user forgot/lost their authentication data
[81] [selection: *change_default, query, modify, delete, clear,* [assignment: *other operations*]]
[82] [assignment: *list of authentication data*]
[83] [assignment: the authorised identified roles]
[84] [selection: *change_default, query, modify, delete, clear,* [assignment: *other operations*]]
[85] [assignment: *list of TSF data*]
[86] [assignment: *the authorised identified roles*]; here instead of role(s) the ability has been further restricted to just the user, who the Reference Authentication Data belong to.
[87] [assignment: *access control SFP(s), information flow control SFP(s)*].
[88] [selection: *change_default, query, modify, delete,*[assignment: *other operations*]]
[89] [assignment: *list of security attributes*]
[90] [assignment: *the authorised identified roles*]
[91] [assignment: *access control SFP(s), information flow control SFP(s)*].
[92] [selection: *change_default, query, modify, delete, clear,* [assignment: *other operations*]]
[93] [assignment: *list of security attributes*]
[94] [assignment: *the authorised identified roles*]

FMT_MSA.3.1 The TSF shall enforce the *Key Management SFP, Cryptographic Operation SFP and Mode Transition SFP*[95] *to* provide *restrictive*[96] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *Crypto Officer*[97] to specify alternative initial values to override the default values when an object or information is created.

6.1.5 TSF protection

6.1.5.1 FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret *security attributes of cryptographic keys, key components and CSPs*[98] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use *the following rules:*

(1) *the TOE reports about conflicts between the Key identity of stored cryptographic keys and cryptographic keys to be imported,*

(2) *the TOE does not change the security attributes Key identity, Key entity, Key type, Key usage type of keys being imported or exported,*

(3) *the TOE reports about conflicts between the Key component identity of stored key components and key components to be imported,*

(4) *the TOE does not change the security attributes Key component identity, Key entity, Key entry method of key components being imported,*

(5) *the TOE reports about conflicts between the CSP identity of stored CSPs and CSPs to be imported,*

(6) *the TOE does not change the security attributes CSP identity and CSP usage type of CSP being imported or exported*[99]

when interpreting the TSF data from another trusted IT product.

6.1.5.2 FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: *self-test fails*[100]**.**

Refinement:

When the TOE is in a secure error mode the TSF shall not perform any cryptographic operations and all data output interfaces/ports shall be inhibited by the TSF.

[95] [assignment: *access control SFP, information flow control SFP*]
[96] [selection, choose one of: *restrictive, permissive,* assignment: *other property*]]
[97] [assignment: *the authorised identified roles*]
[98] [assignment: *list of TSF data types*]
[99] [assignment: *list of interpretation rules to be applied by the TSF*]
[100] [assignment: *list of types of failures in the TSF*]

6.1.5.3 FPT_EMS.1 TOE Emanation

FPT_EMS.1.1 The TOE shall not emit [assignment: *conducted emissions, radiated emissions*][101] in excess of [assignment: *state of the art limits in order to have unintelligible emission*][102]

enabling access to

(1) *authentication data,*
(2) [assignment: *no other TSF data*][103]

and

(1) *"red data" containing confidential information,*
(2) *plaintext cryptographic secret or private key,*
(3) *cryptographic key components,*
(4) *confidential CSP,*
(5) [assignment: *no other user data*][104]

FPT_EMS.1.2 The TSF shall ensure [assignment: *unauthenticated users, unidentified users*][105] are unable to *use any interfaces or port*[106] to gain access to

(1) *authentication data (except the authentication interface/port during authentication process of the user),*
(2) [assignment: *no other TSF data*][107]

and

(1) *"red data" containing confidential information (except the red data input and output interface/port),*
(2) *plaintext cryptographic secret or private key,*
(3) *cryptographic key components (except key interface during import of the cryptographic key component),*
(4) *confidential CSP (except key interface during import of the confidential CSP).*
(5) [assignment: *no other user data*][108].

6.1.5.4 FPT_TST.1 TSF testing

FPT_TST.1.1 The TSF shall run a suite of self-tests [selection: *during initial start-up*] to demonstrate the correct operation of [selection: *the TSF*].

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of [selection: [assignment: *key object attributes marked as private for partition stored objects*]].

[101] [assignment: types of emissions]
[102] [assignment: specified limits]
[103] [assignment: *list of types of TSF data*]
[104] [assignment: list of types of user data]
[105] [assignment: type of users]
[106] [assignment: *types of interfaces/ports*]
[107] [assignment: *list of types of TSF data*]
[108] [assignment: *list of types of user data*]

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of *all TSF executable code stored in the TOE in form of software or firmware*[109].

Application Note: Key object attributes marked private include: secret key values for symmetric keys and private key parameters for all asymmetric key pairs.

6.1.5.5 FPT_TST.2 TSF self-testing

FPT_TST.2.1 The TSF shall perform self-testing at power-up to verify the correctness of [assignment: CTR_DRBG, *SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512, RSA, DSA, ECDSA, TDES, AES, ECDH, Counter Mode KDF* and of [assignment: *signature generation/verification, hash/keyed hash generation/verification, encryption/decryption*] and to verify the integrity of the TSF-software/firmware.

FPT_TST.2.2 The TSF shall perform self-testing at the conditions [assignment: *RNG generation, asymmetric key generation, loading of firmware*] to verify the correctness of [assignment: *CTR_DRBG, RSA, DSA, ECDSA*]

FPT_TST.2.3 The TSF shall perform self-testing at the conditions [assignment: *power on*] to verify the correctness of [assignment: *hashing functions*], and to verify the integrity of [assignment: *none*].

FPT_TST.2.4 The TSF shall perform self-testing at the conditions [assignment: *upon start up*] to verify the integrity of [assignment: *firmware*].

FPT_TST.2.5 The TSF shall provide [assignment: *none*] with the capability to invoke the following self-tests:

[assignment: *none*]

FPT_TST.2.6 During *initial start-up self-test, power-up self-test, self-test at the request of the authorised user [assignment: no other self-tests*[110]*]* the TSF shall *inhibit all output via the data interfaces/ports, and [assignment: prevent authentication to the TOE, any cryptographic operations]*[111].

FPT_TST.2.7 After completion of self-testing the TSF shall *output the results of the self-tests via the status output interface/port, and [assignment: none]*[112].

FPT_TST.2.8 If the self- testing result is fail the TSF shall *enter a secure state (see FPT_FLS.1) and output an error indicator via the status output interface/port, and [assignment: prevent further cryptographic operations]*[113]**.**

Refinement:

A *start-up test* shall be performed when the TOE is powered up (after being powered off) or on reset. A *List of cryptographic algorithms* shall include all Endorsed cryptographic algorithms employed by the TOE. In order to *verify the correctness* of cryptographic algorithms self-testing shall perform a known answer or a pair-wise consistency test. If the TOE module includes two

[109] [selection*: [assignment: parts of TSF],TSF*]

[110] [assignment: *list of tests*]

[111] [assignment: *list of actions to be performed*]

[112] [assignment: *list of actions to be performed*]

[113] [assignment: *list of actions to be performed*]

independent implementations of the same cryptographic algorithm, then the outputs of two implementations shall be compared.

In order to *verify the integrity of the TSF-software/firmware* a self-testing using an Endorsed error detection code (EDC) or Endorsed authentication technique shall be applied.

The *self-testing at the conditions* shall cover, if applicable, the following conditions: i) when a critical cryptographic algorithm or critical TSF operation is invoked, ii) pairwise consistency test for newly generated asymmetric key-pairs, iii) on software/firmware load test, iv) on manual key entry, and v) and on bypass events.

If the TOE provides *generation of public/private key pairs*, then the following pair-wise consistency tests for public and private keys shall be performed. The consistency of the keys shall be tested by the calculation and verification of a digital signature. If the digital signature cannot be verified, the test shall fail.

If *manual import of cryptographic keys or key components* into the TOE is supported, then the following manual key entry tests shall be performed. The cryptographic key or key components shall have an Endorsed EDC applied, or shall be entered using duplicate entries. If the EDC cannot be verified, or the duplicate entries do not match, the test shall fail.

If *load of software or firmware into the TOE* is supported, then the following software/firmware load tests shall be performed. An Endorsed authentication technique shall be applied to all validated software and firmware components when the components are externally loaded into the TOE. The calculated result shall be compared with a previously generated result. If the calculated result does not equal the previously generated result, the software/firmware integrity test shall fail.

If the TOE implements a *bypass capability* where the services may be provided without cryptographic processing, then the following bypass tests shall be performed to ensure that a single point of failure of TOE components will not result in the unintentional output of plaintext. The TSF shall test for the correct operation of the services providing cryptographic processing when a switch takes place between an exclusive bypass service and an exclusive cryptographic service. If the TOE can automatically alternate between a bypass service and a cryptographic service, providing some services with cryptographic processing and some services without cryptographic processing, then the TSF shall test for the correct operation of the services providing cryptographic processing when the mechanism governing the switching procedure is modified.

Application note: The TOE does not implement a bypass capability. A cryptographic algorithm shall have an independent known-answer self-test or the known-answer self-test shall be included with the associated cryptographic algorithm self-test. If the calculated output does not equal the known answer, the known answer self-test shall fail. If a known-answer self-test is not appropriate because the output of the cryptographic algorithms vary for a given set of inputs (e.g., a digital signature generated by means of the Digital Signature Algorithm) it shall be tested using a known-answer test or using the inverse cryptographic function (e.g., a digital signature is verified). Random number generators shall implement statistical or other appropriate tests.

Application note: KAT from block ciphers AES and TDES as listed in FPT_TST.2.1 include test for all supported key sizes and modes including their use to generate Message Authentication Codes i.e. TDES MAC, TDES CMAC, AES CMAC and AES GMAC as listed under FCS_COP.1/MAC.

6.2 TOE Security Assurance Requirements

In Table 6-1 the security assurance requirement components for the TOE are listed, they comprise EAL4 package augmented with ALC_FLR.2. The table also contains information, which of the chosen components has been refined in in [Cryptographic Modules, Security Level “Enhanced Basic”]. The actual refinements are then stated in the subchapters following the table. There only the refined security assurance requirement elements are stated, any security assurance requirement elements taken unchanged from CC Part 2 are not reproduced in this ST.

Assurance class	Assurance component	augmented?	refined?
ADV: Development	ADV_ARC.1 Security architecture description	No	Yes
	ADV_FSP.4 Complete functional specification	No	Yes
	ADV_IMP.1 Implementation representation of the TSF	No	Yes
	ADV_TDS.3 Basic modular design	No	Yes
AGD: Guidance documents	AGD_OPE.1 Operational user guidance	No	Yes
	AGD_PRE.1 Preparative procedures	No	No
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation	No	No
	ALC_CMS.4 Problem tracking CM coverage	No	No
	ALC_FLR.2[114] Flaw reporting procedures	Yes	No
	ALC_DEL.1 Delivery procedures	No	No
	ALC_DVS.1 Identification of security measures	No	No
	ALC_LCD.1 Developer defined life-cycle model	No	No
	ALC_TAT.1 Well-defined development tools	No	No
	ASE_CCL.1 Conformance claims	No	No
	ASE_ECD.1 Extended components definition	No	No

[114] Augmented by developer

Assurance class	Assurance component	augmented?	refined?
ASE: Security Target evaluation	ASE_INT.1 ST introduction	No	No
	ASE_OBJ.2 Security objectives	No	No
	ASE_REQ.2 Derived security requirements	No	No
	ASE_SPD.1 Security problem definition	No	No
	ASE_TSS.1 TOE summary specification	No	No
ATE: Tests	ATE_COV.2 Analysis of coverage	No	No
	ATE_DPT.1 Testing: basic design	No	No
	ATE_FUN.1 Functional testing	No	No
	ATE_IND.2 Independent testing - sample	No	No
AVA: Vulnerability assessment	AVA_VAN.4[115] Methodical vulnerability analysis	Yes	No

Table 6-1 – TOE Security Assurance Requirements (EAL4 augmented)

6.2.1 Refinement of ADV_ARC.1 (Security architecture description)

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

Refinement:

The security architecture description shall describe domain separation in terms of red-black separation. This red-black separation shall describe that the TOE physically or logically separates the interfaces for red data and black data. Further, the security architecture description shall describe that the data output is disabled while performing (1) key generation and manual key entry for the communication through this data port, (2) self-tests, (3) software loading and key destruction.

The security architecture description shall describe domain separation in terms of a semiformal Finite state model. The Finite state model of the TOE shall describe at least the following modes:

(1) Power on/off modes

[115] Augmented by developer.

(2) Crypto officer modes

(3) Key Management modes[116]

(4) Crypto User modes[117]

(5) Self-test modes

(6) Error modes

(7) Bypass modes, if any

(8) Maintenance modes, if TOE provides maintenance functionality.

The Finite state model of the TOE shall describe the mode transition in terms of the input and internal events and internal conditions that cause transitions from one mode to another and the output events resulting from transitions from one mode to another.

The security architecture description shall describe that the data output interface is inhibited when the TOE is in an error mode or in self- test mode.

If any bypass mode exists, the finite state model shall demonstrate that for all transitions into any bypass mode, two independent internal actions are required for the transition into each bypass mode to prevent the inadvertent bypass of plaintext data due to a single error (e.g., two different software or hardware flags are set, one of which may be user- initiated).

If any bypass mode exists, the security architecture description shall demonstrate that functions solely intended for bypass are not executable in any other mode of operation.

If any maintenance mode exists, the security architecture shall demonstrate that the mode transition entering or exiting maintenance mode shall destruct all plaintext secret and private keys and unprotected CSPs.

If any maintenance mode exists, the security architecture description shall demonstrate that functions solely intended for maintenance are not executable in any other mode of operation.

Application note: The TOE does not support bypass modes. The term “finite state model” describes finite set of states related to the modes of operation of the cryptographic module, and the state transition in the model in terms of internal actions and conditions for changing the modes of operation of the cryptographic module. The term “mode” for the states in the model is used according to the mode addressed in FDP_ACC.2/Mode_Trans and FDP_ACF.1/Mode_Trans.

ADV_ARC.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement:

The evaluator shall confirm that the security architecture for the red- black separation and the finite state model are consistent with the TSF presentation in the functional specification, TOE design, TSF implementation, guidance documentation, and evaluator tests.

6.2.2 Refinement of ADV_FSP.4 (Complete functional specification)

ADV_FSP.4.2C The functional specification shall describe the purpose and method of use for all TSFI.

[116] updated from Key/CSP Entry Mode in Protection Profile – BSI-CC-PP-0078.
[117] updated from user mode to reflect renaming of User to Crypto User as renamed throughout ST from Protection Profile – BSI-CC-PP-0078.

Refinement:

The functional specification shall identify the logical interfaces and physical ports as of the following types ("input" and "output" are indicated from the perspective of the module):

- **Data input interface/port: All data (except control data entered via the control input interface) that is input to and processed by the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and status information from another entities),**
- **Data output interface/port: All data (except status data output via the status output interface) that is output from the cryptographic module (including plaintext data, ciphertext data, cryptographic keys and CSPs, authentication data, and control information for another entity). All data output via the data output interface shall be inhibited when the TOE is in an error mode or in start-up (power on) self-test mode,**
- **Control input interface/port: All input commands, signals, and control data (including function calls and manual controls such as switches, buttons, and keyboards) used to control the operation of a cryptographic module shall enter via the "control input" interface.**
- **Status output interface/port: All output signals, indicators, and status data (including return codes and physical indicators such as Light Emitting Diodes and displays) used to indicate the status of a cryptographic module,**
- **Power interface/port: all external electrical power supply.**

The *functional specification* shall describe the interface indicating the selection of an Endorsed mode of operation and the interfaces for user data and TSF data as Endorsed modes of operation.

ADV_FSP.4.3C The functional specification shall identify and describe all parameters associated with each TSFI.

Refinement:

The functional specification shall also specify at minimum the normal voltage and temperature operating ranges of the cryptographic module.

Application note: Please take note, the TOE shall separate logically the interfaces for red data, black data, CSP (including plaintext cryptographic keys and cryptographic key components) and administrative functions according to refinements of ADV_ARC.1. The functional specification shall describe this logical separation according to the content and presentation elements of ADV_FSP.4.

6.2.3 Refinement of ADV_IMP.1 (Implementation representation of the TSF)

ADV_IMP.1.1C The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.

Refinement:

The implementation representation for all software and firmware of the TOE shall be done in a high-level language. The exceptional limited usage of low-level language (e.g., assembly language or microcode) is allowed if essential to the performance of the TOE or when a high-level language is not available. The implementation representation for all hardware components of the TOE within the cryptographic module shall be done in a high-level specification language.

6.2.4 Refinement of ADV_TDS.3 (Basic modular design)

ADV_TDS.3.3C The design shall identify all subsystems of the TSF.

Refinement:

The design shall identify all subsystems with ports or interfaces used for the import or export of secret keys, private keys or key components. These subsystems and all subsystem which transfer or store any secret keys, private keys or key components shall be SFR-enforcing.

ADV_TDS.3.6C The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

Refinement:

When doing so, the design shall identify all modules with ports or interfaces used for the import or export of secret keys, private keys or key components. These modules and all modules which transfer or store any secret keys, private keys or key components shall be SFR-enforcing.

6.2.5 Refinement of AGD_OPE.1 (Operational user guidance)

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

Refinement:

The operational user guidance shall describe how the user is able to determine when an Endorsed mode of operation is selected and what the current status of the cryptographic module is.

The operational user guidance shall describe how the user is able to initiate self-tests as specified in FPT_TST.1 and how the user is informed about the result of these self-tests.

For some situations it is possible that some events cannot be automatically generated in the audit records. This is usually due to the audit functions not being operational at the time these events occur. Such events shall be documented in the operational user guidance, along with recommendation on how manual auditing should be established to cover these events.

If the cryptographic module does not contain the reference authentication data required to authenticate the user for the first time the module is accessed, then other authorised methods (e.g., procedural controls or use of factory-set or default authentication data) shall be described in the operational user guidance, how to control access to the module and initialize the authentication mechanisms.

6.2.6 ALC_FLR.2 Flaw reporting procedures[118]

Dependencies: No dependencies.

Objectives

In order for the developer to be able to act appropriately upon security flaw reports from TOE users, and to know to whom to send corrective fixes, TOE users need to understand how to submit security flaw reports to the developer. Flaw remediation guidance from the developer to the TOE user ensures that TOE users are aware of this important information.

Developer action elements:

ALC_FLR.2.1D The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3D The developer shall provide flaw remediation guidance addressed to TOE users.

Content and presentation of evidence elements:

ALC_FLR.2.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6C The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

Evaluator action elements:

ALC_FLR.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

[118] Augmented by developer

6.3 IT Security Requirements Rationale

Table 8-2 in shows the necessity of the Security Functional Requirements.

6.4 Appropriateness of Assurance Requirements

The assurance requirements chosen for the TOE, EAL 4 augmented by ALC_FLR.2 are considered to be appropriate for the TOE in its assumed (and intended) operating environment for the following reasons:

(1) There are specific customer requirements for Certification Authority (CA) or Trust Service Provider (TSP) components that meet the EAL 4 assurance requirements. The TOE, as part of a larger CA or CSP system, must meet the EAL 4 requirements at a minimum, but does not need to exceed them.

(2) Because the CA and TSP systems, for example, are critical infrastructure systems, customers require a relatively high level of assurance that the components that make them up have been developed and are maintained using sound engineering security practices.

(3) It is assumed that, for most of its life-cycle, the TOE will be contained within a larger secure environment. It will, therefore, not be exposed to a threat environment that allows easy access by highly capable outsiders. The main exception to this is when it is in transit when it will be in a state that is either zeroised or where all of its sensitive data will be encrypted using TDES encryption. Thus, the assumption of moderate attack potential for outsiders is considered appropriate.

(4) Although the TOE will normally be contained within a secure environment, the potential value of the key material stored within the TOE may be sufficient to result in insider attacks. Because insiders would typically have access to the TOE or components of it and would be likely to have detailed knowledge of the TOE and its configuration in their environment, they are considered to have High attack potential.

(5) The augmentation of including ALC_FLR.2 is in response to existing company practice that has been implemented to meet customer requirements for flaw reporting and fixing.

(6) The augmentation of AVA_VAN.4 is consistent with the level expected of HSM deployed in a secured environment.

7 TOE Summary Specification

7.1 Overview

The TOE is used primarily as a Hardware Security Module (HSM) for the protection of the private signing keys at the Certification Authority (CA), or Trust Service Provider (TSP), within a Public Key Infrastructure (PKI). As such, its primary functions are to securely generate and protect the private signing key used by the CA or TSP when signing digital certificates.

The Luna® PCI-E cryptographic module provides storage capability for cryptographic material generated by the module or generated by the host application as well as storage for non-cryptographic data provided by the host application. Non-cryptographic data can be stored in the form of certificate objects or in the form of data objects. When storing or generating keys (secret or private), the module imposes some restrictions on how these keys are handled. Security policy enforcement is described in more detail in section 7.2.

7.1.1 Object Model

All user data is managed by the module as objects. Objects are owned by external processes/users and manipulated by the module. They are characterized by different attributes used by the module to determine the handling rules to be applied. The module provides two ways of storing objects: permanent (also known as PKCS #11 token objects) and volatile (also known as session objects). Permanent objects are kept inside the module even when no power is applied to it. They are stored encrypted in a flash memory device. Session objects only exist when power is applied to the module and they are stored in volatile RAM.

The Luna® object model is very closely related to the PKCS#11 standard. More details on the Luna® interface are provided in the Luna® Interface Control Document [6].

7.1.2 Multi-Session Capability

The Luna® PCI-E cryptographic module manages communication with external processes on a per session basis. Applications running on the host system requiring data and cryptographic services from the module have to open a session with the module before gaining access to the module’s functions and objects. The session provides a logical connection between the application and the module and it is the session to which the authentication state is bound. It is possible for an application to open multiple sessions with the module or to have multiple applications each opening different sessions with the module.

The module provides a higher level of connection abstraction based on an Access ID that associates a group of sessions to a particular application. This approach allows an application or applications to share sessions, and associated authentication state, within the scope of that Access ID.

7.1.3 TOE Roles

The following roles are supported by the TOE:

- **Security Officer (SO)** – authorised to install and configure the TOE, set and maintain security policies, and create and delete users (Crypto Officer and Crypto User roles). The TOE can have only one SO.
- **Crypto Officer (CO)** – authorised to create, use, destroy and backup/restore cryptographic objects.
- **Crypto User** – authorised to use cryptographic objects (e.g., sign, encrypt/decrypt).

The Crypto Officer and Crypto User interface to the Luna® PCI-E for cryptographic operations using the supplied non-TOE software. Utility software is also provided to enable easier access to commonly used functionality. The Security Officer uses a separate Command Line Interface (CLI), which is part of the interface software, to perform configuration, security policy settings and user creation/deletion. The CLI is also used by the Crypto Officer to perform backup and restoration of cryptographic objects.

The TOE allows for the creation of multiple users in the Crypto Officer and Crypto User roles. Each user is created within a cryptographically separated partition in the Luna® PCI-E cryptographic module and each partition must have one and only one user in the Crypto Officer role. A partition may also have one and only one user in the Crypto User role.

7.1.4 Multi-User Capability

A user must access the module through a session. Sessions are initially opened as Public sessions and may remain Public or become Private (authenticated) following a successful user authentication. Session states are kept separate based on the user authentication state stored by the module. The module allows multiple user identities to be authenticated at a time. Once authenticated, a session becomes bound to the user identity and has access to all cryptographic operations appropriate to the user’s role and may access private objects generated on behalf of the user in previous sessions. Although there may be many users authenticated to the cryptographic module, there is effectively only one thread of execution within the module and, therefore, only one command being executed from request through to response at any given time.

7.2 Capability and Policy Settings

The Luna® PCI-E was designed with the flexibility needed to support a number of different product variants. The main method used to control the behaviour of different products is a fixed set of “capabilities” set at the factory. The settings made for the TOE configuration are shown in sub-sections 7.2.1 and 7.2.2. For each of the capabilities, a corresponding policy element exists. The SO establishes the policy that will govern the cryptographic module’s operation, according to the requirements of the customer organization, by enabling/disabling or refining the corresponding policy elements to equate to or to be more restrictive than the pre-assigned capabilities.

Policy set elements can only refine capability set elements to more restrictive values. Specifically, if a capability is set to allow, the corresponding policy element may be set to either enable or disable. However, if a capability is set to disallow, the corresponding policy element is set to disabled and is not SO-configurable. Thus, an SO cannot use policy configuration to lift a restriction set in a capability definition.

There are also several elements of the cryptographic module’s behaviour that cannot be changed, these specific elements are the following:

- Non-sensitive secret keys are not allowed.
- Non-sensitive private keys are not allowed.
- Non-private (Public) secret keys are not allowed.
- Non-private (Public) private keys are not allowed.
- Creation of secret keys and private keys through the PKCS #11 create object interface is not allowed. That is, the API cannot be used to create keys by passing in known plaintext values.

In the next two sub-sections, all capability elements described as “enable some functionality” are Boolean values where false (or zero) equates to disallow the functionality and true (or one) equates to allow the functionality. The remainder of the elements are integer values with either the default value or the maximum in number of bits shown.

Except as noted, all Boolean capabilities are allowed (i.e. configurable by the SO).

7.2.1 Cryptographic Module Level Capabilities

The following is the set of capabilities supported at the module level and relevant to the scope of the CC certification:

- 0: Enable PIN-based authentication **(disallowed in CC configuration – must be disabled)**
- 1: Enable PED-based authentication **(required in CC configuration – must be enabled)**
- 6: Enable masking **(disallowed in CC configuration – must be disabled)**
- 7: Enable cloning
- 12: Enable non-FIPS algorithms **(disallowed in CC configuration – must be disabled)**
- 15: Enable SO reset of partition PIN
- 16: Enable network replication
- 17: Enable Korean Algorithms **(disallowed in CC configuration – must be disabled)**
- 20: Enable Remote Authentication **(disallowed in CC configuration – must be disabled)**
- 21: Enable forcing user PIN change
- 22: Enable offboard storage
- 23: Enable partition groups **(disallowed in CC configuration – must be disabled)**
- 25: Enable remote PED usage
- 26: Enable External Storage of MTK Split
- 28: Enable HA mode CGX **(disallowed in CC configuration – must be disabled)**
- 29: Enable Acceleration **(must be enabled in CC configuration)**
- 30: Enable unmasking **(disallowed in CC configuration – must be disabled)**
- 31: Enable FW5 compatibility mode
- 34: Enable ECIES support.

7.2.2 Partition Level Capabilities

The following is the set of capabilities are supported at the partition level.

- 0: Allow private key cloning
- 1: Allow private key wrapping
- 2: Allow private key unwrapping

- 3: Allow private key masking **(disallowed in CC configuration – must be disabled)**
- 4: Allow secret key cloning
- 5: Allow secret key wrapping
- 6: Allow secret key unwrapping
- 7: Allow secret key masking **(disallowed in CC configuration – must be disabled)**
- 10: Allow multipurpose keys
- 11: Allow changing key attributes
- 14: Challenge for authentication not needed **(disallowed in CC configuration – must be disabled)**
- 15: Ignore failed challenge responses **(disallowed in CC configuration – must be disabled)**
- 16: Operate without RSA blinding
- 17: Allow signing with non-local keys
- 18: Allow raw RSA operations
- 20: Max failed user logins allowed
- 21: Allow high availability recovery
- 22: Allow activation
- 23: Allow auto-activation
- 25: Minimum pin length
- 26: Maximum pin length
- 28: Allow Key Management Functions
- 30: Allow Remote Authentication **(disallowed in CC configuration – must be disabled)**
- 31: Allow private key unmasking **(disallowed in CC configuration – must be disabled)**
- 32: Allow secret key unmasking **(disallowed in CC configuration – must be disabled)**

The following capabilities are only configurable at the partition level if cloning is allowed and enabled at the cryptographic module level:

- 0: Allow private key cloning
- 4: Allow secret key cloning

The following capabilities are only configurable if masking is allowed and enabled at the cryptographic module level:

- 3: Allow private key masking **(disallowed in CC configuration – must be disabled)**
- 7: Allow secret key masking **(disallowed in CC configuration – must be disabled)**
- 31: Allow private key unmasking **(disallowed in CC configuration – must be disabled)**
- 32: Allow secret key unmasking **(disallowed in CC configuration – must be disabled)**

7.3 IT Security Functions

7.3.1 User Identification and Authentication

A user is defined as an entity that acts to perform an operation on the TOE. In most instances, this will be a host application program such as a PKI Certification Authority implementation. The TOE supports three user roles: Security Officer (SO), Crypto Officer (CO), and Crypto User (CU). For a user to assume any role the module enforces user identification and authentication.

The TOE requires that all users (SO, CO and CU) be authenticated by proving knowledge of a secret shared by the user and the cryptographic module. When the module is first powered on, no users are authenticated and actions are limited to those stated in section 7.3.1.1.

The TOE generates the authentication secrets using its DRBG. For the SO, the authentication secret is a 48-byte random secret and it is generated at the time the cryptographic module is initialised. For Users, the authentication secrets consist of a 48-byte random secret and separate challenge secret; these are generated at the time the partition is created by the SO. The authentication secret(s) are provided to the operator via the trusted PIN entry device and must be entered by the operator via the trusted PIN entry device and via a logically separate trusted channel (in the case of the response based on the challenge secret) during the login process. Both the Crypto Officer and the Crypto User use the same 48-byte random secret. If a Partition is created with Crypto Officer and Crypto User roles, a separate challenge secret is generated for each role.

SO authentication requires the transmission to the cryptographic module of separate credentials via a trusted PIN entry device.

User authentication is a two-stage process. The first stage is termed “Activation” and is performed using a locally attached trusted PIN entry device or optionally for the CO and CU roles a remote trusted PIN entry device.

To proceed with the activation stage, the login command must be sent to the host via a command line command issued over the local serial or remote connection. Parameters passed to the command also specify the identity of the user that the operator wants to assume.

Once Activation has been performed by an authenticated Crypto Officer, the partition data is ready for use within the cryptographic module. Access to key material and cryptographic services, however, is not allowed until the second stage of authentication, equivalent to “User Login”, has been performed. This requires the input of a partition’s challenge secret as part of an application program’s login operation.

The authentication challenge secret (or secrets if both the Crypto Officer and Crypto User roles are used) for the partition is generated by the cryptographic module as a random 75-bit value that is transmitted as a 16-character string intended to be displayed on the trusted PIN entry device. The challenge secret is then provided, via a secure out-of-band means, to each external entity authorised to connect to the partition and is used by the external entity to form the response to a random one-time challenge from the cryptographic module. The encrypted one-time response is returned to the cryptographic module where it is verified to confirm the “User Login”.

For the purposes of the scope of this certification, only a single step of each login process is considered being: (1) PED key presentation for the SO via local PED and (2) Challenge-Response mechanism for the CO and CU. Further, in the CC approved configuration as covered in [24], all SO login alongside initialization of roles must be performed with a Trusted PIN Entry Device connected directly to the PCI-E card.

The TOE enforces a maximum login attempts policy. This feature serves to prevent an exhaustive search approach to find the authentication data of the SO, CO or CU. The implementation of this feature differs for an SO authentication data search and a User authentication data search.

In the case of a CO:

> If “y” consecutive CO logon attempts fail (“y” is defined by the SO in the configurable policy for the partition), the TOE will either lock the partition or erase the partition, as defined by the SO in the configurable policy. If it has been locked, the partition must be unlocked by the SO in order to allow user login. If it has been erased, the partition cannot be recovered directly. If recovery is required, the SO must create a new partition and the new Crypto Officer must recover the partition’s data from a backup module.

In the case of a CU:

> If “y” consecutive CU logon attempts fail (“y” is defined by the SO in the configurable policy for the partition), the TOE lock the partition, as defined by the SO in the configurable policy. If it has been locked, the partition must be unlocked by the CO in order to allow CU login.

In the case of the Security Officer:

> If three (3) consecutive SO logon attempts fail, the module is zeroised and must be re-initialised.

Following a successful login, the user is bound to the subject acting on its behalf by having the User Authorization Vector (UAV) data included in the state data maintained by the session manager. In the case of the Luna® PCI-E cryptographic module the subject acting on behalf of a user is a session. The relationship between the user and the session is discussed in more detail in section 7.1.4 and the data contained in the UAV is described in section 7.3.3.

It is not possible for a user to be bound to more than one role simultaneously. To assume a different role, the user must log out and log in as the new role. Subsequent authentication requests will be subject to the same authentication checks (i.e., Activation and User Login).

Authentication state information is not persistent across power cycles or reboot operations.

7.3.1.1 Unidentified and Unauthenticated Users

The TOE allows the following actions on behalf of the user to be performed before the user is identified:

- Get (container info, HSM/container capability set, HSM/container policy set, all accesses, mechanism info, handle, OUID, container status)
- Query container configuration
- Query container object identify (from known OUID or object handle)
- Session manager functions (open, close all, open/close/clean access)
- Self-test
- HSM deactivation
- PED (connect, disconnect, configuration)
- MTK functions (unlock challenge / response, restore, resplit, zeroize)
- Retrieve license list and query license details
- Perform Host-to-HSM communication channel tests;
- Query log status and submit external log messages for addition to secure audit log;
- Submit data to be tunneled from Host-to-Luna PCI-E external USB interface;

The user must be identified before any other TSF-mediated action is allowed to proceed.

The TOE allows the following actions on behalf of the user to be performed before the user is authenticated:

- *[All operations permitted prior to identification (above)];*
- Zeroize
- Login against a container or request login against a paired HA container.
- Object handler functions (create, modify, destroy, and get attributes…)
- Get Random
- Session manager functions (insert / extract context, close, get session info)
- Digest functions
- Request a challenge

The user must be authenticated before any other TSF-mediated action is allowed to proceed.

7.3.2 Authentication Data Selection

The User authentication data is a 48-byte value that is randomly generated by the module and stored on a compatible token (e.g. SafeNets iKey) plus the Crypto Officer and Crypto User Challenge Secrets, which are initially provided to the user via the trusted PIN entry device. . The TOE allows for user in the role of either Crypto Officer or Security Officer to change their respective authentication data.

7.3.3 User Account Data

The Security Officer is the only role allowed to create users, modify user status and delete users. The TOE maintains a user's account data in a User Authorization Vector (UAV) that is stored in memory reserved for the TOE's use. The UAV includes the following data:

- User ID number
- User checkword
- User function vector
- User failed login count
- User "lockout" status

The User checkword contains the User's secret key, Crypto Officer and Crypto User Challenge Secrets, and a validation string encrypted using a key derived from the User's authentication data. The secret key is randomly generated by the module at the time the User is created and is used to encrypt a User's objects on the module. The validation string is a known byte string used to verify that the checkword has been decrypted correctly. The checkword, itself, is stored encrypted using a key derived from the authentication data.

7.3.4 Access Control

The TOE enforces an identity-based access control policy that applies to all objects on the module, in particular to private key and secret key objects, and governs a subject's access to an object using the following operations:

- Read (Query Attribute Value)
- Modify
- Destroy
- Generate[119]
- Wrap (export of Secret Keys)
- Use[120]
- Clone

[119] The Generate operation is intended primarily to indicate symmetric key or asymmetric key pair generation. However, it also includes other methods of creating an object in the TOE, such as importing (unwrapping) a key and generic data object creation.

[120] The Use operation includes symmetric key encryption/decryption, private key signing and decryption, and public key verification and encryption.

A subject's access to objects stored on the module is mediated on the basis of the following subject and object attributes:

- Subject attributes:
  - Session and Access ID
  - User ID associated with session (Access Owner)
  - Role.
- Object attributes:
  - Private. If True, object is Private. If False, object is Public.
  - Owner. Object ownership is assigned to the object creator, if the object is Private. Public objects are not owned by a user. Ownership is enforced by user identity and internal key management.
  - Sensitive. If True, object is Sensitive. If False, object is Non-Sensitive.
  - Extractable. If True, object may be extracted. If False, object may not be extracted.
  - Modifiable. If True, object may be modified. If False, object may not be modified.

Private data objects are labeled with a number corresponding to their owner and sensitive attributes are encrypted using the owner's secret key. Private data objects are only accessible by the object owner. Public data objects may be accessed by any user with an active session on the module. Secret key and private key objects are always created as Private, Sensitive objects and can only be used for cryptographic operations by a logged in User. Only data and certificate objects can be non-sensitive. Secret key objects that are marked as extractable may be exported from the module using the Wrap operation.

The module does not allow any granularity of access other than owner or public (i.e., a Private data object cannot be accessible by two users and restricted to other users). Ownership of an object gives the owner access to the object through the allowed operations but does not allow the owner to assign a subset of rights to other users. Allowed operations are those permitted by the configurable policy settings in sections 7.2.1 and 7.2.2.

7.3.5 Object Reuse

The TOE enforces an object reuse policy in that every object is allocated its own portion of memory (flash or volatile RAM). Permanent objects (stored in flash) are maintained in an encrypted state at all times, and their information content is, therefore, never available except when decrypted for use in volatile memory within the TSF boundary. The policy also ensures that no permanent object is placed in a previously allocated memory location unless all previous memory content is purged and zeroised. When cryptographic functions are performed, a cryptographic context is created to hold data required by the function (e.g., an AES key schedule for an AES function). The cryptographic context only exists in volatile RAM memory and is not accessible to any functions except those defined by its owner function. The memory assigned to a cryptographic context is always purged of its content before being handed over to another function. Direct access to either volatile or flash memory locations is never provided to users; all user interaction with the objects within the module is via memory handles.

7.3.6 Data Authentication

The TOE provides data authentication at two different levels. At the first level, the TOE calculates the SHA-1 fingerprint of each object it stores and the user may query the value of the fingerprint at any time. This allows the user to verify the continuing validity of the object.

At the second level, the TOE will generate evidence of the validity of a private key and its corresponding public key in a special digitally signed certificate format, known as a Public Key Confirmation. The signature is performed using a private key that is either generated by SafeNet specifically for this purpose and whose public key certificate has been signed by the SafeNet trust anchor or generated by a customer organization and whose public key certificate has been signed by a third-party CSP or Trust Centre. The Public Key Confirmation permits a user to verify the validity of an asymmetric key pair, verify that the TOE generated it and identify the trusted third party providing the guarantee of validity and origin.

7.3.7 Key Export and Import Protection

Secret (Symmetric keys) may only be exported from the TOE boundary in a wrapped (encrypted) form if the Extractable attribute is True. This feature is supported on all TOE configurations, whether or not cloning (backup) is supported. Private keys (Asymmetric) may be exported from the TOE boundary when cloned between HSMs – using the cloning (backup) configuration, or wrapped when private key wrapping is enabled. If the Extractable attribute is False, the key may not be exported from the module boundary under any condition.

Objects may be imported into the module under the control of the Access Control policy. Secret keys and/or private keys generated in the host IT environment may only be imported into the module by an unwrapping operation on the module. Unwrapped keys have their Sensitive attribute set to True by the TOE. The configurable policy for a partition may also be set to prohibit the use of externally generated private keys for signing operations.

Wrapping and unwrapping of key material between the TOE and other entities can only take place if prior agreement has been reached regarding the key to be used for the wrap and unwrap operations. This can either be through key sharing of a secret key for use with a symmetric encryption algorithm or through the use of the public key of the intended recipient with an asymmetric encryption algorithm.

7.3.8 Cryptographic Material Management

Cryptographic material (key) management functions protect the confidentiality of key material throughout its life-cycle. The following functions are provided by the module to support this:

(1) Deterministic Random Bit Generator (DRBG) based on CTR-DRBG with AES256 from NIST SP 800-90 (section 10.2.1) as used for all cryptographic strength entropy.

(2) Cryptographic key generation in accordance with the following indicated standards:
   a. RSA 1024, 2048, 3072 and 4096 bits key pairs in accordance with ANSI X9.31 and FIPS PUB 186-3[121]
   b. TDES 112 bits and 168 bit keys (NIST SP 800-67).
   c. AES 128, 192, 256 bit keys (FIPS PUB 197).
   d. DSA 1024, 2048 and 3072 bit key pairs in accordance with FIPS PUB 186-3.
   e. Elliptic Curve key pairs (curves P-192, P-224, P-256, P-384, P-521, K-163, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409 and B-571 in accordance with SP 800-57) in accordance with FIPS PUB 186-3.

(3) EC Diffie-Hellman (ECDH) (curves P-192, P-224, P-256, P-384, P-521, K-163, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409 and B-571 in accordance with SP 800-57) key establishment in accordance with NIST SP 800-56A.

[121] FIPS PUB 186-3 only allows 1024, 2048 and 3072-bit RSA keys

(4) Symmetric key wrap / unwrap: TDES 168 bits and AES 128, 192, and 256 bits in accordance with PKCS #11 (key transport provides 112 bits of security strength with TDES and between 128 and 256 bits of security strength with AES).

(5) Asymmetric key wrap / unwrap: RSA 1024 – 4096 (RSAES-OAEP from PKCS #1 v2.1) (key transport provides between 80 and 152 bits of security strength).

(6) Encrypted key storage (using AES 256 bit encryption and key access following the PKCS #11 standard.

(7) Derivation of keys using Counter Mode KDF with AES or TDES from SP 800-108.

(8) Destruction of cryptographic keys using one of three ways as described below:

   a. An object on a Luna® cryptographic module that is destroyed using the PKCS #11 function C_DestroyObject is marked invalid and remains encrypted with the Partition User's key or a Luna® cryptographic module's general secret key until such time as its memory locations (flash or RAM) are re-allocated for additional data on a Luna® cryptographic module, at which time they are purged and zeroised before re-allocation.

   b. Objects on a Luna® cryptographic module that are destroyed as a result of authentication failure are zeroised (all flash blocks in the Partition User's memory turned to 1's). If it is an SO authentication failure, all flash blocks used for key and data storage on a Luna® cryptographic module are zeroised.

   c. Objects on a Luna® cryptographic module that are destroyed through C_InitToken (the SO-accessible command to initialize a Luna® cryptographic module available through the API) are zeroised, along with the rest of the flash memory being used by the SO and Partition Users.

7.3.8.1 Key Storage and Access Protection

Keys are always stored as secret key or private key objects with the Sensitive attribute set. The key value is, therefore, stored in encrypted form using the owning Partition User's Storage Key (USK) and the Master Tamper Key (MTK) stored in the battery-backed RAM. Access to keys is never provided directly to a calling application. A handle to a particular key is returned that can be used by the application in subsequent calls to perform cryptographic operations.

Private key and secret key objects may be imported into a module using the Unwrap or Derive operation under the control of the Access Control Policy. Any externally-set attributes of keys imported in this way are ignored by a module and their attributes are set by a module to values required by the Access Control Policy.

7.3.8.2 Key Cloning

Key cloning is a Luna® product feature that uses a one-time AES-256 key as a session key to encrypt an object being transferred from one Luna® module to another. Objects transferred using the cloning protocol may be keys, user data, or module data. The AES session encrypting key is obtained by combining the 24 byte cloning domain value (randomly generated by the module) with random one-time data generated by source and target modules and exchanged using RSA 4096-based transport.

7.3.8.3 Key Wrap / Unwrap

The key wrap operation encrypts a key value for output, using either an RSA public key (only if wrapping a symmetric key) or a symmetric key to wrap another symmetric key.

The unwrap operation takes as input an encrypted key value and a handle to the key that was originally used to do the wrapping. It decrypts the key value, stores it in the module as a key object and returns the handle to the imported key.

Note that for both wrap and unwrap operations, the user (or calling application acting on the user's behalf) never has access to the actual key values – only handles assigned to the key objects in the module.

Keys and key components imported and exported through the use of wrapping/unwrapping operations are validated during the wrapping/unwrapping operation and protected from modification through the use of the specified encryption method.

7.3.9 Cryptography

Because of its generic nature, the module's cryptographic co-processor and firmware support a wide range of cryptographic algorithms and mechanisms. The TOE supports the following cryptographic functions:

- Signature Generation and Validation:
    - RSA with a supported modulus length between 1024-4096 bits (RSASA-PSS from PKCS #1 v2.1, FIPS PUB 186-3) and with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (FIPS PUB 186-3).
    - RSA with modulus length between 1024-4096 bits (FIPS PUB 186-3 and ANSI X9.31) and with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (FIPS PUB 186-3).
    - DSA signature generation and validation with supported modulus length 1024, 2048 and 3072 bits with SHA-1, SHA-256, SHA-384, SHA 512 (FIPS PUB 186-3).
    - ECDSA signature generation and validation curves P-192, P-224, P-256, P-384, P-521, K-163, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409 and B-571 with SHA-1, SHA-256, SHA-384, SHA 512 (FIPS PUB 186-3).
- Message Digest:
    - Hash generation SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (FIPS PUB 180-4).
- Data Encryption and Decryption:
    - AES in ECB, CBC or GCM modes with key sizes 128, 192 and 256 bits (FIPS PUB 197).
    - Triple DES in ECB, CBC or CFB-8 modes with key sizes 112 and 168 bits (NIST SP 800-67)
    - RSA with a supported modulus length between 1024-4096 bits (RSAES-OAEP from PKCS #1 v2.1)
- Deterministic Random Number Generation:
    - CTR-DRBG using AES256 (NIST SP 800-90, Section 10.2.1).
- Message Authentication Code Generation and Validation:
    - HMAC using SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (FIPS PUB 198-1).
    - TDES MAC (FIPS PUB 113).
    - AES CMAC with 128, 192 and 256 bit keys (NIST SP 800-38B).
    - TDES CMAC with 112 and 168 bit keys (NIST SP 800-38B).
    - AES GMAC with 128, 192 and 256 bit keys (NIST SP 800-38D).

- Key Derivation:
    a. Counter Mode KDF and either AES or TDES (NIST SP800-108).

7.3.10 Data Exchange

The TOE provides security functions that support secure data exchange in two main ways:

- Data integrity and authenticity is protected through the use of RSA and DSA digital signatures. The digital signature of the data object provides evidence of data validity. The TOE provides logged in Users the ability to generate evidence in the form of a digital signature provided they have access to the private signing key and to verify the evidence and the identity of the originator who generated the evidence provided they have possession of the digitally signed information and access to the signer's verification public key.
- Data confidentiality is protected through the use of symmetric and/or asymmetric encryption/decryption of user data and in the Wrapping and Unwrapping operations.

7.3.11 Specification of Security Management Functions

The TOE provides the following security management functions:

- disable, enable and modify the behaviour of configurable policy settings at the HSM and Partition levels (FMT_MOF.1),
- change_default, query, modify and delete the security attributes User Locked Flag,
- modify the security attributes UAV – Checkword,
- change_default and delete the security attributes User ID and UAV – Checkword,
- change_default and modify the security attributes SOV – Checkword,
- modify the security attributes CKA_PRIVATE (for data and certificate objects only), CKA_EXTRACTABLE (for secret keys only), CKA_DERIVE (for secret keys only) and CKA_MODIFIABLE,
- change_default the Number of User Login Failures Allowed.

Details of these management capabilities are provided in sections 7.3.12 and 7.3.13.

7.3.12 Security Function Management

The TOE provides security management capabilities for the Security Officer (SO) to disable, enable and modify the behaviour of the functions listed below.

- The set of policies supported at the HSM level are defined in section 7.2.1.
- The set of policies supported at the partition level are defined in section 7.2.2.

7.3.13 Security Data Management

The TOE allows the Security Officer and the Crypto Officer to manipulate security-relevant data stored on the module. Specifically, it allows only the Security Officer to change the default values of the settings listed below:

- Number of failed Partition User logins allowed before partition is locked out/cleared. (Default is 10, SO can configure to be 3 <= N <= 10)

The User Authorization Vector, described in section 7.3.3, is the data structure used by the module to store the user's security attributes. The TOE restricts the ability to manipulate the UAV data as described below:

- Only the Security Officer role can change_default, query, modify and delete the UserLockedFlag.
- Only the Security Officer role can change_default and delete the:
  - UserID.
  - Checkword, which includes the user secret key plus a fixed value used for authentication in encrypted form.
- Only the Security Officer and User roles can modify the Checkword (for the SO or applicable User ID).

The Access Control policy also restricts the ability to modify, the security attributes CKA_PRIVATE (for data and certificate objects only), CKA_EXTRACTABLE (for secret keys only), CKA_DERIVE (for secret keys only) and CKA_MODIFIABLE to the Crypto Officer role.

The TOE assigns default attributes to objects as they are created. The creator of the object may specify values different from the defaults with the exceptions described below.

There are security-relevant object attributes that are set to restrictive default values that cannot be changed by anyone. These attributes and their settings are the following:

- The CKA_SENSITIVE attribute is set TRUE for all secret and private key objects.
- The CKA_EXTRACTABLE attribute is set FALSE for all private key objects.

7.3.14 Logical Self-Protection of Security Functions

The TOE ensures the logical protection of its security functions from attempts to subvert or bypass security enforcement by implementing a number of self-protection measures. The main self-protection features are described below.

7.3.14.1 Memory and Firmware Integrity Check

The Boot Loader provides an integrity check to ensure the integrity of the firmware and to ensure the integrity of any permanent security-critical data stored within the cryptographic module. The firmware integrity is protected by a SHA-1 hash. The firmware's integrity is checked when the firmware is initially loaded or updated and every time the module is started. The module will halt if the firmware integrity is not verified. Similarly, the module's memory is checked for consistency every time the module is started and the module will halt if the memory consistency check fails.

7.3.14.2 Self-Tests

The TOE performs a number of tests of security-critical functions each time it is activated. The TOE offers two categories of self-tests that can be called up by the user at any time: hardware and cryptographic checks. The hardware self-test verifies access to all of the volatile RAM memory. The cryptographic self-tests perform a test of all of the cryptographic algorithms provided by the module. The cryptographic self-tests are based on a known answer test methodology where a known key, or initial configuration, is used to process a known data input and the result obtained is compared to a previously-calculated answer.

7.3.14.3 Prevention of By-pass and Separate Execution Domain

The TOE prevents bypass by ensuring that TSP enforcement functions are invoked and succeed before allowing a subsequent firmware function to proceed. It maintains a separate domain for its own execution that is protected from external agents. It also separates users by encrypting private objects with the user's secret key and by allowing only one thread of execution on the module at any one time and, therefore, allowing only one user's command to be active at any time.

7.3.14.4 Preservation of Secure State

The TOE preserves itself in a secure state in the event of failures detected by the abstract machine test and self-test functions. Behaviour in the event of other failure conditions is described in sub-section 7.3.17.

7.3.15 Cloning

For performance and secure backup purposes, Luna® PCI-E cryptographic modules may be grouped in clusters that are referred to as "domains." A domain is established by generating a 24 byte secret, known as a cloning domain key or cloning domain identifier, on one module (that could be considered to be the "master" for the domain) and transferring the secret securely to other modules that are to be part of the domain. The cloning domain key is then used during the mutual authentication and key agreement exchange that takes place between modules, or between a Luna® PCI-E and a corresponding cryptographic module acting as a backup, as described briefly below. This mutual authentication ensures that the two modules participating in the cloning operation belong to the same cloning domain and can thus participate in the cloning process.

When modules are members of a domain, they must be capable of operating in such a way that they behave as one identical module to the calling application. The cloning function provides the capability to duplicate the cryptographic state of a module by cloning token objects from a source module to a target module within the same cryptographic domain in a cryptographically protected fashion that prevents modification and disclosure.

When cloning is invoked, the cloning protocol protects security-relevant data from disclosure and modification when it is transferred between the TOE and the remote trusted component (i.e. another Luna® PCI-E module). The protocol is designed such that source and target modules both participate in ensuring that objects are all transferred correctly between modules. It also ensures that any data exchanged during the cloning operation cannot be replayed in order to gain unauthorised access to the module. The source module maintains its original state and, therefore, any sort of failure of the cloning function will not result in a loss of use of the original objects.

The cloning protocol implements a mutual authentication mechanism to ensure that both modules are members of the same domain by providing mutual authentication of the two modules. The mechanism uses cryptographic techniques to provide mutual authentication, proof of origin, integrity and confidentiality of the objects being transferred from source to target module within a domain. The key management scheme used within the cloning protocol also protects against replay attacks and minimizes the impact of possible key compromise or modification by ensuring that a unique AES key is used for each cloning operation.

7.3.16 Physical Self-Protection

The module's physical design was developed to limit the effectiveness of passive attacks via conducted and radiated emissions.

7.3.17 Failure Handling

If power is lost to the module for whatever reason, permanent objects (private keys, etc.) are preserved and remain cryptographically protected; session objects are cleared from the module. The module can be placed back into operation without compromise of its functionality or permanently stored data. In case of power failure in the host IT environment, host system restart or other circumstances that do not affect the module's operational capability, the module will ensure continued protection of sensitive material and will permit recovery from the last logged in state.

Data input/output failures would only affect the processing of the current command and, because no PKCS #11 API function returns sensitive plaintext data, there could be no compromise of the user data protection capabilities. Because of the way in which commands are handled, the module would remain in the state it was at the last successful command completion. When data input/output capability is restored the module would resume operation in that state.

7.3.18 Backup and Recovery

As previously described, the module maintains its secure state in the event of a failure. In the event of a catastrophic damage to or failure of the module itself, recovery is accomplished via an included backup module[122] as described below.

The TOE provides the capability to securely backup a module using the cloning function (see Section 7.3.8.2). Because the cloning function securely duplicates all objects from the primary module to the backup module, the backup module allows recovery by cloning the backed up objects to a new module that has been initialised with the same cloning domain. The basic data authentication mechanism described in section 7.3.6 can be used at both the TOE and the backup module before and after cloning operations to ensure the integrity of backed up and restored key objects.

[122] Only applicable to TOE configurations including Backup HSM

8 RATIONALE TABLES

8.1 Introduction

The following tables provide mappings and descriptions of the rationale for the Security Objectives, IT Security Requirements and Dependencies, Assurance Measures and Security Functional Requirements.

- Table 8-1 demonstrates the necessity of the security objectives and their appropriateness in countering the stated threats and providing for the stated assumptions.
- Table 8-2 demonstrates the necessity of the security objectives and their appropriateness in countering the stated threats and providing for the stated assumptions.
- Table 8-3 demonstrates interdependency of Security Functional Requirements with rationale.
- Table 8-4 provides a mapping of IT Security Functions to IT Security Requirements and Security Functional Requirements (SFRs).

Note: A mapping of the Security Objectives Rationale was previously presented in Table 4-1.

8.2 SFR to Objective Mapping

The following table provides the mapping between SFR and Objectives as originating from [Cryptographic Modules, Security Level "Enhanced Basic"] but updated for changes summarised in Section 2.2 and with extended SFR names added for convenience:

Security Functional Requirement	SFR Full Title	O.Endorsed_Crypto	O.I&A	O.Control_Services	O.Control_Keys	O.Roles	O.Keys_Export	O.Key_Generation	O.Key_Import	O.Key_Management	O.Key_Destruction	O.Check_Operation	O.Prevent_Inf_Leakage
FCS_CKM.1	Cryptographic key generation	x						x		x			
FCS_CKM.2/ Private_Key_Import	Cryptographic key distribution – Secret Key Import	x							x	x			
FCS_CKM.2/ Secret_Symmetric_Key_Import	Cryptographic key distribution – Private Key Import	x							x	x			
FCS_CKM.2/Cloning_Import	Cryptographic key distribution – Cloning Import	x							x	x			
FCS_CKM.2/Secret_Key_Export	Cryptographic key distribution – Secret Key Export	x					x			x			
FCS_CKM.2/Secret_Symmetric_Key_Export	Cryptographic key distribution – Secret Symmetric Key Export	x					x			x			
FCS_CKM.2/Cloning_Export	Cryptographic key distribution – Cloning Export	x					x			x			
FCS_CKM.4	Cryptographic key destruction	x								x	x		
FTP_ITC.1/Cloning	Inter-TSF trusted channel - Cloning						x		x	x			
FCS_COP.1/Sign	Cryptographic operation - Digital signature	x											
FCS_COP.1/Digest	Cryptographic operation - Message digest	x											
FCS_COP.1/RSA_Enc_Dec	Cryptographic operation - RSA Encrypt/Decrypt	x											

Security Functional Requirement	SFR Full Title	O.Endorsed_Crypto	O.I&A	O.Control_Services	O.Control_Keys	O.Roles	O.Keys_Export	O.Key_Generation	O.Key_Import	O.Key_Management	O.Key_Destruction	O.Check_Operation	O.Prevent_Inf_Leakage
FCS_COP.1/TDES_Enc_Dec	Cryptographic operation - TDES Encrypt/Decrypt	x											
FCS_COP.1/AES_Enc_Dec	Cryptographic operation - AES Encrypt/Decrypt	x											
FCS_COP.1/Key_Derive	Cryptographic operation – Key Derive	x											
FCS_COP.1/MAC	Cryptographic operation – MAC	x											
FCS_RNG.1	Random number generation	x											
FIA_ATD.1	User attribute definition		x										
FIA_UID.1	Timing of identification		x										
FIA_UAU.1	Timing of authentication		x										
FIA_UAU.6	Re-authenticating		x										
FIA_UAU.7	Protected authentication feedback		x										
FIA_USB.1	User-subject binding		x										
FIA_AFL.1/CO	Authentication failure handling – Crypto Officer		x										
FIA_AFL.1/SO	Authentication failure handling – Security Officer		x										
FIA_AFL.1/User	Authentication failure handling – User		x										
FDP_ACC.2/Key_Man	Complete access control – Key Management.			x	x								
FDP_ACF.1/Key_Man	Security attribute based access control –Key Management.			x	x								
FDP_ACC.2/Oper	Complete access control – Cryptographic Operation.			x									

Security Functional Requirement	SFR Full Title	O.Endorsed_Crypto	O.I&A	O.Control_Services	O.Control_Keys	O.Roles	O.Keys_Export	O.Key_Generation	O.Key_Import	O.Key_Management	O.Key_Destruction	O.Check_Operation	O.Prevent_Inf_Leakage
FDP_ACF.1/Oper	Security attribute based access control – Cryptographic Operation.			x									
FDP_ACC.2/Mode_Trans	Complete access control – Mode Transition.			x									
FDP_ACF.1/Mode_Trans	Security attribute based access control – Mode Transition.			x	x								
FDP_ITC.2	Import of user data with security attributes	x							x	x			
FDP_ETC.2	Export of user data with security attributes	x					x						
FDP_RIP.2	Full residual information protection												x
FMT_SMF.1	Specification of Management Functions			x									
FMT_SMR.2	Restrictions on security roles			x		x				x			
FMT_MOF.1/CO	Management of security functions behavior – Crypto Officer			x									
FMT_MOF.1/SO	Management of security functions behavior – Security Officer			x									
FMT_MTD.1/Admin	Management of TSF data - Admin		x										
FMT_MTD.1/User	Management of TSF data - User		x										
FMT_MSA.1/Key_Man_1	Management of security attributes – (change default and query)			x	x					x			
FMT_MSA.1/Key_Man_2	Management of security attributes – (modify or delete)			x	x					x			
FMT_MSA.2	Secure security attributes			x	x					x			
FMT_MSA.3	Static attribute initialisation			x				x		x			
FPT_TDC.1	Inter-TSF basic TSF data consistency						x		x	x			

Security Functional Requirement	SFR Full Title	O.Endorsed_Crypto	O.I&A	O.Control_Services	O.Control_Keys	O.Roles	O.Keys_Export	O.Key_Generation	O.Key_Import	O.Key_Management	O.Key_Destruction	O.Check_Operation	O.Prevent_Inf_Leakage
FPT_FLS.1	Failure with preservation of secure state											x	
FPT_EMS.1	TOE Emanation												x
FPT_TST.1	TSF testing											x	
FPT_TST.2	TSF self-testing											x	

Table 8-1 – Necessity of Security Functional Requirements

8.3 SFR Rationale

The following table provides rationale for the inclusion of chosen SFR against objectives:

Objectives	Security Functional Requirements	Rationale
O.Endorsed_Crypto	FCS_CKM.1 FCS_CKM.2/Secret_Symmetric_Key_Import FCS_CKM.2/Private_Key_Import FCS_CKM.2/Cloning_Import FCS_CKM.2/Secret_Symmetric_Key_Export FCS_CKM.2/Secret_Key_Export FCS_CKM.2/Cloning_Export FCS_CKM.4 FCS_COP.1/Sign FCS_COP.1/Digest FCS_COP.1/RSA_Enc_Dec FCS_COP.1/TDES_Enc_Dec FCS_COP.1/AES_Enc_Dec FCS_COP.1/Key_Derive FCS_COP.1/MAC FCS_RNG.1 FDP_ITC.2 FDP_ETC.2	FCS_CKM.1, FCS_CKM.2/Secret_Symmetric_Key_Import, FCS_CKM.2/Private_Key_Import, FCS_CKM.2/Cloning_Import, FCS_CKM.2/Secret_Symmetric_Key_Export, FCS_CKM.2/Secret_Key_Export, FCS_CKM.2/Cloning_Export, FCS_CKM.4, FCS_COP.1/Sign, FCS_COP.1/Digest, FCS_COP.1/RSA_Enc_Dec, FCS_COP.1/TDES_Enc_Dec, FCS_COP.1/AES_Enc_Dec, FCS_COP.1/Key_Derive, FCS_COP.1/MAC and FCS_RNG.1 require meeting Endorsed standards for cryptographic functions. FDP_ITC.2 and FDP_ETC.2 enforce the use of Endorsed cryptographic functions for import and export of confidential cryptographic keys.

Objectives	Security Functional Requirements	Rationale
O.I&A	FIA_UID.1 FIA_UAU.1 FIA_UAU.6 FIA_UAU.7 FIA_AFL.1/CO FIA_AFL.1/SO FIA_AFL.1/User FIA_ATD.1 FIA_USB.1 FMT_MTD.1/Admin FMT_MTD.1/User	FIA_UID.1 allows unidentified users to run self-test of the TOE only and requires identification before any other TSF mediated action. FIA_UAU.1 allows unauthenticated users to run self-test of the TOE, identification according FIA_UID.1 and selection of a claimed role and requires authentication before any other TSF mediated action. FIA_UAU.6 requires re-authentication after start-up of the TOE and if the user changes the role after authentication. FIA_UAU.7 requires limitation of the feedback to the user while authentication is in progress. FIA_AFL.1/CO, FIA_AFL.1/SO and FIA_AFL.1/User require detection and reaction to unsuccessful authentication attempts. FIA_ATD.1 requires maintaining security attributes to individual users including User identity, Role and Reference authentication data as prerequisite for identification and authentication of authorised users. FIA_USB.1 requires associating the identity and the role with the subjects acting for the authenticated user. FMT_MTD.1/Admin restricts the creation, clearing and deletion of Reference Authentication Data to the Security Officer. FMT_MTD.1/User restricts the ability to modify the Reference authentication data to the corresponding user.
O.Roles	FMT_SMR.2	FMT_SMR.2 which requires the TOE to provide at least the Security Officer, the Crypto Officer, the Crypto User roles, Unidentified User role, Unauthenticated User role.

Objectives	Security Functional Requirements	Rationale
O.Control_Services	FDP_ACC.2/Key_Man FDP_ACF.1/Key_Man FDP_ACC.2/Oper FDP_ACF.1/Oper FDP_ACC.2/Mode_Trans FDP_ACF.1/Mode_Trans FMT_SMF.1 FMT_SMR.2 FMT_MOF.1/CO FMT_MOF.1/SO FMT_MSA.1/Key_Man_1 FMT_MSA.1/Key_Man_2 FMT_MSA.2 FMT_MSA.3	FDP_ACC.2/Key_Man and FDP_ACF.1/Key_Man require access control to the key management services of the TOE, FDP_ACC.2/Oper and FDP_ACF.1/Oper require access control to the cryptographic operation services of the TOE, FDP_ACC.2/Mode_Trans and FDP_ACF.1/Mode_Trans require access control to the operational modes of the TOE which limit the available services. FMT_SMF.1 lists the security management functions including the management of TSF behaviour to FMT_MOF.1/CO and FMT_MOF.1/SO. FMT_SMR.2 describing the minimum list of roles and restrictions to these roles. FMT_MOF.1/CO and FMT_MOF.1/SO limit the management of TSF behaviour to the specified authorised roles FMT_MSA.1/Key_Man_1 and FMT_MSA.1/Key_Man_2 require limitation to the management of security attributes of cryptographic keys, key components and CSP describing the available services for these objects. FMT_MSA.2 and FMT_MSA.3 describe additional requirements to the management of security attributes to enforce the access control SFP for FDP_ACF.1/Key_Man, FDP_ACF.1/Oper and FDP_ACF.1/Mode_Trans.

Objectives	Security Functional Requirements	Rationale
O.Control_Keys	FDP_ACC.2/Key_Man FDP_ACF.1/Key_Man FDP_ACC.2/Oper FDP_ACF.1/Oper FMT_MSA.1/Key_Man_1 FMT_MSA.1/Key_Man_2 FMT_MSA.2 FDP_ACF.1/Mode_Trans	FDP_ACC.2/Key_Man and FDP_ACF.1/Key_Man require access control to the key keys, key components and other CSP according to their security attributes, FDP_ACC.2/Oper and FDP_ACF.1/Oper require access control to the keys and other CSP of the TOE according to their security attributes, FMT_MSA.1/Key_Man_1 and FMT_MSA.1/Key_Man_2 require limitation to the management of security attributes of cryptographic keys, cryptographic key components and CSP describing the access rights, available services and properties for these objects. FMT_MSA.2 ensures that only secure values for cryptographic keys, key components and CSP are accepted for security attributes. FDP_ACF.1/Mode_Trans ensures that operational keys and CSP cannot be used in maintenance mode and maintenance keys and CSP cannot be used outside the operational mode to protect user data.
O.Key_Export	FCS_CKM.2/Secret_Symmetric_Key_Export FCS_CKM.2/Secret_Key_Export FCS_CKM.2/Cloning_Export FTP_ITC.1/Cloning FDP_ETC.2 FPT_TDC.1	FCS_CKM.2/ Secret_Symmetric_Key_Export, FCS_CKM.2/Secret_Key_Export and FCS_CKM.2/Cloning_Export require the TSF to distribute keys by export methods meeting Endorsed standards and provides a refinement for keys exported for manual import. FTP_ITC.1/Cloning requires the TSF to provide a trusted channel for key export. FDP_ETC.2 requires the TSF to export keys unambiguously associated with their security attributes. FPT_TDC.1 requires ensuring inter-TSF basic TSF data consistency for exported security attributes of cryptographic keys, key components and CSP.

Objectives	Security Functional Requirements	Rationale
O.Key_Generation	FCS_CKM.1 FCS_RNG.1 FMT_MSA.3	FCS_CKM.1 requires the use of Endorsed key generation algorithms FCS_RNG.1 describes requirements for the random number generator needed for key generation. FMT_MSA.3 requires restrictive values of security attributes for cryptographic keys and limits the ability to specify their initial value to the Crypto officer.
O.Key_Import	FCS_CKM.2/Secret_Symmetric_Key_Import FCS_CKM.2/Private_Key_Import FCS_CKM.2/Cloning_Import FTP_ITC.1/Cloning FDP_ITC.2 FPT_TDC.1	FCS_CKM.2/Secret_Symmetric_Key_Import, FCS_CKM.2/Private_Key_Import and FCS_CKM.2/Cloning_Import require the TSF to distribute by key import methods meeting Endorsed standards and provides a refinement for manually imported keys. FTP_ITC.1/Cloning requires the TSF to provide a trusted channel of key import. FDP_ITC.2 requires the TSF to import keys unambiguously associated with their security attributes. FPT_TDC.1 requires ensuring inter-TSF basic TSF data consistency for imported security attributes of cryptographic keys, key components and CSP.

Objectives	Security Functional Requirements	Rationale
O.Key_Management	FCS_CKM.1, FCS_CKM.2/Secret_Symmetric_Key_Import FCS_CKM.2/Private_Key_Import FCS_CKM.2/Cloning_Import FCS_CKM.2/Private_Key_Exort FCS_CKM.2/Secret_Key_Export FCS_CKM.2/Cloning_Export FCS_CKM.4 FTP_ITC.1/Cloning FDP_ACC.2/Key_Man FDP_ACF.1/Key_Man FDP_ITC.2 FDP_ETC.2 FMT_SMF.1 FMT_MSA.1/Key_Man_1 FMT_MSA.1/Key_Man_2 FMT_MSA.2 FMT_MSA.3 FPT_TDC.1	FCS_CKM.1, FCS_CKM.2/Secret_Symmetric_Key_Import, FCS_CKM.2/Private_Key_Import, FCS_CKM.2/Cloning_Import, FCS_CKM.2/Secret_Symmetric_Key_Export, FCS_CKM.2/Secret_Key_Export, FCS_CKM.2/Cloning_Export, and FCS_CKM.4 provide the Endorsed cryptographic functions used by key management. FTP_ITC.1/Cloning provides a trusted channel for key import and export. FDP_ACC.2/Key_Man and FDP_ACF.1/Key_Man provide the access control to the key management functions. FDP_ITC.2 and FDP_ETC.2 ensure the import and export of cryptographic keys, cryptographic key components and CSP with security attribute, which are associated with these objects for key management. FMT_SMF.1 list the security management functions and FMT_SMR.2 the roles for key management (i.e. the Crypto Officer for operational keys and the Maintenance Personnel role for maintenance keys). FMT_MSA.1/Key_Man_1, FMT_MSA.1/Key_Man_2 FMT_MSA.2 and FMT_MSA.3 describes the management of security attributes of cryptographic keys, cryptographic key components and CSP. FPT_TDC.1 ensures the consistency of the security attributes of cryptographic keys, cryptographic key components and CSP.
O.Key_Destruction	FCS_CKM.4 FDP_ACF.1/Key_Man	FCS_CKM.4 requires the TSF to provide Endorsed mechanisms for key destruction. FDP_ACF.1/Key_Man limits key destruction to users in the Crypto Officer role.
O.Check_Operation	FPT_TST.1 FPT_TST.2 FPT_FLS.1	FPT_TST.1 and FPT_TST.2 require TSF self-tests. FPT_FLS.1 requires the TSF to preserve a secure state when self-test fails.

Objectives	Security Functional Requirements	Rationale
O.Prevent_Inf_Leakage	FDP_RIP.2 FPT_EMS.1	FDP_RIP.2 requires the TOE to ensure that any previous information content of a resource is made unavailable. FPT_EMS.1 requires the TOE to prohibit the flow of confidential information through any emanation and "black data" interface.

Table 8-2 – Mapping of Security Functional Requirements to Objectives

8.4 SFR Dependencies

The following SFR dependencies are recorded:

Security Functional Requirement	Dependencies	Rationale
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.2/Export[123] FCS_COP.1[124] FCS_CKM.4
FCS_CKM.2/Secret_Symmetric_Key_Export FCS_CKM.2/Private_Key_Export FCS_CKM.2/Cloning_Export	[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.1 FCS_CKM.4

[123] FCS_CKM.2/Export from the PP is now iterated in this ST to FCS_CKM.2/Secret_Symmetric_Key_Export, FCS_CKM.2/Secret_Key_Export and FCS_CKM.2/Cloning_Key_Export.

[124] All iterations of FCS_COP should be mapped with the exception of FCS_COP.1/digest which does not use or consume cryptographic keys in.

Security Functional Requirement	Dependencies	Rationale
FCS_CKM.2/Secret_Symmetric_Key_Import FCS_CKM.2/Private_Key_Import FCS_CKM.2/Cloning_Import	[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	FDP_ITC.2 FCS_CKM.1 FCS_CKM.4
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]	FDP_ITC.2 FCS_CKM.1
FCS_COP.1/Sign FCS_COP.1/Digest FCS_COP.1/RSA_Enc_Dec FCS_COP.1/TDES_Enc_Dec FCS_COP.1/AES_Enc_Dec FCS_COP.1/Key_Derive FCS_COP.1/MAC	[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	FDP_ITC.2 FCS_CKM.1 FCS_CKM.4
FCS_RNG.1	No dependencies	n/a
FDP_ACC.2/Oper	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Oper
FDP_ACC.2/Mode_Trans	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Mode_Trans

Security Functional Requirement	Dependencies	Rationale
FDP_ACC.2/Key_Man	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Key_Man
FDP_ACF.1/Key_Man	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/KeyMan FMT_MSA.3
FDP_ACF.1/Mode_Trans	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/Mode_Trans FMT_MSA.3
FDP_ACF.1/Oper	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/Oper FMT_MSA.3
FDP_ETC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.2/Key_Man FDP_ACC.2/Oper (hierarchical to FDP_ACC.1)
FDP_ITC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.2/Key_Man FDP_ACC.2/Oper (hierarchical to FDP_ACC.1) FTP_ITC.1/Cloning FPT_TDC.1
FDP_RIP.2	No dependencies	n/a
FIA_AFL.1/CO FIA_AFL.1/SO FIA_AFL.1/User	FIA_UAU.1 Timing of authentication	FIA_UAU.1
FIA_ATD.1	No dependencies	n/a
FIA_UAU.1	FIA_UID.1 Timing of identification	FIA_UID.1
FIA_UAU.6	No dependencies	n/a

Security Functional Requirement	Dependencies	Rationale
FIA_UAU.7	FIA_UAU.1 Timing of authentication	FIA_UAU.1
FIA_UID.1	No dependencies	n/a
FIA_USB.1	FIA_ATD.1 User attribute definition	FIA_ATD.1
FMT_MOF.1/CO FMT_MOF.1/SO	FMT_SMF.1 Specification of management functions, FMT_SMR.1 Security roles	FMT_SMR.2 (hierarchical to FMT_SMR.1) FMT_SMF.1
FMT_MSA.1/Key_Man_1 FMT_MSA.1/Key_Man_2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMF.1 Specification of Management Functions FMT_SMR.1 Security roles	FDP_ACC.2/Key_Man, FDP_ACC.2/Oper (hierarchical to FDP_ACC.1) FMT_SMF.1 FMT_SMR.2 (hierarchical to FMT_SMR.1)
FMT_MSA.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FDP_ACC.2 (all iterations, hierarchical to FDP_ACC.1) FMT_MSA.1/Key_Man_1, FMT_MSA.1/Key_Man_2 FMT_SMR.2 (hierarchical to FMT_SMR.1)
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1/Key_Man_1 FMT_MSA.1/Key_Man_2 FMT_SMR.2 (hierarchical to FMT_SMR.1)
FMT_MTD.1/Admin FMT_MTD.1/User	FMT_SMF.1 Specification of Management Functions FMT_SMR.1 Security roles	FMT_SMR.2 (hierarchical to FMT_SMR.1) FMT_SMF.1
FMT_SMF.1	No dependencies	n/a

Security Functional Requirement	Dependencies	Rationale
FMT_SMR.2	FIA_UID.1 Timing of identification	FIA_UID.1
FPT_EMS.1	No dependencies	n/a
FPT_FLS.1	No dependencies	n/a
FPT_TDC.1	No dependencies	n/a
FPT_TST.1	No dependencies	n/a
FPT_TST.2	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1
FTP_ITC.1/Cloning	No dependencies	n/a

Table 8-3 – Dependency Rationale for Security Functional Requirement

8.5 Mapping of IT Security Functions to IT Security Requirements and SFRs

IT Security Function	TSS Reference	CC Requirement Title	CC Functional Component	ST Reference	Rationale
Cryptographic Material Management	7.3.8	Cryptographic key generation	FCS_CKM.1	6.1.1.1	The security function satisfies the SFR by providing mechanisms for the generation of RSA, DSA, ECDSA, DH, ECDH, AES and TDES keys of the specified lengths in accordance with the appropriate standards.
Key Cloning Key Wrap/Unwrap	7.3.18 7.3.8.3	Cryptographic key distribution	FCS_CKM.2	6.1.1.2 6.1.1.3 6.1.1.4 6.1.1.5 6.1.1.6 0	The security function satisfies the SFR by providing (i) a secure means to derive a key as part of the cloning protocol to encrypt the backed up keys when transferred from the TOE to another distinct TOE or Trusted IT Product or (ii) using RSA, or wrapped according to PKCS #8 using AES or TDES
Cryptographic Material Management	7.3.8	Cryptographic key destruction	FCS_CKM.4	6.1.1.8	The security function satisfies the SFR by providing mechanisms that destroys keys in accordance with Endorsed methods
Cryptography	7.3.9	Cryptographic operation	FCS_COP.1	6.1.1.10 6.1.1.11 6.1.1.12 6.1.1.13 6.1.1.14 6.1.1.15 6.1.1.16	The security function satisfies the SFR by providing mechanisms that implement the specified set of cryptographic algorithms in accordance with the appropriate standards.

IT Security Function	TSS Reference	CC Requirement Title	CC Functional Component	ST Reference	Rationale
Cryptography	7.3.9	Random Number Generation	FCS_RNG.1	6.1.1.17	The security function satisfies the SFR by providing a Random Number Generator that can be tested to demonstrate sufficient entropy.
Access Control	7.3.4	Complete access control	FDP_ACC.2	6.1.3.1 6.1.3.3 6.1.3.5	The security function satisfies the SFR by enforcing the Token Access Control (TAC) policy on subjects (sessions), objects and a set of controlled operations.
Access Control	7.3.4	Security attribute based access control	FDP_ACF.1	6.1.3.2 6.1.3.4 6.1.3.6	The security function satisfies the SFR by enforcing the access control policies based on the specified sets of subject and object attributes.
Key Export and Import Protection	7.3.7	Export of user data with security attributes	FDP_ETC.2	6.1.3.8	The security function satisfies the SFR by enforcing the TAC when data is exported through a Wrap operation. Objects are exported without security-related attributes.
Key Export and Import Protection	7.3.7	Imported user data with security attributes	FDP_ITC.2	6.1.3.7	The security function satisfies the SFR through two routes: (1) by enforcing the TAC when data is imported through an Unwrap operation. The TSF ignores any security-related attributes that may have been associated with the imported object and sets the object's attributes to the appropriate values for its type and, in particular, the CKA_SENSITIVE attribute is always set. (2) ensuring consistent direct transfer using cloning including direct transfer of all key attributes.

IT Security Function	TSS Reference	CC Requirement Title	CC Functional Component	ST Reference	Rationale
Object Reuse	7.3.5	Full residual information protection	FDP_RIP.2	6.1.3.9	The security function satisfies the SFR by ensuring that the information content of resources is made unavailable when the resource is re-allocated.
User Identification and Authentication	7.3.1	Authentication failure handling	FIA_AFL.1	6.1.2.7 6.1.2.8 6.1.2.9	The security function satisfies the SFR by detecting when the maximum number of login failures occur (3 for SO, set in the TPV for Token User; 3-10 for the CO, as configured by the SO) and performing on of the following: Zeroise the device in the case of SO authentication failure and if configured by the SO, CO authentication failure. Locking out the CO user Remove the user and zeroise the user’s memory space, if a Token User authentication failure.
User Account Data	7.3.3	User attribute definition	FIA_ATD.1	6.1.2.1	The security function satisfies the SFR by maintaining the required list of security attributes within the UAV for each Token User.
User Identification and Authentication	7.3.1	Timing of authentication	FIA_UAU.1	6.1.2.3	The security function satisfies the SFR by allowing a user to perform a specified set of actions before authentication and by requiring the user to be successfully authenticated before allowing the user to perform any other actions on the module.
User Identification and Authentication	7.3.1	Re-authenticating	FIA_UAU.6	6.1.2.4	The security function satisfies the SFR by only allowing a user to authenticate against a single role, enforcing re-authentication via a trusted IT device (PED) and credentials when needing to assume a new role, and powering up in an unauthenticated state.

IT Security Function	TSS Reference	CC Requirement Title	CC Functional Component	ST Reference	Rationale
User Identification and Authentication	7.3.1	Protected authentication feedback	FIA_UAU.7	6.1.2.5	The security function satisfies the SFR by not outputting sensitive data to the trusted PIN entry device.
User Identification and Authentication	7.3.1	Timing of identification	FIA_UID. 1	6.1.2.2	The security function satisfies the SFR by allowing a user to perform a specified set of actions before identification and by requiring the user to be successfully identified before allowing the user to perform any other actions on the module.
User Identification and Authentication	7.3.1	User subject binding	FIA_USB.1	6.1.2.6	The security function satisfies the SFR by specifying that the user identity be bound to the subject (session) acting on behalf of the user by including the UAV data within the session state.
Security Function Management	7.3.12	Management of security functions behaviour	FMT_MOF.1	6.1.4.3	The security function satisfies the SFR by restricting the ability to perform the specified security management operations to the CO and SO roles.
Security Data Management	7.3.13	Management of security attributes	FMT_MSA.1	6.1.4.7 6.1.4.8	The security function satisfies the SFR by enforcing the TAC Policy to restrict the ability to manipulate user and object security attributes as specified.
Security Data Management	7.3.13	Secure security attributes	FMT_MSA.2	6.1.4.9	The security function satisfies the SFR by ensuring that only secure values are accepted for security attributes.
Security Data Management	7.3.13	Static attribute initialisation	FMT_MSA.3	6.1.4.10	The security function satisfies the SFR by requiring restrictive values for security attributes that cannot be changed based on the capability and policy settings.
Security Data Management	7.3.13	Management of TSF data	FMT_MTD.1	6.1.4.5	The security function satisfies the SFR by enforcing the TAC Policy to restrict the ability to manipulate the policy settings to the SO or corresponding user.

IT Security Function	TSS Reference	CC Requirement Title	CC Functional Component	ST Reference	Rationale
Specification of Security Management Functions	7.3.11	Specification of management functions	FMT_SMF.1	6.1.4.1	The security function satisfies the SFR by specifying the security management functions that may be performed.
TOE Roles	7.1.3	Restrictions on security roles	FMT_SMR.2	6.1.4.2	The security function satisfies the SFR by specifying the security roles that are implemented by the TOE – Security Officer, Crypto Officer and Crypto User.
Physical Self-Protection	7.3.16	TOE Emanation	FPT_EMS.1	6.1.5.3	The security function satisfies the SFR by not allowing emissions beyond levels specified during conducted and radiated emissions testing.
Preservation of Secure State	7.3.14.4	Failure with preservation of secure state	FPT_FLS.1	6.1.5.2	The security function satisfies the SFR by preserving the module in a secure state when the specified failure conditions occur.
Backup and Recovery	7.3.18	Inter-TSF confidentiality during transmission	FTP_ITC.1/Cloning	6.1.1.9	The security function satisfies the SFR by encrypting the transmitted data according to the cloning protocol.
Key Storage and Access Protection	7.3.8.1	Inter-TSF basic TSF data consistency	FPT_TDC.1	6.1.5.1	The security function satisfies the SFR by ensuring any externally-set attributes of imported keys are ignored by the module and values are set according to the values required by the Access Control Policy.
Key Cloning	7.3.18	Inter-TSF basic TSF data consistency	FPT_TDC.1	6.1.5.1	The security function satisfies the SFR by encrypting objects transferred between HSM modules
Key Wrap/Unwrap	7.3.8.3	Inter-TSF basic TSF data consistency	FPT_TDC.1	6.1.5.1	The security function satisfies the SFR by validating keys and key components imported and exported through the wrapping/unwrapping operation and protecting them from modification through the use of RSA or symmetric encryption.

IT Security Function	TSS Reference	CC Requirement Title	CC Functional Component	ST Reference	Rationale
Self-Tests	7.3.14.2	TSF testing	FPT_TST.1	6.1.5.4	The security function satisfies the SFR by providing a suite of self-tests to verify the correct operation of the security functions on start-up.
Memory and Firmware Integrity Check Self-Tests	7.3.14.1 7.3.14.2	TSF self-testing	FPT_TST.2	6.1.5.5	The security function satisfies the SFR by implementing start-up and/or conditional known-answer tests, random number tests, pair-wise consistency tests and firmware integrity tests.

Table 8-4 – Mapping of IT Security Functions to IT Security Requirements and SFRs

APPENDIX A – REFERENCES

ID	Document Number	Revision	Author	Title
[1]	ISO/IEC 15408-1	V3.1R4	N/A	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and General Model
[2]	ISO/IEC 15408-2	V3.1 R4	N/A	Common Criteria for Information Technology Security Evaluation – Part 2: Security Functional Requirements
[3]	ISO/IEC 15408-3	V3.1R4	N/A	Common Criteria for Information Technology Security Evaluation – Part 3: Security Assurance Requirements
[4]	FIPS 140-2	December 2002	NIST	Security Requirements for Cryptographic Modules
[5]	CR-3854	1	SafeNet, Inc.	Overview of Documentation Required to Support Luna® PCI-E in Luna SA v5.2.3 Common Criteria Evaluation
[6]	CR-3416	11	SafeNet, Inc.	This document defines the logical interface specification for the Luna® PCI-E in Luna SA v5.2.3 Cryptographic Module.
[7]	AIS20	Version 2.1, 2011-12-02	BSI	Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 20, Funktionalitätsklassen und Evaluationsmethodologie für deterministischeZufallszahlengeneratoren
[8]	AIS31	Version 2.1, 2011-12-02	BSI	Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 31, Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren
[9]	BSI-CC-PP-00078	V2.0.2, 2012-12-06	BSI	Protection Profile – Cryptographic Modules, Security Level “Enhanced Basic”
[10]	BSI-CC-PP-00079	V2.0.2, 2012-12-19	BSI	Protection Profile – Cryptographic Modules, Security Level “Standard”
[11]	FIPS PUB 186-3	3, Issued June 2009	NIST	Digital Signature Standard (DSS)
[12]	ANSI X9.31	Sept 2008	ANSI	Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA).

ID	Document Number	Revision	Author	Title
[13]	NIST SP 800-67	January 2012	NIST	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher.
[14]	NIST SP 800-57	July 2012, Revision 3.	NIST	Recommendations for Key Management – Part 1: General.
[15]	FIPS PUB 197	Nov, 2001	NIST	Advanced Encryption Standard (AES)
[16]	ANSI X9.62	November 2005	ANSI	The Elliptic Curve Digital Signature Algorithm (ECDSA)
[17]	NIST SP-800-56A	May 2013	NIST	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography.
[18]	PKCS #1	V2.1	RSA Laboratories	PKCS #1: RSA Cryptography Standard
[20]	PKCS #8	V1.2	RSA Laboratories	PKCS #8: Private-Key Information Syntax Specification.
[21]	PKCS #11	Version 2.20, June 2004	RSA Laboratories	PKCS #11: Cryptographic Token Interface Standard
[22]	CWA 14167-2	Version: 0.28, October 2003	CEN/ISSS	Cryptographic Module for CSP Signing Operations with Backup – Protection Profile
[23]	*FIPS PUB 180-4*	March 2012	NIST	Secure Hash Standard (SHS)
[24]	CR-4119	2.0, May 30th 2017	SafeNet, Inc	Luna PCI-E Cryptographic Module, Common Criteria User Guidance.
[25]	SP 800-108	October 2009	NIST	Recommendation for Key Derivation Using Pseudorandom Functions.
[26]	SP 800-38D	November 2007	NIST	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.
[27]	SP 800-38B	May 2005	NIST	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication.
[28]	FIPS PUB 113	May 1985	NIST	Computer Data Authentication
[29]	FIPS PUB 198-1	July 2008	NIST	The Keyed-Hash Message Authentication Code (HMAC)