SECURITY TARGET FOR LUNA® CA4 SYSTEM VERSION 2.6

DOCUMENT NUMBER:	CR-2951
AUTHOR:	Terry Fletcher
DEPARTMENT:	Engineering
LOCATION OF ISSUE:	Ottawa
DATE ORIGINATED:	February 4, 2009
REVISION LEVEL:	7
REVISION DATE:	December 20, 2011
SUPERSESSION DATA:	CR-2951, Revision 5 dated October 21, 2011
SECURITY LEVEL:	Non-sensitive

SafeNet, Inc. reserves the right to make changes in the product or its specifications mentioned in this publication without notice. Accordingly, the reader is cautioned to verify that information in this publication is current before placing orders. The information furnished by SafeNet, Inc. in this document is believed to be accurate and reliable. However, no responsibility is assumed by SafeNet, Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

DOCUMENT CHANGE HISTORY

Revision	Date	Reason for Change	Sections Affected
Original	February 4, 2009	First release of document	All
1	March 31, 2009	Updates in accordance with issued ORs.	All
2	October 29, 2009	Minor updates in accordance with issued ORs.	All
3	March 19, 2010	Minor updates in accordance with issued ORs	1.3, 3.2, 6.1.1.8
4	July 21, 2010	Update TOE Version to 2.5	Title, 1.2, 3.1, Table 8-7, Appendix A
5	October 4, 2011	Update TOE Version to 2.6	Title, 1.2, 3.1, Table 8-7, Appendix A
6	October 21, 2011	Update to TOE boundary	Sections 1.2, 1.3, 1.4
7	December 20, 2011	Removal of Proprietary notice.	All

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF TERMS

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

Approved algorithms and parameters means cryptographic algorithms and parameters approved for use in electronic signatures, secure signature creation devices and trustworthy systems by the appropriate national standards body.

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

Auditor means a user exporting the TOE audit data and reviewing the audit data with tools in the TOE environment.

Backup means secure export and external storage of the keys, the TSF data and the system data (backup data) sufficient to recreate the state of the TOE at the time the backup was created.

Certificate means an electronic attestation which links the SVD to a person and confirms the identity of that person.

Certification Authority (CA) is a synonym for CSP, defined below.

Certification-Service-provider (CSP) means an entity or a legal or natural person who issues certificates or provides other services related to electronic signatures.

Data to be signed (DTBS) means the complete electronic data to be signed, such as certificate content data or certificate status information.

Digital signature means data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery e.g. by the recipient. [ISO 7498-2]

Dual person control means a special form of access control of a task which requires two users with different identities to be authenticated and authorised to the defined roles at the time this task is to be performed.

Hardware security module (HSM) means a cryptographic module used to generate the advanced signature in qualified certificates. The TOE specified in this Security Target is an HSM.

Restore means import of the backup data to recreate the state of the TOE at the time the backup was created.

User means any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

User data means data created by and for the user that does not affect the operation of the TSF.

LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
CA	Certificate Authority
CC	Common Criteria
CIMS	Certificate Issuing and Management System
CLI	Command Line Interface
CO	Crypto Officer
COTS	Commercial Off-the-Shelf
CSP	Certification Service Provider
CU	Crypto User
DES	Data Encryption Standard
DLL	Dynamic Linked Library
DSA	Digital Signature Algorithm
DTBS	Data to be Signed
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
HOK	Hardware Origin Key
HSM	Hardware Security Module
IETF	Internet Engineering Task Force
IT	Information Technology
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PED	PIN Entry Device
PIN	Personal Identification Number
PKC	Public Key Confirmation
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
PP	Protection Profile
PRNG	Pseudo-Random Number Generator
RAM	Random Access Memory
RNG	Random Number Generator (Generation)
ROM	Read-Only Memory
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
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

Document Organisation

A **Glossary of Terms and a List of Acronyms and Abbreviations** list is provided to define frequently used terms and acronyms.

Section 1 provides the introductory material for the Security Target and provides a high-level description of the TOE.

Section 2. states the Conformance Claims for this Security Target.

Section 3 is the Security Problem Definition. It provides a discussion of the expected environment for the TOE. It defines the set of threats that are to be addressed by either the technical countermeasures implemented in the TOE hardware, the TOE software, or through the environmental controls and the organizational security policies to be met.

Section 4 defines the security objectives for both the TOE and the TOE environment.

Section 5 defines the extended security functional requirement used by the Security Target.

Section 6 contains the functional requirements and assurance requirements derived from the Common Criteria (CC), Part 2 [2] and Part 3 [3], that must be satisfied by the TOE.

Section 7 provides the TOE Summary Specification.

Section 8 provides the rationale to explicitly demonstrate that the information technology security objectives satisfy the policies and threats. Arguments are provided for the coverage of each policy and threat. The section then explains how the set of requirements are complete relative to the objectives, and that each security objective is addressed by one or more component requirements. Arguments are provided for the coverage of each objective. Section 8 also addresses dependency analysis and presents a summary of the evidence used to demonstrate conformance with the Security Assurance Requirements.

Appendix A contains the list of references used in compiling this ST.

1. ST INTRODUCTION

1.1. ST Reference

Title: Luna® CA4 System Security Target (ST)
Assurance level: EAL 4-augmented by ALC_FLR.2
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® CA4 System
TOE Version Luna® CA4 Hardware Versions 808-000014-002 (900578-001) and 808-000003-001 (900578-002), Firmware Version 4.8.7, Software Version 2.6

1.3. TOE Overview

The Target of Evaluation (TOE) provides high assurance key management and cryptographic services to user application systems. The TOE consists of the following components:

- two (2) SafeNet, Inc. Luna® CA4 devices, each in a PC Card form factor, (referred to as tokens) (Hardware Versions 808-000014-002 [900578-001] and 808-000003-001 [900578-002]; Firmware Version 4.8.7),
- a dual-slot Luna® Dock II PC Card Reader (Hardware Version 908-55007-001 [006850-001]; Firmware Version 0x00C1),
- Luna® PIN Entry Device (PED II) (Hardware Versions 908-25024-001 and 908-000008-002 [808-00012-002]; Firmware Versions 2.0.2 and 2.4.0) and iKeys with labels
- lunacm (setup and administration) software (Version 2.3.3),
- Luna® CA4 software, version 2.6 (including PKCS #11 Cryptographic API Software) (700-010445-002, Revision A), and
- Luna® CA4 2.6 Guidance Documentation (700-010446-002, Revision A).

The TOE Security Functions (TSF) are all implemented within the Luna® CA4 cryptographic module along with the PED II. The CA4 is contained in its own secure enclosure that provides physical resistance to tampering.

The SafeNet Luna® CA4 is ideally suited as a Hardware Security Module (HSM) for use in digital identity and data protection applications. The Luna® CA4 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® CA4 to prevent compromise.

The Luna® CA4 also provides advanced features like direct hardware-to-hardware backup, split user role administration, multi-person authentication, and trusted path authentication that are used by security-conscious organizations in deployments around the world.

1.3.1. 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 of 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 and verification using RSA and ECDSA key pairs.

The TOE comprises processors, read-only and random-access memory, and firmware packaged in a tamper-resistant form along with Cryptographic API software that resides on the host computer. It is accessed directly (i.e., electrically) via either the PED II serial interface or via the PCMCIA interface. Logical access to key material and cryptographic services for users and user application software is provided indirectly through the Cryptographic API software on the host computer.

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 its 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 (3) roles to be identified and authenticated before they are authorized to perform any cryptographic and/or key management operations. Authentication is performed using the Luna® PED II or a combination of Luna® PED II and a one-time challenge-response mechanism for the Crypto Officer and Crypto User roles.

In order to support requirements for strict separation of duties and/or multi-person control of critical security functions, the TOE also supports an optional M of N secret sharing mechanism to split the authentication data stored on the users' iKeys.

1.3.2. TOE Type

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

1.3.3. Required non-TOE Components

The TOE must run in conjunction with a standard server platform and the TOE's software components – Cryptographic API library software and lunacm – must be hosted on the computer platform. The host platform could be any computer suitable for Windows Server 2003 or 2008, Red Hat Enterprise Linux 5 or for Sun Solaris 10.

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, digital signature and verification used by application systems that provide cryptographic support functions such as a Certificate Authority/Certification Service Provider (CA/CSP) or Time Stamp Authority (TSA). It includes processors, read-only and random-access memory, and firmware packaged in a tamper-resistant form along with Cryptographic API software that resides on the host computer.

The TOE supports backup and restoration of cryptographic objects, such as private signing keys to facilitate recovery from a failure. Backup and restoration is done using cryptographic protocols and mechanisms that protect the confidentiality of the backup data and detect loss of the integrity of the backup data. Measures must be taken within the non-IT environment to ensure the availability of the backup data.

Figure 1 shows the TOE in its typical deployment configuration – as a system including the Dock II card reader, PED II and iKeys.

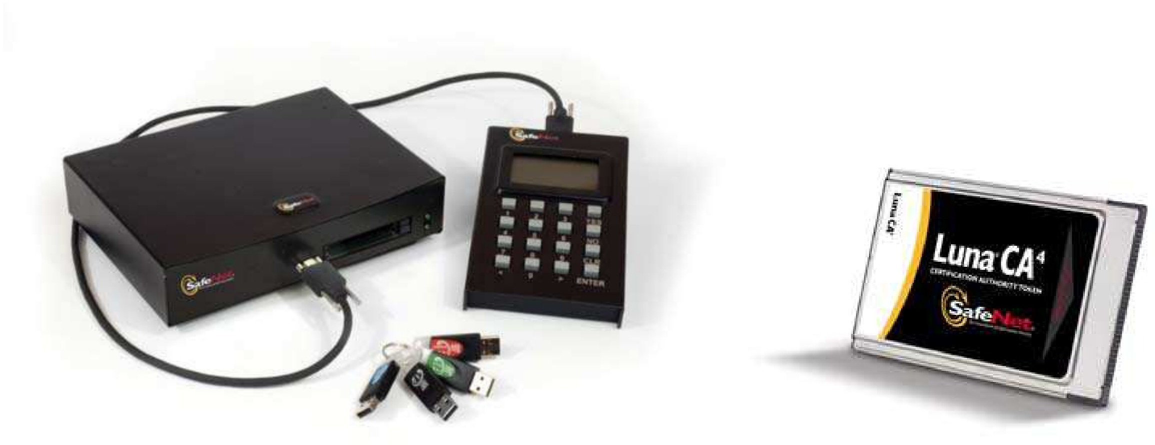

Figure 1: Luna® CA4 with Luna® Dock II, PED II and iKeys

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

1. The Luna® CA4 cryptographic module – a printed circuit board in PC card format enclosed within tamper-resistant metal covers. The printed circuit board hosts volatile and non-volatile memory, a microprocessor, with its associated firmware, data, control and key transfer signal paths, an FPGA that provides an entropy selection function for the on-board random bit generator, input/output controller, power management and a local oscillator.
2. The Luna® PIN Entry Device (PED II), which is housed in a separate physical enclosure and, through a physically and electrically separate data port connection to the module, provides a trusted path for the communication of critical security parameters (authentication data and plaintext cryptographic parameters) to and from the module.
3. iKeys, which are USB token devices used to securely store authentication data and other critical security parameters for entry through the PED II.
4. The Luna® Dock II (Dock II) card reader, which is housed in a separate physical enclosure, facilitates the connection of the PED II to the Luna CA4 cryptographic module.
5. PKCS #11 cryptographic API library and driver software provides the programming and communications interface normally used to access the cryptographic module.
6. lunacm – administrative software for use with Luna® CA4.
7. User and Administrative Guidance documentation for the TOE is provided on CD-ROM along with client PKCS #11 software.

The TSF boundary is the Luna® CA4 cryptographic module plus the PED II.

The TOE in the evaluated configuration is supported on the following operating systems:

- Windows Server 2008
- Windows Server 2003 (32-bit and 64-bit versions)
- Solaris 10 (32-bit and 64-bit versions)
- Red Hat Enterprise Linux 5 (RHEL-5) (32-bit and 64-bit versions)

Figure 2 below illustrates the relationship between the TOE and the software running on the host platform and also shows the division between the TSF and non-TSF portions of the TOE, specifically the Luna® CA4 HSM and the supporting Luna® PED II and library and driver software.

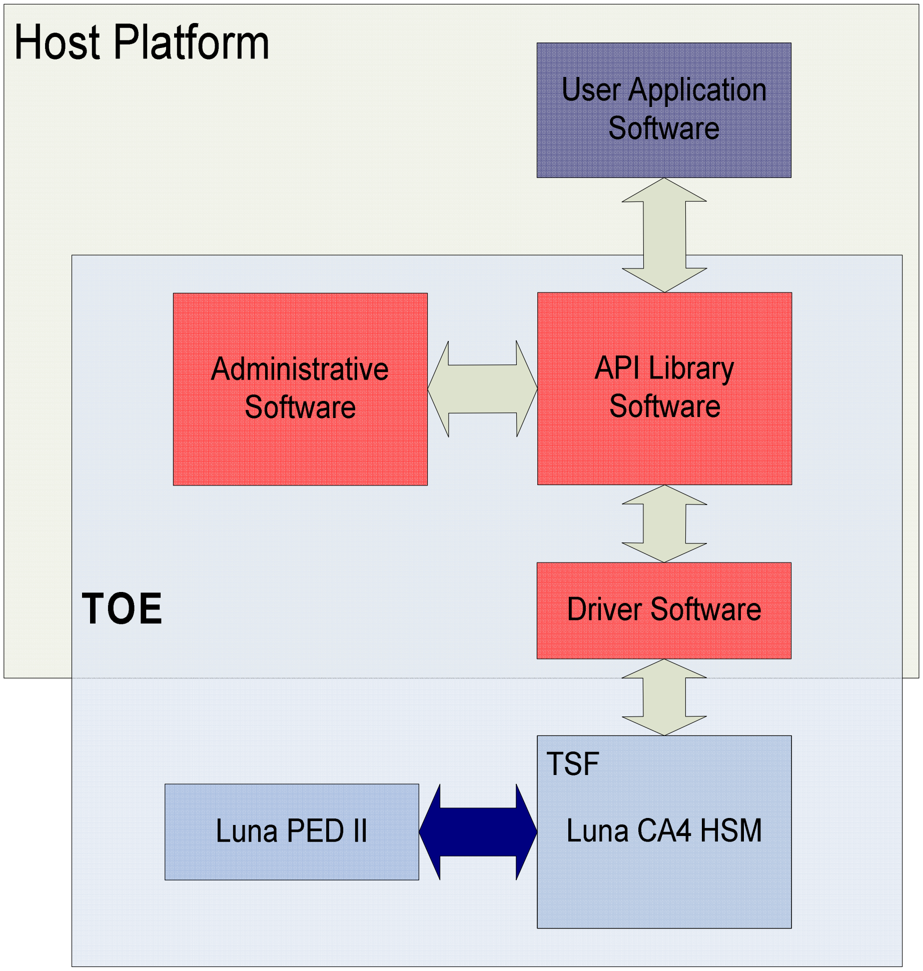

Figure 2: Illustration of the TOE and TSF in the Context of the Host Platform

1.4.1. TOE Roles

The following authenticated roles are supported by the TOE:

- Security Officer (SO) – authorized 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) – authorized to create, use, destroy and backup/restore cryptographic objects.
- Crypto User (CU) – authorized to use cryptographic objects (e.g., sign, encrypt/decrypt).

The CO and CU communicate with the Luna® CA4 for cryptographic operations using the PKCS #11 API. The SO uses a separate Command Line Interface (CLI), which is part of the administrative software, to perform configuration, security policy settings and user creation/deletion. The CLI is also used by the CO to perform backup and restoration of cryptographic objects.

The TOE allows for the creation of multiple users in the CO and CU roles. Each user is created within a cryptographically separated partition in the Luna® CA4 cryptographic module and each partition must have one and only one user in the CO role. A partition may also be assigned the CU role in addition to CO. The partition user can then adopt either role by providing the appropriate password at login time. Throughout the remainder of the ST, the term User will be used to refer to a partition user, in either the CU or CO role, when it is either not required or not appropriate to distinguish between the roles. The term user will be used to refer to a generic user, either unauthenticated or authenticated in any one of the three roles.

In Table 1-1 the roles supported by the TOE are compared to the roles defined in FIPS PUB 140-2 and PKCS#11.

Table 1-1 – FIPS 140-2 to TOE Role Comparison

Function	PKCS#11 Role	FIPS 140-2	TOE Role
Initialisation, configuration	Security Officer	Crypto Officer	Security Officer
Key Management	User	Crypto Officer	Crypto Officer
Use	User	Crypto User	Crypto User

1.4.2. Cryptographic Services

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

Random Number Generation

A trustworthy Random Number Generator is required to support secure generation of symmetric keys and asymmetric key pairs.

- FIPS 140-2 validated Deterministic Random Bit Generator (Pseudo-random Number Generator) seeded by internal Hardware Non-deterministic Random Bit Generator
- Based on ANSI X9.31, Appendix A section 2.4

Generate Public/Private Key Pairs

It is important that key pairs are properly generated in accordance with approved standards. The TOE provides key pair generation in accordance with the following standards.

- RSA 1024, 2048, 4096 bits key pairs in accordance with ANSI X9.31
- DSA 1024 bits key pairs in accordance with FIPS PUB 186-3
- Elliptic Curve key pairs in accordance with FIPS PUB 186-3

Generate Secret (Symmetric) Keys

It is important that symmetric keys are properly generated in accordance with approved standards. The TOE provides key generation in accordance with the following standards.

- TDES 168 bits (security strength) in accordance with SP 800-67
- AES 128, 192, 256 bits in accordance with FIPS PUB 197

Secure Key Material Storage and Access

Sensitive key values must be strongly protected and never be visible in plaintext form. The TOE ensures this in the following ways.

- Key material is stored in hardware and strongly encrypted
- Access to private keys and symmetric keys is provided via key handles only

Compute Digital Signatures and Verify Digital Signatures

The TOE computes and verifies digital signatures in accordance with the following standards.

- RSA 1024 bits, 2048 bits and 4096 bits (PKCS #1 V1.5, PKCS #1 PSS, ANSI X9.31) with SHA-1
- RSA 1024 bits, 2048 bits and 4096 bits (PKCS #1 V1.5, PKCS #1 PSS) with SHA-224, 256, 384, 512
- DSA 1024 bits (FIPS PUB 186-3) with SHA-1
- ECDSA (FIPS PUB 186-3 Appendix D recommended curves) with SHA-1, 224, 256, 384, 512.

Encrypt / Decrypt Data

The TOE performs encryption and decryption operations on user data in accordance with the following standards.

- RSA 1024, 2048 and 4096 bits in accordance with PKCS #1 V1.5 and OAEP
- TDES (ECB and CBC mode) 168 bits (security strength) in accordance with SP 800-67
- AES (ECB and CBC mode) 128 and 256 bits in accordance with FIPS PUB 197

Import (Unwrap) Private Keys

The TOE can import private keys using an Unwrap operation in accordance with the following standard.

- RSA 1024, 2048 and 4096 bit private keys in PKCS #8 format with TDES and AES in CBC mode

Export (Wrap) and Import (Unwrap) Secret Keys

The TOE can export and import symmetric keys using Wrap and Unwrap operations in accordance with the following standards.

- TDES, AES with TDES and AES in ECB mode
- TDES, AES with RSA 1024, 2048 and 4096 bits in accordance with PKCS #1 V1.5

Handling of key material and the use of cryptographic functions must be done in accordance with the key management procedures and policies of the user organization.

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 the creation and destruction of keys,
- Access control for security administration functions,
- Access control for usage of keys with cryptographic functions, and
- Self-test of the TOE.

1.4.4. Trusted Path – Luna® PED II

User authentication data and other critical security parameters are protected through the use of a separate port and data path for their transfer, and by providing mechanisms to protect their confidentiality and integrity. Attached to this separate data port is the Luna® PIN Entry Device or Luna® PED II.

The Luna® PED II, with accompanying iKeys, is depicted in Figure 3. It houses a number of input/output interfaces that, in combination, provide a trusted path device for the communication of authentication data and critical security parameters to and from the Luna® CA4 cryptographic module. The Luna® PED II has a graphics display used to display status and prompt messages, and a challenge secret that is output by the cryptographic module at the time a partition is created [see sub-section 1.4.5]. It has a keypad used to enter simple responses (Yes/No/Enter) and to enter an optional PIN that is combined with the authentication data stored on an iKey as part of the authentication process. It has a USB receptacle for the input/output of data to the iKey and it has a serial communications cable that connects to the separate data port, which is wired directly to the cryptographic module. Because the PED II has a direct serial communications interface to the cryptographic module, only local entry of iKey authentication data is possible.

The following types of iKey are used with the Luna® PED II:

- Blue (SO) iKey – for the storage of SO authentication data,
- Black (User) iKey – for the storage of User authentication data,
- Red (Domain) iKey – for the storage of the cloning domain data, used to control the ability to clone from a cryptographic module to a backup token, and
- Green (M of N) iKeys – used to store M of N secret shares, used for multi-person control of critical functions.

Any iKey, once data has been written to it, is an Identification and Authentication device and must be safeguarded accordingly by the administrative or operations staff responsible for the operation of the TOE within the customer's environment.

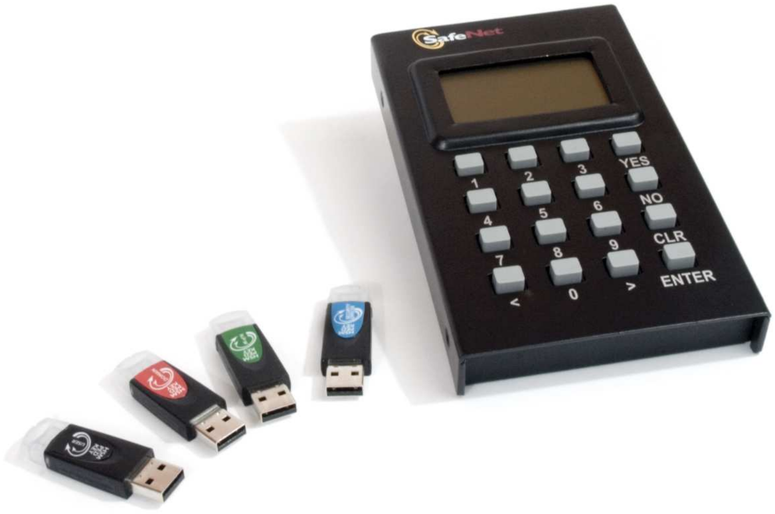

Figure 3 Luna® PED II with iKeys

1.4.5. User Authentication

The TOE requires that all users (SO, CO and CU roles) be authenticated by proving knowledge of a secret shared by the user and the cryptographic module.

The TOE generates the authentication secrets using its Pseudo-Random Number Generator (PRNG). 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(s); these are generated at the time the user's partition is created by the SO. The authentication secret(s) are provided to the operator via the Luna® PED II and iKey, described in sub-section 1.4.4, and must be entered by the operator via the Luna® PED II and via a logically separate trusted channel (in the case of the response based on the challenge secret) during the login process. Both the CO and CU use the same 48-byte random secret. If a partition is created with CO and CU roles, a separate challenge secret is generated for each role.

SO authentication requires the transmission to the cryptographic module of the Blue (SO) iKey data combined with the optional PIN through the trusted path.

User authentication is a two-stage process. The first stage is termed "Activation" and is performed using the Luna® PED II. Activation requires the transmission to the cryptographic module of the Black (User) iKey data combined with the optional PIN through the trusted path. Once Activation has been performed, 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 typically 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 the CO and CU roles are used) for the partition is generated by the cryptographic module as a 75-bit random value that is displayed as a 16-character string on the visual display of the trusted path device. The challenge secret is then provided, via a secure out-of-band means, to each external entity authorized to connect to the partition. It is entered by the external entity during login and is used by the TOE's PKCS #11 library software, acting on behalf of the 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".

1.4.6. Configurable Policy Settings

The Luna® CA4 firmware 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 that are possible to make for the TOE configuration are shown in sections 1.4.6.1 and 1.4.6.2. 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.

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.

In the next two sub-sections, all capability elements described as “allow/disallow some functionality” are Boolean values where false (or zero) equates to disallow the functionality and true (or one) equates to allow the functionality. Except as noted, all Boolean capabilities are Allowed, thus leaving them configurable by the SO. The remainder of the elements are integer values with either the default value or the maximum in number of bits shown.

1.4.6.1. Cryptographic Module Capabilities

The following is the set of capabilities supported at the cryptographic module level:

- Allow/disallow non-FIPS algorithms available.
- Allow/disallow password authentication (disallowed in TOE configuration).
- Allow/disallow trusted path authentication (allowed and must be enabled in TOE configuration).
- Allow/disallow M of N.
- Allow/disallow cloning.
- Allow/disallow masking (disallowed in TOE configuration).
- Allow/disallow M of N auto-activation.
- Allow/disallow ECC mechanisms.
- Allow/disallow Remote Authentication.
- Allow/disallow SO reset of partition PIN.
- Allow/disallow network replication.
- Allow/disallow forcing change of User authentication data.
- Number of failed SO logins allowed before the HSM is zeroized (set to 3, non-configurable).

1.4.6.2. Partition Capabilities

The following is the set of capabilities supported at the partition level:

- Allow/disallow partition reset.
- Allow/disallow activation.
- Allow/disallow automatic activation.
- Allow/disallow High Availability.
- Allow/disallow multipurpose keys.
- Allow/disallow changing of certain key attributes once a key has been created.
- Allow/disallow operation without RSA blinding.
- Allow/disallow signing operations with non-local keys.
- Allow/disallow raw RSA operations.
- Allow/disallow private key wrapping (disallowed in TOE configuration).
- Allow/disallow private key unwrapping.
- Allow/disallow secret key wrapping

- Allow/disallow secret key unwrapping.
- Allow/disallow Level 3 operation without a challenge (disallowed in TOE configuration).
- Allow/disallow user key management capability. (Allowed in TOE configuration. This would be disabled by the SO at the policy level to prevent any key management activity in the partition, even by a user in the CO role. This could be used, for example, at a CA once the root signing key pair has been generated and backed up, if appropriate, to lock down the partition for signing use only.)
- Allow/disallow incrementing of failed login attempt counter on failed challenge response validation.
- Allow/disallow RSA signing without confirmation.
- Allow/disallow RA type wrapping (disallowed in TOE configuration).
- Minimum/maximum password length (not applicable in TOE configuration).
- Level of storage space available for key storage (4 bits).
- 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 following capabilities are only configurable if cloning is allowed and enabled at the cryptographic module level:

- Allow/disallow private key cloning (allowed in TOE configuration).
- Allow/disallow secret key cloning (allowed in TOE configuration).

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

- Allow/disallow private key masking (disallowed in TOE configuration).
- Allow/disallow secret key masking (disallowed in TOE configuration).

1.4.7. Backup and Restoration

In order to support backup and transparent recovery of the cryptographic keys and supporting data stored within the Luna® CA4 cryptographic module, an optional backup and restoration capability can be employed. Each Luna® CA4 cryptographic module may have its cryptographic objects backed up using the Luna® Key Cloning protocol to a second Luna® CA4, acting as a backup token. Conversely, the cryptographic objects stored on a backup token may be restored to a properly initialised Luna® CA4 cryptographic module when bringing it into service.

1.4.8. Firmware Upgrade

The Luna® CA4 cryptographic module provides a capability to upgrade the module's firmware. Only the SO can perform a firmware upgrade. Each valid firmware upgrade package is digitally signed by SafeNet and encrypted. The customer must possess the correct authorization code in order to open the firmware upgrade package and the signature must be verified by the cryptographic module before the module will accept the upgrade.

1.4.9. User and Administrator Guidance Documentation

User and Administrator Guidance documentation is provided through the Luna® on-line help system provided to the customer on CD-ROM as part of the delivered TOE.

1.4.10. Environment

The TOE is used as the cryptographic module for a customer application (either customer-developed or third-party). As such, its environment is determined by the details of the customer deployment and it is, therefore, difficult to predict the complete description of the environment in which the TOE will operate. There are, however, a few characteristics common to any deployment.

The client PKCS #11 Cryptographic API software is provided, as part of the TOE, in the form of a Windows DLL or Unix-type shared-object library, depending on the host platform configuration. It runs within the host environment and provides the programming interface to the host software application, which normally acts as the user of the TOE.

The TOE is normally operated in a physically secured environment by users who have been specifically authorized to do so by the owning organization. Because the TOE is typically used within a larger system, such as a Public Key Infrastructure (PKI), as part of a Certification Authority (CA) or Certification Service provider (CSP), the environment will often also include a variety of hardware, software, telecommunications and networking devices as well as uninterruptible power supplies and environmental controls.

Note also that the TOE does not, in itself, provide full security audit functionality. It does export the raw data (with identifying sequence numbers but without time stamps) needed to compile an audit record. If security audit is required for the system within which the TOE is operating, the host IT environment must provide the means of recording security relevant events, so as to assist an administrator in the detection of potential attacks or mis-configuration of the system, of which the TOE is a part, that could leave the host system and/or the TOE itself susceptible to attack, and also to hold users accountable for any actions they perform that are relevant to the security of the system.

2. CONFORMANCE CLAIMS

2.1. CC Conformance Claim

Version: Part 1 Common Criteria Version 3.1R1

Part 2 and Part 3 Common Criteria Version 3.1R2

ST conformance:

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

- FDP_BKP.1 (Backup and restoration)

2) CC Part 3 conformant – 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)

2.2. Protection Profile Conformance Claim

This ST does not claim conformance to any Protection Profile.

3. SECURITY PROBLEM DEFINITION

3.1. Assumptions

A.Admin *Trustworthy TOE Administration*

When in operation, it is assumed that there will be a competent authority assigned to manage the TOE and the security of the information that it contains and who can be trusted not to deliberately abuse their privileges so as to undermine security. She/he is, however, capable of error.

A.Audit_Support *Data review*

It is assumed that a competent authority within the TOE environment reviews the raw data generated and exported by the TOE to generate any audit records required by the policy in place in the environment.

A.Backup_Data_Availability *Storage and Handling of TOE Backup Data*

The TOE environment ensures the availability of the backup data.

A.Controlled_Access *Physical Security Controls*

When in operation and when stored as a backup, the TOE is assumed to be located within a controlled access facility providing physical security that is adequate to prevent physical access by unauthorized persons.

A.Correct_Data *Correct Content Data*

The data submitted to the TOE by the host application is assumed to be correct. This requires that the data (e.g. certificate content data) has been generated and formatted correctly and maintains this correctness until it is passed to the TOE.

A.Human_Interface *Interface with Human Users*

The host application will provide an appropriate interface and communication path between human users and the TOE because the TOE does not have a human interface for authentication and management services. The TOE environment transmits identification, authentication and management data of TOE users correctly and in a confidential way to the TOE.

A.Legitimate_FW_Update *Legitimate Firmware Update Signed by the Vendor*

It is assumed that legitimate firmware update packages are digitally signed by the vendor using a private key whose use is restricted to this purpose and that the digital signature is verifiable by an instance of the TOE.

A.User_Authentication *Authentication of Users*

In the most general case, the host application software is assumed to be operating as the TOE user on behalf of a human user. As such, any direct interaction with the TOE, including authenticating, is performed by the host application as the user of the TOE. Individual human users authorised to access the TOE cryptographic services may not be known to the TOE itself. In this case, the TOE environment performs identification and authentication of the individual users and allows successfully authenticated users to use the host application as their agent for the cryptographic services.

A.User_Management *User Management*

The TOE will not, in general, be aware of the identities of end-users authorised for the TOE services. It is assumed that the management of the individual user assignments for the three TOE roles is done in the environment in a trustworthy fashion according to a well-defined policy.

3.2. Threats

Threat agents may include both unauthorized and authorized[1] users (persons or software entities) acting out of deliberate intent or through errors and omissions. Threat agents may also include malicious code of varying levels of sophistication, many of which are readily available on the Internet.

Relevant expertise required by threat agents may be in general semiconductor technology, software engineering and hacking techniques, and the resources may range from personal computers and peripherals to general-purpose test and measurement devices.

Motivation may include economic reward, a desire to damage an organization or the satisfaction and notoriety of defeating expert security.

Unauthorized users are assumed to be moderately motivated and to have a low to moderate level of relevant expertise and a low level of access to required resources. Attack potential for unauthorized users would be rated as Enhanced-Basic.

Authorized users could be assumed to be highly motivated (if they are acting deliberately) and to have at least a moderate level of relevant expertise and a high level of access to the required resources. Although authorized users are assumed to be trustworthy and acting in accordance with organizational operational and security policies and are, therefore, not considered attackers per se, the impact of errors and omissions could be rated as equivalent to an attacker with High attack potential.

3.2.1. Threat Statements

T.Bad_FW_Load *Loading Malicious Firmware into the TOE*

An authorized user or an unauthorized user who has gained access to the TOE may modify the existing firmware by loading unauthorized code and, thereby, compromise the security functions of the TOE.

T.Exchange *Compromise of Exchanged Data*

An unauthorized entity may gain access to sensitive user data exchanged between the TOE and other IT entities and cause unauthorized/undetected disclosure of and/or modifications to the data being exchanged.

T.Init_Data_Errors *Insecure Initialization of the TOE*

An authorized person or an unauthorized person accidentally or deliberately places the TOE in an untrusted state or otherwise compromises the security functions of the TOE by making an error in inputting initialization data.

T.Key_Discover *Finding All or Part of a Secret or Private Key*

An authorized user uses logical and/or physical attacks, including through weaknesses in key management functions, on the TOE to discover all or part of a secret or private key

T.Malfunction *Malfunction of TOE*

Internal malfunction or other unexpected interruption during the initialization or operation of the TOE could result in the modification of sensitive user data, misuse of TOE services, disclosure of key material or denial of service for authorized users.

T.Misuse_Management *Misuse of Management Services*

An authorized person may misuse the TOE administrative services, including attempts to exploit weak policy settings for the TOE, in order to forge user data and manipulate TOE security data and/or services.

T.Misuse_Sign *Misuse of signature-creation function*

A user of the host application or of the TOE misuses the TOE service for signature-creation to sign forged documents or transactions.

[1] Authorized user in this context means a person or software entity with organizational permission to access the TOE and its services.

T.PIN_Compromise *Compromise of Authentication data*

An unauthorized user may gain access to the authentication data of authorized users by intercepting the authentication data as it is entered or by guessing weak authentication data, and may impersonate an authorised user of the TOE.

T.Unauth_Function *Exploiting Unauthorized Functions*

An unauthorized person who has gained access to the TOE may reveal, discover or modify security data within the TOE by exploiting unauthorized functions of the TOE[2].

3.3. Organisational Security Policies

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

Only algorithms and algorithm parameters (e. g. key length) approved for use in FIPS PUB 140-2 validated products shall be employed in the TOE for key management, encryption/decryption, authentication, and signature generation/verification operations.

[2] An unauthorized function in this context is one which is not permitted in authorized versions of the TOE firmware, but which an unauthorized person is able to execute by causing malicious code to execute through some form of attack, such as a buffer overflow attack.

4. SECURITY OBJECTIVES

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

4.1. Security Objectives for the TOE

O.Admin *TOE Administration*

The TOE must provide facilities to enable the Security Officer (SO) to effectively manage the TOE and its security functions, and must ensure that only authorized users are able to access such functionality.

O.Approved_Algorithms *Use of Approved Algorithms*

The TOE must implement FIPS PUB 140-2 approved cryptographic algorithms and algorithm parameters (e.g., key length) for key management, encryption/decryption, authentication, and signature generation/verification operations.

O.Auth_Data_Protect *Protection of User Authentication data by the TOE*

The TOE must protect authentication data in a way that ensures that user authentication data cannot be easily guessed or captured.

O.Backup *Backup and Restoration for the TOE*

The TOE shall protect the confidentiality of the backup data and detect loss of the integrity of the backup keys, other user data and TSF data needed to restore an operational state after failure.

O.Check_Operation *Check for Correct Operation*

The TOE shall perform self-tests during installation, start-up and at the request of the authorised user to check that its critical hardware and firmware components operate correctly and shall ensure the tests succeed before allowing any privileged or cryptographic operations to proceed.

O.Control_Access *Access Control for Data and TOE Services*

The TOE shall control access to mediated commands, information, and services based on a user's identification and role, the requested command or service and the security attributes associated with the object to which access is requested.

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

The TOE shall provide integrity and confidentiality protection measures for all user data requiring integrity or confidentiality protection when it is exported from or imported to the TOE.

O.Detect_Attack *Tamper Resistance and Detection*

The TOE must be constructed in a manner that resists physical tampering and attempts at probing and shall detect tamper events and securely destroy all plaintext key material and other security critical data in this case.

O.Import_Code *Prevention of Unauthorised Code Loading*

The TOE must prevent executable code from being loaded on the TOE unless it is signed as per an authorized firmware update.

O.Key_Secure *Secure Key and Key Pair Generation and Management*

The TOE shall protect the confidentiality and integrity of cryptographic keys throughout their entire life-cycle. The TOE shall ensure secure key and key pair generation, use and management.

O.Multi-Person_Control *Multi-person Control of Sensitive Functions*

The TOE must provide a capability for multi-person control of sensitive functions.

O.Secure_Init *Secure Initialisation of TOE*

The TOE must assume its initial secure state immediately upon power-up, reset, or after other restart conditions.

O.Self_Protect *TOE Logical Self-Protection*

The TOE must protect itself against attempts to logically subvert or bypass the TOE security functions.

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

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

O.User_Data_Protect *Protection of User Data by the TOE*

The TOE shall protect the confidentiality and integrity of user data stored within the TOE and shall provide the means for the user to verify the authenticity of stored data.

4.2. Security Objectives for the Operational Environment

O.ENV_Application *Security in the Host application*

The applications which use the TOE shall perform the necessary security checks on the data passed to the TOE. The applications shall also perform the required user authentication and access control functions that cannot be performed within the TOE. Security controls in the TOE environment shall also prevent unauthorised manipulation of data submitted to the TOE.

O.ENV_Audit *Audit Trail Storage Availability*

The environment ensures the availability and provides a review of the audit trail generated in the environment based on data from the TOE.

O.ENV_Auth_Data *Personal Protection of Authentication Data*

Those responsible for the TOE must ensure that the authentication data for each user account for the TOE is held securely and not disclosed to persons not authorized to use that account.

O.ENV_Backup *Backup and Restoration Protection in the Environment*

The IT environment shall provide a means to protect the confidentiality of the backup data and detect loss of the integrity of the backup keys, other user data and TSF data needed to restore an operational state after failure when it is transmitted and stored in the TOE environment.

O.ENV_Outage_Protection *Protection From Unplanned Outages*

Those responsible for the host IT environment must ensure that the power supplied to the TOE is adequately protected against unexpected interruptions and the effects of surges and voltage fluctuations outside the normal operating range of the device and that the TOE is operated in an environment that is provided adequate protection against disasters such as fire and flood.

O.ENV_Personnel *Reliable Personnel*

The personnel using the TOE services shall be aware of civil, financial and legal responsibilities, as well as the obligations they have, depending on their role. The personnel shall be trained on correct usage of the TOE and have a level of competence sufficient to ensure the correct management and operation of the TOE.

O.ENV_Protect_Access *Prevention of Unauthorised Physical Access*

The TOE shall be protected by physical, logical and organisational protection measures to restrict access to the TOE and its IT environment to authorised persons only, in order to prevent any TOE theft, modification or disclosure of protected assets.

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

Recovery plans and procedures shall exist that allow a secure and timely recovery in the case of a major problem with the TOE. These procedures shall ensure that the confidentiality and integrity of TOE assets are maintained during recovery and that the recovery does not result in a situation that allows personnel to extend the TOE services they are allowed to use.

O.ENV_Secure_Init *Secure Initialisation Procedures*

Procedures and controls in the TOE environment shall be defined and applied to ensure secure set-up and initialisation of the TOE for operation.

O.ENV_Signed_FW_Update *Firmware Updates Signed by the Vendor*

Procedures shall exist to ensure that legitimate firmware update packages are digitally signed by the vendor using a private key whose use is restricted to this purpose and that the digital signature is verifiable by an instance of the TOE.

4.3. Security Objectives Rationale

Table 8-1, Table 8-2 and Table 8-3 demonstrate the necessity of the security objectives to address assumptions and threats and their appropriateness in countering the stated threats and providing for the stated assumptions.

5. EXTENDED COMPONENTS DEFINITION

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

5.1. FDP_BKP Backup and restoration:

Family Behaviour:

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

Component levelling:

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

Management: FDP_BKP.1

There are no management activities foreseen.

Audit: FDP_BKP.1

The following actions should be auditable if security audit is maintained by the host system:

a) Use of the backup function,

b) Use of the restoration function,

FDP_BKP.1 Backup and restoration

Hierarchical to: No other components.

FDP_BKP.1.1 The Security Officer or Crypto Officer shall be capable of invoking the backup function on demand.

FDP_BKP.1.2 The data stored in the backup shall be sufficient to recreate the state of the TOE at the time the backup was created using only the backup token.

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

FDP_BKP.1.4 Keys and other critical security parameters shall be transferred to the backup token in encrypted form only.

Dependencies: [FCS_CKM.1 Cryptographic key generation or

FCS_CKM.2 Cryptographic key distribution],

FCS_COP.1 Cryptographic operation,

FPT_ITC.1 Inter-TSF confidentiality during transmission,

FPT_ITI.1 Inter-TSF detection of modification,

FTP_ITC.1 (Key Cloning) Inter-TSF trusted channel

Rationale:

The HSM supports backup of key material and other user data and TSF data to restore the operational state of the system from a backup token in the event of a system failure or other serious error. The export, import and protection of the backup data are combined in a specific way. The HSM ensures the confidentiality of the backup data. The availability of the backup data will be ensured by the TOE environment.

This component is necessary to specify a unique requirement of certificate issuing and management components that is not addressed by the Common Criteria. The specific requirements address the protection of cryptographic keys and TSF data for backup and recovery.

6. SECURITY REQUIREMENTS

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

Security functional requirements components given in section 6.1 TOE Security Functional Requirements are drawn from the Common Criteria (ISO 15408), Version 3.1, Part 2 [2]. One security functional requirement represents an extension to [2]. This extended requirement is defined in section 5.1, with the rationale given in section 5.1. Operations for assignment, selection and refinement have been made.

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

The following convention is used to indicate operations that have been performed on the CC functional components:

- Assignment is indicated by **bold** lettering.
- Selection is indicated by underlining the selection(s).
- Refinement is indicated by *italic* lettering.
- Iterations are indicated by supplementary bracketed information with the functional component, such as **FIA_AFL.1.1 (SO) and FIA_AFL.1.1 (User).**

6.1. TOE Security Functional Requirements

6.1.1. Cryptographic support (FCS)

6.1.1.1. FCS_CKM.1 Cryptographic key generation

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

(1) RSA 1024, 2048, 4096 bits key pairs in accordance with ANSI X9.31.
(2) TDES 168 bits (security strength) in accordance with NIST SP 800-67
(3) AES 128, 192, 256 bits in accordance with FIPS PUB 197.
(4) DSA 1024 bits key pairs in accordance with FIPS PUB 186-3.
(5) ECDSA in accordance with FIPS PUB 186-3.

6.1.1.2. FCS_CKM.2 (Backup) Cryptographic key distribution

FCS_CKM.2.1 (Backup) The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **key agreement** that meets the following: **Luna® Key Cloning protocol**.

Application note:

The Luna® Key Cloning Protocol is used to negotiate a session key to encrypt the data transferred between the primary and the backup token. The protocol uses cryptographic techniques to provide mutual authentication, proof of origin, integrity and confidentiality of the objects being transferred from source to target token 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 by ensuring that a unique AES key is used for each cloning operation.

6.1.1.3. FCS_CKM.2 (FW Update) Cryptographic key distribution

FCS_CKM.2.1 (FW Update) The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **key exchange** that meets the following: **Luna® firmware update protocol**.

6.1.1.4. FCS_CKM.3 Cryptographic key access

FCS_CKM.3.1 The TSF shall perform **key access** in accordance with a specified cryptographic key access method, **return of a key handle,** that meets the following: **PKCS #11 standard**.

6.1.1.5. FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **logical or physical (by overwriting) deletion of the memory space** that meets the following: **FIPS 140-2 Level 3.**

6.1.1.6. FCS_COP.1 (SIGN) Cryptographic operation - Digital signature

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

(1) RSA 1024 bits, 2048 bits, 4096 bits with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (PKCS #1 V1.5, PKCS #1 PSS),
(2) RSA 1024 bits, 2048 bits, 4096 bits with SHA-1 (ANSI X9.31),
(3) DSA 1024 bits with SHA-1 (FIPS PUB 186-3),
(4) ECDSA with SHA-1, 224, 256, 384, 512 (FIPS PUB 186-3 Appendix D recommended curves).

6.1.1.7. FCS_COP.1 (DIGEST) Cryptographic operation - Message digest

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

(1) SHA-1 (FIPS PUB 180-3),

(2) SHA-224 (FIPS PUB 180-3),

(3) SHA-256 (FIPS PUB 180-3),

(4) SHA-384 (FIPS PUB 180-3),

(5) SHA-512 (FIPS PUB 180-3).

6.1.1.8. FCS_COP.1 (RSA ENC/DEC) Cryptographic operation - RSA Encrypt/Decrypt

FCS_COP.1.1 (RSA ENC/DEC) The TSF shall perform **asymmetric encryption and decryption** in accordance with a specified cryptographic algorithm **RSA** and cryptographic key sizes **1024 bits, 2048 bits and 4096 bits** that meet the following: **PKCS #1 V1.5 and OAEP**.

Application note:

RSA encryption/decryption is only used by the TOE to support the encrypted export/import of keys as covered by FDP_ETC.1 and FDP_ITC.1.

6.1.1.9. FCS_COP.1 (TDES ENC/DEC) Cryptographic operation - TDES Encrypt/Decrypt

FCS_COP.1.1 (TDES Enc/Dec) The TSF shall perform **symmetric encryption and decryption** in accordance with a specified cryptographic algorithm **Triple DES (ECB and CBC mode)** and cryptographic key sizes **168 bits** (security strength) that meet the following: **NIST SP 800-67**.

6.1.1.10. FCS_COP.1 (AES ENC/DEC) Cryptographic operation - AES Encrypt, Decrypt

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

6.1.2. User data protection (FDP)

6.1.2.1. FDP_ACC.1 (TAC) Subset access control

FDP_ACC.1.1 (TAC) The TSF shall enforce the **Token Access Control (TAC) SFP** on **the following:**

(1) Device sessions (subjects).

(2) Private keys, public keys, secret keys, certificates, data objects.

(3) Operations:

- **a. Read (Query Attribute Value)**
- **b. Modify**
- **c. Destroy**
- **d. Generate**[3]
- **e. Wrap (export)**
- **f. Use**[4]
- **g. Clone**

6.1.2.2. FDP_ACF.1 (TAC) Security attribute based access control

FDP_ACF.1.1 (TAC) The TSF shall enforce the **Token Access Control (TAC) SFP** to objects based on **the following:**

(1) Subject attributes:

- **a. Session and Access ID**
- **b. User ID associated with session (Access Owner)**
- **c. Role.**

(2) Object attributes:

- **a. Private. If True, object is Private. If False, object is Public.**
- **b. Owner. Object ownership is assigned to the object creator.**
- **c. Sensitive. If True, object is Sensitive. If False, object is Non-Sensitive.**
- **d. Extractable. If True, object may be extracted. If False, object may not be extracted.**
- **e. Modifiable. If True, object may be modified. If False, object may not be modified.**
- **f. Type. Public key, private key, secret key.**

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

A subject may perform an allowed operation on an object if one of the following two conditions

[3] 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.

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

holds:

(1) The object is a “Public” object, i.e., the PRIVATE attribute is FALSE, or

(2) The User ID of the subject is the same as the object’s owner.

An allowed operation is one permitted by the object attribute definitions within the constraints of the HSM and Partition level capability and policy settings. Table 6-2 summarizes the operations allowed by the object attribute settings.

The operations allowed for each user type in Table 6-2 have been abbreviated as indicated in Table 6-1 below.

Table 6-1 – Operation Abbreviations

Operation	Abbreviation	Operation	Abbreviation
Clone	C	Read	R
Destroy	D	Use	U
Generate	G	Wrap	W
Modify	M		

Table 6-2 – Access Matrix

Object Attribute				Subject (ID/Role)		
Private	Sensitive	Modifiable	Extractable	User ID/CO	User ID/CU	Public/Nil
0	0	0	N/A	C,D,G,R	R	D,G,R
0	0	1	N/A	C,D,G,M,R	R	D,G,M,R
1	0	0	N/A	C,D,G,R	R	---
1	0	1	N/A	C,D,G,M,R	R	---
1	1	0	0	C,D,G,R(1),U	R(1),U	---
1	1	1	0	C,D,G,M,R(1),U	R(1),U	---
1	1	0	1	C,D,G,R(1),U,W	R(1),U	---
1	1	1	1	C,D,G,M,R(1),U,W	R(1),U	---

1. The plaintext value of key material stored in objects whose CKA_SENSITIVE attribute is set cannot be read although other object values, such as the object label, are accessible.
2. A “0” in the table entry indicates that the attribute labelling the column has not been set. A “1” indicates that the attribute has been set.

FDP_ACF.1.3 (TAC) The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **None**.

FDP_ACF.1.4 (TAC) The TSF shall explicitly deny access of subjects to objects based on the **following additional rules**:

- **A subject shall not have access to the plaintext value of an object whose CKA_SENSITIVE attribute is set.**

6.1.2.3. FDP_BKP.1 Backup and restoration

FDP_BKP.1.1 The Security Officer or Crypto Officer shall be capable of invoking the backup function on demand.

FDP_BKP.1.2 The data stored in the backup shall be sufficient to recreate the state of the TOE at the time the backup was created using only the backup token.

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

FDP_BKP.1.4 Keys and other critical security parameters shall be transferred to the backup token in encrypted form only.

Application Note:

The SHA-1 checksum is calculated by the TSF and by the equivalent function within the backup token. It can be compared before and after backup and before and after recovery to validate integrity. Because it is done within the TSF and backup token, there is no chance of substituting an illegitimate key value and corresponding digest value. A keyed hash or digital signature is, therefore, not required.

6.1.2.4. FDP_DAU.1 Basic data authentication

FDP_DAU.1.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of **user objects**.

FDP_DAU.1.2 The TSF shall provide the **Users** with the ability to verify evidence of the validity of the indicated information.

Application Note:

The evidence is provided by the SHA-1 fingerprint of the object. It is generated by the TOE and can be queried at any time by the user.

6.1.2.5. FDP_DAU.2 Data authentication with identity of guarantor

FDP_DAU.2.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of **private and public keys**.

FDP_DAU.2.2 The TSF shall provide the **Users** with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence.

Application Note:

The evidence is provided by the Public Key Confirmation (PKC). It is generated in an X.509 certificate format by the TOE and is signed by either the internal Hardware Origin Key (HOK), whose public key is certified by the SafeNet trust anchor or by a customer private key, whose public key certificate has been issued by a third party CSP or Trust Centre.

6.1.2.6. FDP_ETC.1 Export of user data without security attributes

FDP_ETC.1.1 The TSF shall enforce the **Token Access Control (TAC) SFP** when exporting user data, controlled under the SFPs, outside of the TOE.

FDP_ETC.1.2 The TSF shall export the user data without the user data's associated security attributes.

Application Note:

All user data whose CKA_SENSITIVE attribute is set is exported in encrypted form.

6.1.2.7. FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1 The TSF shall enforce the **Token Access Control (TAC) SFP** when importing user data controlled under the SFP from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data, controlled under the SFP, from outside the TOE*:* **The CKA_SENSITIVE attribute of the object storing user data imported via an Unwrap operation shall be set.**

6.1.2.8. FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the de-allocation of the resource from the following objects: **private keys, secret keys**.

6.1.2.9. 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 allocation of the resource to all objects.

6.1.3. Identification and authentication (FIA)

6.1.3.1. FIA_AFL.1 (SO) Authentication failure handling

FIA_AFL.1.1 (SO) The TSF shall detect when **three (3)** unsuccessful authentication attempts occur related to **Security Officer authentication**.

FIA_AFL.1.2 (SO) When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **change the state of the TOE to require re-initialization**.

6.1.3.2. FIA_AFL.1 (User) Authentication failure handling

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

FIA_AFL.1.2 (User) When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **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:

- removing the User and clearing the User’s memory space and permanent storage, or
- disabling the User account by setting the User locked flag in the User’s attributes (FIA_ATD.1).

6.1.3.3. FIA_ATD.1 User attribute definition

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

User ID number
User checkword (RAD)
User role
User failed login count
User “locked” flag.

Application Note:

User role is derived from the type of iKey used – Blue Key for SO and Black Key for User, plus the type of Challenge-response authentication – either Crypto Officer or Crypto User. Each has a different challenge secret, which is stored encrypted by the User Security Key (retrieved from the encrypted checkword).

6.1.3.4. FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **the minimum length established by the TOE for each authentication secret**.

6.1.3.5. FIA_SOS.2 TSF generation of secrets

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet **the minimum lengths for each function for which they are required and that are random:**

- **SO and User PED authentication data 48 bytes**
- **User Challenge secret 75 bits**
- **M of N activation 32 bytes**
- **Cloning 24 bytes.**

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for **the following TSF functions:**

- **SO and User PED authentication**
- **M of N activation**
- **Cloning.**

6.1.3.6. FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow **the following actions** on behalf of the user to be performed before the user is authenticated:

- **Perform start-up, self-test (FPT_TST.1), detection of the secure blocking state (FPT_FLS.1), detection of violation of physical integrity (FPT_PHP.1),**
- **Perform basic diagnostic functions, such as checking the communications from the host to the card, checking firmware level and token info and checking information on mechanisms supported.**
- **Open a session**
- **Access Public data objects**
- **Identification (FIA_UID.1).**

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.

6.1.3.7. FIA_UAU.4 Single-use authentication mechanisms

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to **Challenge-response authentication for Users**.

6.1.3.8. FIA_UAU.5 Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide **the following authentication mechanisms** to support user authentication:

- **M of N secret sharing (SO and User)**.
- **iKey entry (SO and User)**
- **PIN entry (SO and User)**
- **Challenge-response (User)**

FIA_UAU.5.2 The TSF shall authenticate any user’s claimed identity according to **the following rules:**

- **The user must enter their authentication data using, at a minimum, the PED and an iKey.**
- **Optionally, the user enters a Personal Identification Number (PIN) via the PED in addition to the entering the iKey.**

- **If required by the policy defined for the TOE, M out of N secret shares must first be entered via the Luna® PIN Entry Device (PED) in order to enable the TOE for operation.**
- **The User, in the Crypto User and Crypto Officer roles, must enter the challenge secret corresponding to one of the roles via the application interface in order to access cryptographic data and services.**

6.1.3.9. FIA_UID.1 Timing of identification

FIA_UID.1.1 The TSF shall allow **the following actions** on behalf of the user to be performed before the user is identified:

- **Perform start-up, self-test (FPT_TST.1), detection of the secure blocking state (FPT_FLS.1), detection of violation of physical integrity (FPT_PHP.1),**
- **Perform basic diagnostic functions, such as checking the communications from the host to the card, checking firmware level and token info and checking information on mechanisms supported.**

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

6.1.3.10. 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:

- **User ID**
- **User checkword**
- **User role.**

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:

- **User ID is Public (unidentified)**
- **User checkword 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:

When the user is successfully authenticated the user security attributes change from their initial values to the values appropriate for that authenticated user.

6.1.4. Security Management

6.1.4.1. FMT_MOF.1 Management of security functions behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to disable, enable *and* modify the behaviour of the functions **listed below** to **the Security Officer role:**

HSM Level

M of N Activation – SO may enable and disable.
M of N Auto-activation – SO may enable and disable.
HSM Cloning – SO may enable and disable.
Remote Authentication – SO may enable and disable.

Network Replication – SO may enable and disable.
Force change of User authentication data – SO may enable and disable.

Partition Level

Partition reset – SO may enable and disable.

Partition activation – SO may enable and disable.

Partition auto-activation – SO may enable and disable.

High Availability – SO may enable and disable.

Multi-purpose keys – SO may enable and disable.

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

Operation without RSA blinding – SO may enable and disable.

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

Performing raw RSA operations – SO may enable and disable.

Private key unwrapping – SO may enable and disable.

Secret key wrapping – SO may enable and disable.

Secret key unwrapping – SO may enable and disable.

User key management capability – SO may enable and disable.

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

RSA signing without confirmation – SO may enable and disable.

6.1.4.2. FMT_MSA.1 (Object Attributes) Management of security attributes

FMT_MSA.1.1 (Object Attributes) The TSF shall enforce the **TAC SFP** to restrict the ability to <u>modify</u> 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.**

6.1.4.3. FMT_MSA.2 (Object Attributes) Secure security attributes

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

6.1.4.4. FMT_MSA.3 (Object Attributes) Static attribute initialization

FMT_MSA.3.1 (Object Attributes) The TSF shall enforce the **TAC SFP** to provide <u>restrictive</u> default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 (Object Attributes) The TSF shall allow the **Crypto Officer** to specify alternative initial values to override the default values when an object or information is created.

6.1.4.5. FMT_MTD.1 (Login Failures) Management of TSF data

FMT_MTD.1.1 (Login Failures) The TSF shall restrict the ability to <u>change_default</u> the **Number of User Login Failures Allowed (FIA_AFL.1.1 (User))** to the **Security Officer.**

6.1.4.6. FMT_MTD.1 (UAV – User Locked Flag) Management of TSF data

FMT_MTD.1.1 (UAV – User Locked Flag) The TSF shall restrict the ability to <u>change_default</u>, <u>query</u>, <u>modify</u> *and* <u>delete</u> the **User Locked Flag** to **the Security Officer role.**

6.1.4.7. FMT_MTD.1 (UAV – Challenge Secret – Crypto Officer) Management of TSF data

FMT_MTD.1.1 (Challenge Secret – Crypto Officer) The TSF shall restrict the ability to modify the **Crypto Officer Challenge Secret** to **the Crypto Officer role.**

6.1.4.8. FMT_MTD.1 (SOV) Management of TSF data

FMT_MTD.1.1 (SOV) The TSF shall restrict the ability to change_default *and* modify the **SO Checkword** to **the Security Officer role.**

6.1.4.9. FMT_SMF.1 Specification of Management Functions

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

1. **User management,**
2. **disable, enable and modify the behaviour of configurable policy settings at the HSM and Partition levels (FMT_MOF.1),**
3. **change_default the Number of User Login Failures Allowed.**

6.1.4.10. FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles **Security Officer, Crypto Officer, Crypto User**.

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

Application note:

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

6.1.5. Protection of the TOE Security Functions (FPT)

6.1.5.1. FPT_FLS.1 Failure with preservation of secure state

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

6.1.5.2. FPT_ITC.1 Inter-TSF confidentiality during transmission

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

Application note:

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

6.1.5.3. FPT_ITI.1 Inter-TSF detection of modification

FPT_ITI.1.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric: **SHA-1 digest**.

FPT_ITI.1.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform **error indication[5] to the Crypto Officer** if modifications are detected.

Application note:

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

The SHA-1 checksum is calculated by the TSF and by the equivalent function within the backup token. It can be compared before and after backup and before and after recovery to validate integrity. Because it is done within the TSF and backup token, there is no chance of substituting an illegitimate key value and corresponding digest value. A keyed hash or digital signature is, therefore, not required.

6.1.5.4. FPT_PHP.1 Passive detection of physical attack

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

6.1.5.5. FPT_RCV.1 Manual recovery

FPT_RCV.1.1 After **a failure or service discontinuity[6]**, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

6.1.5.6. FPT_TST.1 TSF testing

FPT_TST.1.1 The TSF shall run a suite of self-tests during initial start-up *and* at the request of the authorised user to demonstrate the correct operation of the TSF.[7]

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

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

6.1.6. Resource utilization (FRU)

6.1.6.1. FRU_FLT.1 Degraded fault tolerance

FRU_FLT.1.1 The TSF shall ensure the operation of **TOE's user data protection capabilities** when the following failures occur: **power failure or data input/output failure**.

[5] The TSF notifies the Crypto Officer by means of error indicators rather than aural or visual alarm signals.

[6] For the TOE, failure in this context refers to self-test failure. Any other failure would be catastrophic, leaving the TOE in a secure but non-recoverable state. In the case of a service discontinuity, the module will always return to service in a secure state. The details of its operational state when it returns to service are determined by the configurable policy set by the SO.

[7] The selection "at the conditions installation and maintenance" has been removed because there are no states of the TOE that correspond to installation and maintenance.

6.1.7. Trusted path (FTP)

6.1.7.1. FTP_TRP.1 Trusted path

FTP_TRP.1.1 The TSF shall provide a communication path between itself and local users that is *physically and* logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

FTP_TRP.1.2 The TSF shall permit local users to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for initial user authentication, **the following additional functions**:

- **Upload of the TSF-generated authentication data to the iKey**
- **Upload of the TSF-generated challenge secret to the PED display**
- **Entry of M of N Activation secret shares**
- **Entry of the Token Cloning Domain key**

6.1.7.2. FTP_ITC.1 (FW Update) Inter-TSF trusted channel

FTP_ITC.1.1 (FW Update) 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 (FW Update) The TSF shall permit another trusted IT product to initiate communication via the trusted channel.

FTP_ITC.1.3 (FW Update) The TSF shall initiate communication via the trusted channel for **firmware load and update**.

6.1.7.3. FTP_ITC.1 (Key Cloning) Inter-TSF trusted channel

FTP_ITC.1.1 (Key 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 (Key Cloning) The TSF shall permit the TSF, another trusted IT product to initiate communication via the trusted channel.

FTP_ITC.1.3 (Key Cloning) The TSF shall initiate communication via the trusted channel for **key cloning**.

Application Note:

The function specified by FTP_ITC.1 (Key Cloning) is used to establish a trusted channel between the TOE and the Backup Token. The session key used to encrypt the data being backed up or restored is negotiated as part of the establishment of the trusted channel.

6.2. TOE Security Assurance Requirements

The assurance requirements for this TOE are as specified in the Common Criteria Version 3.1 Part 3-EAL 4 package with augmentation. The EAL 4 package has been augmented by the addition of the Part 3 requirements: ALC_FLR.2.

6.2.1. Security Assurance Requirements Augmentation to EAL 4

6.2.1.1. ALC_FLR.2 Flaw reporting procedures

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.

6.3. IT Security Requirements Rationale

Table 8-4 shows the necessity of the Security Functional Requirements and Table 8-5 maps Security Functional Requirements to Security Objectives and provides the rationale that the SFRs, singly or in combination, meet the Security Objectives. Table 8-6 demonstrates that all dependencies for the SFRs have been met.

6.3.1. Extended Security Requirements

This Security Target specifies one Extended Security Functional Requirements, as follows:

FDP_BKP.1 – This requirement is explicitly stated in this ST to address a unique requirement for the TOE to be capable of performing a secure backup of cryptographic material that can be used in the recovery of the host processing environment.

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 Certification Service Provider (CSP) 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 CSP 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 zeroized 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.

6.5. Assurance Measures

Table 8-7 – Assurance Measures shows each of the security assurance requirements of the TOE and maps each to the applicable assurance evidence provided for the evaluation.

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 Certification Service Provider (CSP), 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 CSP when signing digital certificates.

The Luna® CA4 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 exists in the Luna® Interface Control Document [8].

7.1.2. Multi-Session Capability

The Luna® CA4 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 have multiple applications each opening various 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) – authorized 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 – authorized to create, use, destroy and backup/restore cryptographic objects.
- Crypto User – authorized to use cryptographic objects (e.g., sign, encrypt/decrypt).

The Crypto Officer and Crypto User interface to the Luna® CA4 for cryptographic operations using the PKCS #11 API. 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® CA4 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 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 ensuring that sessions cannot be shared among users. 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® CA4 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. See section 7.3.12 for a description of the specific policy elements that are configurable by the SO in the TOE configuration.

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 are truly fixed for all product variants and, therefore, are never subject to configuration by the SO. The 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 "allow/disallow some functionality" are Boolean values where false (or zero) equates to disallow the functionality and true (or one) equates to allow the functionality. Except as noted, all Boolean capabilities are Allowed, thus leaving them configurable by the SO. The remainder of the elements are integer values with either the default value or the maximum in number of bits shown.

7.2.1. HSM Level Capabilities

The following is the set of capabilities supported at the HSM level:

- Allow/disallow non-FIPS algorithms available.
- Allow/disallow password authentication (disallowed in TOE configuration).
- Allow/disallow trusted path authentication

- Allow/disallow M of N.
- Allow/disallow cloning.
- Allow/disallow masking (disallowed in TOE configuration).
- Allow/disallow M of N auto-activation.
- Allow/disallow ECC mechanisms.
- Allow/disallow Remote Authentication.
- Allow/disallow SO reset of partition PIN.
- Allow/disallow network replication.
- Allow/disallow forcing PIN change.
- Number of failed SO logins allowed before the HSM is zeroized (set to 3, non-configurable).

7.2.2. Partition Level Capabilities

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

- Allow/disallow partition reset.
- Allow/disallow activation.
- Allow/disallow automatic activation.
- Allow/disallow High Availability.
- Allow/disallow multipurpose keys.
- Allow/disallow changing of certain key attributes once a key has been created.
- Allow/disallow operation without RSA blinding.
- Allow/disallow signing operations with non-local keys.
- Allow/disallow raw RSA operations.
- Allow/disallow private key wrapping (disallowed in TOE configuration).
- Allow/disallow private key unwrapping.
- Allow/disallow secret key wrapping
- Allow/disallow secret key unwrapping.
- Allow/disallow Level 3 operation without a challenge (disallowed in TOE configuration).
- Allow/disallow user key management capability. (Allowed in TOE configuration. This would be disabled by the SO at the policy level to prevent any key management activity in the partition, even by a user in the Crypto Officer role. This could be used, for example, at a CA once the root signing key pair has been generated and backed up, if appropriate, to lock down the partition for signing use only.)
- Allow/disallow incrementing of failed login attempt counter on failed challenge response validation.
- Allow/disallow RSA signing without confirmation.
- Allow/disallow RA type wrapping (disallowed in TOE configuration).
- Minimum/maximum password length (not applicable in TOE configuration).
- Level of storage space available for key storage (4 bits).
- Number of failed Partition User logins allowed before partition is locked out/cleared (default is 10, SO can configure it to be 3 <= N <= 10)

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

- Allow/disallow private key cloning (allowed in TOE configuration).
- Allow/disallow secret key cloning (allowed in TOE configuration).

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

- Allow/disallow private key masking (disallowed in TOE configuration).
- Allow/disallow secret key masking (disallowed in TOE configuration).

7.3. IT Security Functions

7.3.1. Trusted Path – Luna® PED II

User authentication data and other critical security parameters are protected through the use of a separate port and data path for their transfer, and by providing mechanisms to protect their confidentiality and integrity. Attached to this separate data port is the Luna® PIN Entry Device or Luna® PED II.

The Luna® PED II, with accompanying iKeys, is depicted in Figure 4. It houses a number of input/output interfaces that, in combination, provide a trusted path device for the communication of authentication data and critical security parameters to and from the Luna® CA4 cryptographic module. The Luna® PED II has a character display used to display status and prompt messages, and a challenge secret that is output by the cryptographic module at the time a partition is created [see sub-section 7.1.3]. It has a keypad used to enter simple responses (Yes/No/Enter) and to enter an optional PIN that is combined with the authentication data stored on an iKey as part of the authentication process. It has a USB receptacle for the input/output of data to the iKey and it has a serial communications cable that connects to the separate data port, which is wired directly to the cryptographic module. Because the PED II has a direct serial communications interface to the cryptographic module, only local entry of iKey authentication data is possible.

The following types of iKey are used with the Luna® PED II:

- Blue (SO) iKey – for the storage of SO authentication data,
- Black (User) iKey – for the storage of User authentication data,
- Red (Domain) iKey – for the storage of the cloning domain data, used to control the ability to clone from a cryptographic module to a backup token,
- Green (M of N) iKeys – used to store M of N secret shares, used for multi-purpose control of critical functions,

Any iKey, once data has been written to it, is an Identification and Authentication device and must be safeguarded accordingly by the administrative or operations staff responsible for the operation of the TOE within the customer's environment.

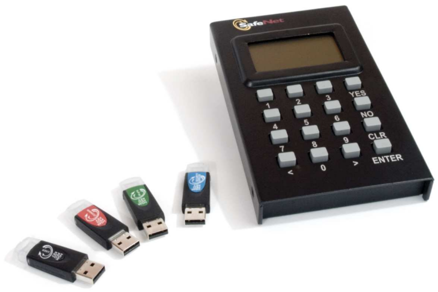

Figure 4. Luna® PED II with iKeys

7.3.2. User Identification and Authentication

The iKey contains the user’s identification number and the pseudo-randomly generated 48-byte authentication secret for the user and is entered into the key receptacle in the PED II in order to identify and authenticate the user.

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 and Crypto User. For a user to assume any role the module enforces user identification and authentication.

The TOE requires that all users (SO, Crypto Officer and Crypto User roles) be authenticated by proving knowledge of a secret shared by the user and the cryptographic module.

The TOE generates the authentication secrets using its PRNG. 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(s); these are generated at the time the partition is created by the SO. The authentication secret(s) are provided to the operator via the Luna® PED II display and iKey, as described in sub-section 7.3.1, and must be entered by the operator via the Luna® PED II 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 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 the Blue iKey data combined with the optional PIN through the trusted path.

User authentication is a two-stage process. The first stage is termed “Activation” and is performed using the Luna® PED II. Activation requires the transmission to the cryptographic module of the Black iKey data combined with the optional PIN through the trusted path. Once Activation has been performed, 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 typically 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 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 displayed as a 16-character string on the visual display of the trusted path device. The challenge secret is then provided, via a secure out-of-band means, to each external entity authorized 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”.

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® CA4 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 7.3.4.

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

In the case of a user:

> If “y” consecutive user 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 token.

In the case of the SO, if three (3) consecutive SO logon attempts fail, the module is zeroized and must be re-initialized.

7.3.2.1. M of N Activation

The TOE can also be configured by the SO to require the use of an M of N secret sharing authentication scheme to enable the module for operation. The M of N activation protocol provides the capability to enforce multi-person integrity over SO operations and activation of each partition.

The M of N capability is based on Shamir's threshold scheme. The Luna® CA4 cryptographic module generates a 32 byte secret and protects it by "splitting" it into "N" pieces and storing each piece on an iKey dedicated to that purpose (Green Key). Any "M" of these "N" pieces must be transmitted to the Luna® CA4 cryptographic module by inserting the corresponding iKeys into the Luna® PED II in order to reconstruct the original secret.

7.3.2.2. Unidentified and Unauthenticated Users

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

- Perform start-up, self-test (FPT_TST.1), detection of the secure blocking state (FPT_FLS.1), detection of violation of physical integrity (FPT_PHP.1),
- Perform basic diagnostic functions, such as checking the communications from the host to the card, checking firmware level and token info and checking information on mechanisms supported.

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:

- Perform start-up, self-test (FPT_TST.1), detection of the secure blocking state (FPT_FLS.1), detection of violation of physical integrity (FPT_PHP.1),
- Perform basic diagnostic functions, such as checking the communications from the host to the card, checking firmware level and token info and checking information on mechanisms supported.
- Open a session
- Access Public data objects
- Identification (FIA_UID.1).

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

7.3.3. Authentication Data Selection

The User authentication data is a 48 byte value that is randomly generated by the module and stored on an iKey (Blue for SO or Black for User) plus the Crypto Officer and Crypto User Challenge Secrets, which are initially provided to the user via the PED II display. The iKey represents the user to the module and, therefore, must be properly protected within the environment in which the module operates. A User, in the Crypto Officer role, and the SO can request to change their respective authentication data at any time using the Command Line Interface.

7.3.4. 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.

7.3.5. 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[8]
- Wrap (export)
- Use[9]
- Clone

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 labelled 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. Private keys are never extractable from the Luna® CA4 cryptographic module.

[8] 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.

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

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 and the access matrix in section 6.1.2.2.

7.3.6. 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 zeroized. When cryptographic functions are performed, a cryptographic context is created to hold data required by the function (e.g., a DES key schedule for a DES 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.

7.3.7. 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.

7.3.8. Key Export and Import Protection

Secret keys may only be exported from the TOE boundary in a wrapped (encrypted) form if the Extractable attribute is True. Private keys may never be exported from the TOE boundary. Secret keys are exported from the module without their associated security attributes. 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. Any attributes of keys imported in this way are ignored by the TOE and their attributes are set to default values by the TOE. 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.

7.3.9. Cryptographic Material Management

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

- Cryptographic key generation in accordance with the following indicated standards:
    - RSA 1024, 2048, 4096 bits key pairs in accordance with ANSI X9.31.
    - TDES 168 bits (security strength) in accordance with NIST SP 800-67.
    - AES 128, 192, 256 bits in accordance with FIPS PUB 197
    - DSA 1024 bits key pairs in accordance with FIPS PUB 186-3.
    - ECDSA in accordance with FIPS PUB 186-3.
- Secure key access following the PKCS #11 standard.
- Destruction of cryptographic keys in accordance with the FIPS PUB 140-2 Level 3 standard.

An object on the module that is destroyed using the PKCS #11 function C_DestroyObject (the user delete command available through the API) is marked invalid and remains encrypted with the user's secret key until such time as its flash locations are re-allocated for additional data on the module; at which time they are purged and zeroized before re-allocation. The same strategy of marking an object invalid and purging the memory content before re-allocation is followed for volatile memory as well as flash.

Objects on the module that are destroyed as a result of authentication failure are zeroized (all flash blocks in user's memory turned to 1's). If it is an SO authentication failure all flash blocks on the module are zeroized.

Objects on the module that are destroyed through C_InitToken (the SO function to initialize the module available through the API) are zeroized, along with the rest of the flash memory being used by the SO and User.

All cryptographic material management functions are performed in the module in accordance with the appropriate cryptographic standards using algorithms and mechanisms that have been formally validated as meeting the FIPS PUB 140-2 Level 3 standard.

7.3.9.1. Key Storage and Access Protection

Keys are always stored as secret key or private key objects with the Sensitive attribute set and, therefore, with the key value encrypted. 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. Key storage and access is performed in accordance with the PKCS #11 object model and function specifications.

7.3.10. Cryptography

Because of its generic nature, the Luna® CA4 cryptographic module firmware supports a wide range of cryptographic algorithms and mechanisms. The cryptographic functions and algorithms that are relevant to the TOE are the following:

- Random Number Generation
    - FIPS 140-2 validated Deterministic Random Bit Generator (Pseudo-random Number Generator) seeded by internal Hardware Non-deterministic Random Bit Generator
    - Based on ANSI X9.31, Appendix A section 2.4
- Compute Digital Signatures And Verify Digital Signatures
    - RSA 1024 bits, 2048 bits, 4096 bits (PKCS #1 V1.5, PKCS #1 PSS, ANSI X9.31) with SHA-1
    - RSA 1024 bits, 2048 bits, 4096 bits (PKCS #1 V1.5, PKCS #1 PSS) with SHA-224, 256, 384, 512
    - DSA 1024 bits (FIPS PUB 186-3) with SHA-1

  - ECDSA (FIPS PUB 186-3 Appendix D recommended curves) with SHA-1
- Encrypt / Decrypt Data
  - RSA 1024, 2048 and 4096 bits in accordance with PKCS #1 V1.5 and OAEP
  - TDES (ECB and CBC mode) 168 bits (security strength) in accordance with NIST SP 800-67
  - AES (ECB and CBC mode) 128 and 256 bits in accordance with FIPS PUB 197
- Export (Wrap) and Import (Unwrap) Secret Keys
  - TDES, AES with TDES and AES in ECB mode
  - TDES, AES with RSA 1024, 2048 and 4096 bits in accordance with PKCS #1 V1.5

The necessary keying material needed by these algorithms may be generated or derived on-board. Random data needed to produce sound key material is generated by the module’s PRNG. In some cases, key material may be imported from an external source in an encrypted (wrapped) form and decrypted (unwrapped) inside the module.

7.3.11. 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.

7.3.12. 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 7.3.13 and 7.3.14.

7.3.13. 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 following is the set of policies supported at the HSM level:

- Enable/disable non-FIPS algorithms available.
- Enable/disable trusted path authentication (allowed and must be enabled in TOE configuration).
- Enable/disable M of N.
- Enable/disable cloning.
- Enable/disable M of N auto-activation.
- Enable/disable ECC mechanisms.
- Enable/disable Remote Authentication.
- Enable/disable SO reset of partition PIN.
- Enable/disable network replication.
- Enable/disable forcing change of User authentication data.

The following is the set of policies supported at the partition level:

- Enable/disable partition reset.
- Enable/disable activation.
- Enable/disable automatic activation.
- Enable/disable High Availability.
- Enable/disable multipurpose keys.
- Enable/disable changing of certain key attributes once a key has been created.
- Enable/disable operation without RSA blinding.
- Enable/disable signing operations with non-local keys.
- Enable/disable raw RSA operations.
- Enable/disable private key unwrapping.
- Enable/disable secret key wrapping
- Enable/disable secret key unwrapping.
- Enable/disable user key management capability. (This would be disabled by the SO at the policy level to prevent any key management activity in the partition, even by a user in the Crypto Officer role. This could be used, for example, at a CA once the root signing key pair has been generated and backed up, if appropriate, to lock down the partition for signing use only.)
- Enable/disable incrementing of failed login attempt counter on failed challenge response validation.
- Enable/disable RSA signing without confirmation.
- Enable/disable private key cloning.
- Enable/disable secret key cloning.

7.3.14. 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 7.3.4, 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 Token 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.

7.3.15. 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.

7.3.15.1. Memory and Firmware Integrity Check

The firmware integrity is protected by an error detection code based on a Cyclic Redundancy Check (CRC) and a cryptographic hash function. The firmware's integrity is checked by the bootblock using the CRC when the firmware is initially loaded or updated and every time the module is started. The firmware also verifies the SHA-1 hash of the loaded firmware before it starts executing. 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.

7.3.15.2. Self-Tests

The TOE performs a number of tests of security-critical functions each time it is activated and on demand from a user. The TOE offers three categories of self-tests that can be called up by the user at any time: hardware, cryptographic and PRNG 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 and PRNG 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.

7.3.15.3. 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.

7.3.15.4. 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 7.3.18.

7.3.15.5. Firmware Loading and Firmware Update

The Luna® CA4 cryptographic module requires the use of a cryptographically protected trusted channel for initial firmware loading at the factory prior to delivery to the customer and when the firmware is later updated at the customer’s site. The trusted channel is provided as described in the following two paragraphs.

Firmware can only be initially loaded onto Luna® CA4 cryptographic module from a separate module dedicated for the purpose and containing a firmware image that has been digitally signed by SafeNet and encrypted using a secret key generated specifically for this purpose and if the module itself is a valid Luna® CA4 cryptographic module.

For firmware updates, the updated image is signed and encrypted using a dedicated module at SafeNet and distributed to customer sites in software form along with a separately distributed authorization code. The TOE verifies the digital signature to ensure that the updated image originated at SafeNet and that it has not been modified. The TOE decrypts the image using a key derived from the authorization code to ensure that its confidentiality has been protected while in transit. The use of the authorization code ensures that only authorized customers may perform the update and ensures that only valid Luna® CA4 cryptographic modules can decrypt the image in order to perform the update. The trusted channel for communicating the firmware image from the dedicated SafeNet module to the target Luna® CA4 cryptographic module is initiated by the dedicated module because it is the one that generates the symmetric keys and authorization code, digitally signs the image and encrypts it for transmission to the target. The actual firmware update process is performed by the target module and the first part of that process completes the communication that was initiated by the dedicated SafeNet module by verifying the digital signature of the image and decrypting it for loading into the TOE.

7.3.16. Cloning

For performance and secure backup purposes, Luna® CA4 cryptographic modules and Luna® backup tokens 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 via the PED II to other modules or backup tokens 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® CA4 and a corresponding CA4 token, 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 (backup token or another Luna® CA4 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 unauthorized 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 by ensuring that a unique AES key is used for each cloning operation.

7.3.17. Physical Self-Protection

Tamper-evident features are implemented in the manufacture of the module. Any tampering that might compromise the module's security can be detected by visually inspecting the physical integrity of the module. The module's physical design also resists visual inspection of the device design, physical probing of the device and attempts to access sensitive data on individual components of the module and provides evidence of the occurrence of such physical tampering. The module responds automatically to attempts to open its enclosure by ensuring that plaintext key material and other sensitive data is erased from the module.

7.3.18. 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.

7.3.19. Backup and Recovery

As described in sub-sections 7.3.15.4 and 7.3.18, the module maintains its secure state in the event of a failure. Depending on the nature of the failure, the module will maintain its secure state and resume operation as described below:

- In the event of host system discontinuity the module maintains its current logged in state and resumes that state when the host system restarts.
- In the event that power is lost to the module for a longer period of time, it maintains its secure state by maintaining the encryption of all sensitive data and it requires the User to activate the partition prior to resuming operation. It will resume operation with all security properties intact but the operational state of the module prior to loss of power will be lost.
- In the event of a catastrophic damage to or failure of the module itself, recovery is accomplished by inserting and activating a backup module, as described below.

The TOE provides the capability to securely backup a module using the cloning function. Because the cloning function securely duplicates all objects from the primary module to the backup token, the backup token allows recovery from the backup token by cloning the backed up objects to a new module that has been initialized with the same cloning domain. The basic data authentication mechanism described in section 7.3.7 can be used at both the TOE and the backup token before and after cloning operations to ensure the integrity of backed up and restored key objects.

7.4. Assurance Measures

The assurance requirements for this TOE are as specified in the EAL 4 package augmented by:

- ALC_FLR.2 (Flaw Reporting Procedures)

Evidence, in the form of documentation, plans and procedures that meet the content and presentation requirements of Part 3 of the Common Criteria [3], is provided to satisfy the specified assurance requirements. References to the appropriate supporting documentation are provided in Table 8-7 – Assurance Measures.

The evidence includes deliverables in the following categories:

1. Security Target
2. ST Rationale
3. Security Architecture (ADV_ARC.1)
4. Functional Specification (ADV_FSP.4)
5. Implementation Representation (ADV_IMP.1)
6. TOE Design (ADV_TDS.3)
7. Developer’s Tests (ATE_COV.2, ATE_DPT.2, ATE_FUN.1, ATE_IND.2)
8. Configuration Management (ALC_CMC.4, ALC_CMS.4)
9. Life-Cycle Documentation (ALC_DVS.1, ALC_LCD.1, ALC.TAT.1)
10. Delivery and Operation Documents (ALC_DEL.1)
11. Guidance Documents (AGD_OPE.1, AGD_PRE.1)
12. Vulnerabilities Documentation (AVA_VAN.3)
13. Flaw Remediation Documentation (ALC_FLR.2)

8. RATIONALE TABLES

Table 8-1 – Necessity of Security Objectives

Objective	Necessitated by:
O.Admin	A.Admin, T.Misuse_Management
O.Approved_Algorithms	P.Algorithms
O.Auth_Data_Protect	T.PIN_Compromise
O.Backup	T.Key_Discover, T.Malfunction
O.Check_Operation	T.Bad_FW_Load, T.Malfunction, T.Unauth_Function
O.Control_Access	T.Bad_FW_Load, T.Init_Data_Errors, T.Misuse_Management, T.Misuse_Sign
O.Data_Exchange_Protect	T.Exchange
O.Detect_Attack	T.Key_Discover
O.Import_Code	T.Bad_FW_Load
O.Key_Secure	T.Key_Discover, T.Misuse_Sign
O.Multi-Person_Control	T.Init_Data_Errors, T.Misuse_Management, T.Misuse_Sign
O.Secure_Init	T.Init_Data_Errors, T.Malfunction
O.Self_Protect	T.Malfunction, T.Unauth_Function
O.User_Authentication	T.Misuse_Management, T.Misuse_Sign
O.User_Data_Protect	T.Misuse_Sign
O.ENV_Application	A.Correct_Data, A.Human_Interface, A.User_Authentication
O.ENV_Audit	A.Audit_Support
O.ENV_Auth_Data	A.Admin, A.User_Authentication, A.User_Management
O.ENV_Backup	A.Backup_Data_Availability
O.ENV_Outage_Protection	T.Malfunction
O.ENV_Personnel	A.Admin, A.Audit_Support, A.User_Management
O.ENV_Protect_Access	A.Controlled_Access
O.ENV_Recovery	T.Init_Data_Errors, T.Malfunction, A.Backup_Data_Availability
O.ENV_Secure_Init	A.Admin, T.Init_Data_Errors
O.ENV_Signed_FW_Update	A.Legitimate_FW_Update

Table 8-2 – Mapping of Objectives to Threats

Threats	Objectives	Rationale
T.Bad_FW_Load	O.Check_Operation, O.Control_Access, O.Import_Code	This combination of objectives counters the threat by ensuring that the TOE will control the loading of firmware code, will load only valid firmware images and will check to verify that the integrity of the code is preserved prior to each activation of the code.
T.Exchange	O.Data_Exchange_Protect	This objective counters the threat by ensuring that the TOE has the capability to protect data exchanges from unauthorised disclosure and modification.
T.Init_Data_Errors	O.Control_Access, O.Multi-Person_Control, O.Secure_Init, O.ENV_Recovery, O.ENV_Secure_Init	This threat is countered by O.Control_Access with respect to the unauthorised use of services in the initialization phase. O.Secure_Init ensures that the TOE assumes its initial secure state immediately upon power-up, reset, or after other restart conditions. In addition, O.Multi-Person_Control provides the ability to enforce multi-person control to ensure that everything is correct before initialization is performed. Also, the objectives on the TOE environment O.ENV_Secure_Init and O.ENV_Recovery assist by ensuring that appropriate steps are taken by the organization to plan for a proper initialization or for recovery in the event of errors.
T.Key_Discover	O.Backup, O.Detect_Attack, O.Key_Secure	O.Key_Secure is responsible to ensure that no information about keys, such as private signing keys, is transmitted to any entity outside the TOE. O.Backup ensures that keys remain protected from disclosure during the backup and restoration processes. O.Detect_Attack permits the user to verify that no tampering has occurred that might allow an attacker to gain knowledge that could lead to discovering keys stored in the module.
T.Malfunction	O.Backup, O.Check_Operation, O.Secure_Init, O.Self_Protect, O.ENV_Outage_Protection, O.ENV_Recovery	This threat is countered by O.Check_Operation and the combination of O.Secure_Init and O.Self_Protect, which ensures that the TOE will start in a secure state and will protect itself from deliberate attempts to subvert its security enforcement by inducing faults. The TOE should also be protected as far as possible from defects caused by accidental mishandling and environmental failures (this is covered by the objective O.ENV_Outage_Protection). On the other hand, if a defect occurs, procedures within the TOE environment have to exist that allow the organisation operating the TOE to recover in a secure way from this defect. This is covered by the objective O.ENV_Recovery.

Threats	Objectives	Rationale
T.Misuse_Management	O.Admin, O.Control_Access, O.Multi-Person_Control, O.User_Authentication	This threat is countered by O.Control_Access, which restricts the use of TOE management functions to authorised users and O.Admin, which ensures that the users are competent to perform the management tasks. O.User_Authentication and O.Multi-Person_Control ensure that the user is properly identified and authenticated prior to invoking a management function.
T.Misuse_Sign	O.Control_Access, O.Key_Secure, O.Multi-Person_Control, O.User_Authentication, O.User_Data_Protect	O.Control_Access counters this threat for the user known to the TOE. O.Key_Secure ensures that the private key value is protected from disclosure, thereby preventing an attacker from using a rogue copy of the key to sign. O.User_Authentication and O.Multi-Person_Control prevents the misuse by persons not authorised to use the TOE. O.User_Data_Protect assists by allowing the user to verify the authenticity and the correctness of key material before it is used.
T.PIN_Compromise	O.Auth_Data_Protect	This objective counters the threat by ensuring that users' authentication data used for authentication cannot be easily guessed or captured via the host computer. Any attempt to impersonate a legitimate user will also be made significantly more difficult by imposing a limit on the number of authentication attempts before the user is locked or erased.
T.Unauth_Function	O.Check_Operation, O.Self_Protect	This combination of objectives counters the threat by ensuring that the TOE provides self-protection capability plus ensuring that error conditions do not leave the TOE is a non-secure state. This ensures that unauthorised functions cannot be executed either by subverting the TOE or by introducing malicious code through exploitation of errors such as buffer overflows.

Table 8-3 – Mapping of Objectives to Assumptions and Policies

Assumptions	Objectives	Rationale
A.Admin	O.Admin, O.ENV_Auth_Data, O.ENV_Personnel, O.ENV_Secure_Init	O.Admin ensures that the TOE can control access to administrative operations to allow only authorized personnel to perform these operations. O.ENV_Personnel satisfies the assumption by providing for competent administrators for the TOE and O.ENV_Auth_Data ensures that authentication data for the administrators is properly protected, thus ensuring that only the proper administrators have access to the administrative functions. O.ENV_Secure_Init requires that there are appropriate procedures in place to ensure secure initial setup, which is a prerequisite to ongoing secure administration.
A.Audit_Support	O.ENV_Audit, O.ENV_Personnel	This combination of objectives satisfies the assumption by ensuring that the environment provides adequate audit processing and review to support the secure operation of the TOE and that there are competent personnel to review and manage the audit data.
A.Backup_Data_Availability	O.ENV_Backup, O.ENV_Recovery	This combination of objectives satisfies the assumption by ensuring that the TOE environment protects the integrity and availability of data required for TOE initialisation, start-up, operation and recovery if stored or handled outside the TOE.
A.Controlled_Access	O.ENV_Protect_Access	This objective satisfies the assumption by ensuring that adequate physical security and procedural measures are in place to control access to the facility hosting the TOE. This protects against attacks requiring direct physical access to the TOE or host system and leakage of information via monitoring the power consumption or via radiation.
A.Correct_Data	O.ENV_Application	O_ENV_Application ensures that the applications that use the TOE will perform the required checks on the data they pass to the TOE.
A.Human_Interface	O.ENV_Application	The host application will provide the human interface for interaction with the TOE and ensures the confidentiality and integrity of data passed to the TOE.
A.Legitimate_FW_Update	O.ENV_Signed_FW_Update	This objective satisfies the assumption by ensuring that a process is in place within the environment to digitally sign firmware update packages to prove their legitimacy.
A.User_Authentication	O.ENV_Application, O.ENV_Auth_Data	This combination of objectives satisfies the assumption by ensuring that the environment provides an application that interacts correctly with human users for the purpose of authentication, protects the authentication data provided by the users and represents those users correctly to the TOE. Also, that the human users exercise proper control over their authentication data.

Assumptions	Objectives	Rationale
A.User_Management	O.ENV_Auth_Data, O.ENV_Personnel	This combination of objectives satisfies the assumption by ensuring that the management policies and procedures governing the assignment of individual human users to roles on the TOE are properly followed and that the users protect their authentication data so as to support their role assignments.
P.Algorithms	O.Approved_Algorithms	This objective satisfies the assumption by ensuring that the TOE provides the algorithms required by organizational policy.

Table 8-4 – Necessity of Security Functional Requirements

SFR	Necessitated by:
FCS_CKM.1	O.Approved_Algorithms, O.Key_Secure
FCS_CKM.2 (Backup)	O.Approved_Algorithms, O.Key_Secure, O.Backup
FCS_CKM.2 (FW Update)	O.Approved_Algorithms, O.Import_Code
FCS_CKM.3	O.Key_Secure
FCS_CKM.4	O.Approved_Algorithms, O.Key_Secure
FCS_COP.1 (SIGN)	O.Approved_Algorithms, O.Data_Exchange_Protect, O.Key_Secure
FCS_COP.1 (DIGEST)	O.Approved_Algorithms, O.Data_Exchange_Protect
FCS_COP.1 (RSA ENC/DEC)	O.Approved_Algorithms, O.Data_Exchange_Protect
FCS_COP.1 (TDES ENC/DEC)	O.Approved_Algorithms, O.Data_Exchange_Protect, O.Import_Code
FCS_COP.1 (AES ENC/DEC)	O.Approved_Algorithms, O.Data_Exchange_Protect
FDP_ACC.1 (TAC)	O.Control_Access, O.Data_Exchange_Protect, O.Key_Secure
FDP_ACF.1 (TAC)	O.Control_Access, O.Data_Exchange_Protect, O.Key_Secure
FDP_BKP.1	O.Backup, O.Key_Secure
FDP_DAU.1	O.User_Data_Protect
FDP_DAU.2	O.User_Data_Protect, O.Key_Secure
FDP_ETC.1	O.Data_Exchange_Protect
FDP_ITC.1	O.Data_Exchange_Protect
FDP_RIP.1	O.User_Data_Protect, O.Key_Secure
FDP_RIP.2	O.User_Data_Protect, O.Key_Secure
FIA_AFL.1 (SO)	O.User_Authentication, O.Auth_Data_Protect
FIA_AFL.1 (User)	O.User_Authentication, O.Auth_Data_Protect
FIA_ATD.1	O.User_Authentication, O.Auth_Data_Protect
FIA_SOS.1	O.User_Authentication, O.Auth_Data_Protect
FIA_SOS.2	O.User_Authentication, O.Auth_Data_Protect
FIA_UAU.1	O.User_Authentication
FIA_UAU.4	O.Auth_Data_Protect, O.User_Authentication
FIA_UAU.5	O.User_Authentication, O.Multi-Person_Control
FIA_UID.1	O.User_Authentication
FIA_USB.1	O.User_Authentication
FMT_MOF.1	O.Admin
FMT_MSA.1 (Object Attributes)	O.Control_Access
FMT_MSA.2 (Object Attributes)	O.Control_Access
FMT_MSA.3 (Object Attributes)	O.Control_Access
FMT_MTD.1 (Login Failures)	O.Admin, O.Control_Access, O.User_Authentication
FMT_MTD.1 (UAV – User Locked Flag)	O.Admin, O.Control_Access, O.User_Authentication
FMT_MTD.1 (UAV – Challenge Secret – Crypto Officer)	O.Admin, O.Control_Access, O.User_Authentication
FMT_MTD.1 (SOV)	O.Admin, O.Control_Access, O.User_Authentication
FMT_SMF.1	O.Admin, O.Control_Access, O.User_Authentication
FMT_SMR.1	O.Admin, O.Control_Access
FPT_FLS.1	O.Secure_Init
FPT_ITC.1	O.Backup
FPT_ITI.1	O.Backup
FPT_PHP.1	O.Detect_Attack
FPT_RCV.1	O.User_Data_Protect, O.Secure_Init
FPT_TST.1	O.Check_Operation, O.Control_Access, O.Secure_Init, O.Self_Protect
FRU_FLT.1	O.User_Data_Protect
FTP_TRP.1	O.Auth_Data_Protect, O.User_Authentication
FTP_ITC.1 (FW Update)	O.Import_Code
FTP_ITC.1 (Key Cloning)	O.Backup

Table 8-5 – Mapping of Security Functional Requirements to Objectives

Objectives	Security Functional Requirements	Rationale
O.Admin	FMT_MOF.1, FMT_MTD.1(Login Failures), FMT_MTD.1(UAV – User Locked Flag), FMT_MTD.1(UAV – Challenge Secret – Crypto Officer), FMT_MTD.1(SOV), FMT_SMF.1, FMT_SMR.1	This combination of SFRs satisfies the objective by requiring that the TOE provides suitable roles and management functions to administer the TOE.
O.Approved_Algorithms	FCS_CKM.1, FCS_CKM.2 (Backup), FCS_CKM.2 (FW Update), FCS_CKM.4, FCS_COP.1 (RSA ENC/DEC), FCS_COP.1 (TDES ENC/DEC), FCS_COP.1 (AES ENC/DEC), FCS_COP.1 (SIGN), FCS_COP.1 (DIGEST)	This combination of SFRs satisfies the objective by requiring that the TOE provide approved algorithms and key management techniques.
O.Auth_Data_Protect	FIA_UAU.4, FIA_AFL.1 (SO), FIA_AFL.1 (User), FIA_ATD.1, FIA_SOS.1, FIA_SOS.2, FTP_TRP.1	This combination of SFRs satisfies the objective by requiring that the TOE: • prevent reuse/replay of authentication data (FIA_UAU.4), • prevent brute force attacks on authentication data (FIA_AFL.1) • ensure that authentication data meets the required minimum strength requirements (FIA_SOS.1) • enforce the use of TSF-generated authentication data (FIA_SOS.2), and • utilise a trusted path to protect authentication data from eavesdropping (FTP_TRP.1).
O.Check_Operation	FPT_TST.1	This security objective is implemented in the TOE by the SFR for TSF testing FPT_TST.1. If these tests detect an error the TOE will transit into a secure state and prevent the normal operation. The FPT_TST.1 includes checks of the executable code.
O.Backup	FCS_CKM.2 (Backup), FDP_ BKP.1, FPT_ITC.1, FPT_ITI.1, FTP_ITC.1 (Key Cloning)	The TOE backup and restore functions requires the SFR FDP_BKP.1 the confidentiality and integrity protection of backup data. The confidentiality and integrity protection of the TSF data as part of the backup data is implemented by the SFR FPT_ITC.1 and SFR FPT_ITI.1 The FDP_BKP.1 needs FCS_CKM.2 to securely derive the session key used to encrypt the transmission of keys from the primary to the backup module. For the TOE, FTP_ITC.1 (Key Cloning) ensures that backup data is protected when transmitted from the TOE to a backup token and vice versa.

Objectives	Security Functional Requirements	Rationale
O.Control_Access	FDP_ACC.1 (TAC), FDP_ACF.1 (TAC), FMT_MSA.1 (Object Attributes), FMT_MSA.2 (Object Attributes), FMT_MSA.3 (Object Attributes), FMT_MTD.1 (Login Failures), FMT_MTD.1 (UAV – User Locked Flag), FMT_MTD.1 (UAV – Challenge Secret – Crypto Officer), FMT_MTD.1 (SOV), FMT_SMF.1, FMT_SMR.1	Access control is implemented in the TOE - FDP_ACC.1 (TAC), FDP_ACF.1 (TAC), with the roles Security Officer, Crypto-officer and Crypto-user as defined by the SFR FMT_SMR.1. The SFRs FMT_MSA.2, FMT_MSA.3, and FMT_SMF.1 assign the management functions for the cryptographic module to the Security Officer and key management functions to the Crypto Officer. FMT_MSA.1 (Object Attributes), FMT_MSA.2 (Object Attributes), FMT_MSA.3 (Object Attributes) require controls over the management of object attributes needed to support access control. The user identification and authentication needed to support enforcement of the access control policy is provided by the SFRs satisfying O.User_Authentication and O.Multi-Person_Control. The iterations of FMT_MTD.1 ensure the proper management and control of TSF data needed to support user identification and authentication.
O.Data_Exchange_Protect	FCS_COP.1 (RSA ENC/DEC), FCS_COP.1 (TDES ENC/DEC), FCS_COP.1 (AES ENC/DEC), FCS_COP.1 (DIGEST), FCS_COP.1 (SIGN), FDP_ACC.1 (TAC), FDP_ACF.1 (TAC), FDP_ETC.1, FDP_ITC.1	This combination of SFRs satisfies the objective by requiring that the TOE provide controls over export and import of user data (FDP_ACC.1 (TAC), FDP_ACF.1 (TAC), FDP_ETC.1, FDP_ITC.1 plus the cryptographic functions and approved algorithms needed to protect data being exchanged (FCS_COP.1 (RSA Enc/Dec), FCS_COP.1 (TDES ENC/DEC), FCS_COP.1 (AES ENC/DEC), FCS_COP.1 (SIGN), FCS_COP.1 (DIGEST)).
O.Detect_Attack	FPT_PHP.1	The SFR FPT_PHP.1 ensures that any attempts at physical tampering will leave readily verifiable evidence of the attack. The use of this SFR is considered reasonable because the TOE will normally be used within a secure area and the TOE does not store keys and other sensitive data in plaintext.
O.Import_Code	FTP_ITC.1 (FW Update), FCS_CKM.2 (FW Update), FCS_COP.1 (TDES ENC/DEC)	This combination of SFRs satisfies the objective by requiring that the TOE provide the mechanisms and approved algorithms needed to receive a firmware update package from the vendor in a trusted manner and for the TOE to verify the authenticity of the firmware update.

Objectives	Security Functional Requirements	Rationale
O.Key_Secure	FCS_CKM.1, FCS_CKM.2 (Backup), FCS_CKM.3, FCS_CKM.4, FCS_COP.1 (SIGN), FDP_ACC.1 (TAC), FDP_ACF.1 (TAC), FDP_BKP.1, FDP_DAU.2, FDP_RIP.1, FDP_RIP.2	The SFRs ensure the cryptographically secure key and key pair generation by FCS_CKM.1 as well as operation by FCS_COP.1(SIGN) according to the list of approved algorithms and parameters. FCS_CKM.3 ensures the security of keys in storage and when accessed by a user. The confidentiality and integrity of the keys will be protected by SFR FDP_RIP.1 and FDP_RIP.2 during internal processing. The SFR FCS_CKM.4 requires secure key destruction to prevent any misuse of keys after their operational life time. FDP_BKP.1 and FCS_CKM.2 (Backup) ensure that keys remain secure when they are backed up. The overall key management and operation is under access control of the SFR FDP_ACC.1 (TAC) and FDP_ACF.1 (TAC).
O.Multi-Person_Control	FIA_UAU.5	This SFR satisfies the objective by requiring that the TOE provide a mechanism for multi-person control over access to the TOE's functions.
O.Secure_Init	FPT_FLS.1, FPT_RCV.1, FPT_TST.1	This combination of SFRs satisfies the objective by requiring that the TOE ensure that it is in its initial secure state immediately upon power-up, reset, or after other restart conditions.
O.Self_Protect	FPT_TST.1, ADV_ARC	This combination of SFR and SAR satisfies the objective by requiring that the TOE protect its own functions by requiring that enforcement functions are invoked and succeed before allowing operations to proceed and maintaining a separate execution space for its functions.
O.User_Authentication	FIA_ATD.1, FIA_AFL.1 (SO), FIA_AFL (User), FIA_UID.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.5, FIA_SOS.1, FIA_SOS.2, FIA_USB.1, FMT_MTD.1 (Login_Failures), FMT_MTD.1 (UAV – User Locked Flag), FMT_MTD.1 (UAV – Challenge Secret – Crypto Officer),, FMT_MTD.1 (SOV), FMT_SMF.1, FTP_TRP.1	This combination of SFRs satisfies the objective because the requirements, FIA_ATD.1, FIA_UID.1, FIA_UAU.4 FIA_SOS.1 FIA_SOS.2, FIA_USB.1, provide identification/authentication mechanisms, using randomly generated secrets of specified minimum lengths, and that users are bound to subjects. FIA_AFL.1 (SO), FIA_AFL (User) protect against password guessing attacks and FTP_TRP.1 protects against snooping attacks. The SFRs , FMT_MTD.1 (Login_Failures), FMT_MTD.1 (UAV – User Locked Flag), FMT_MTD.1 (UAV – Challenge Secret – Crypto Officer),, FMT_MTD.1 (SOV), FMT_SMF.1 provide management functions for identification and authentication.

Objectives	Security Functional Requirements	Rationale
O.User_Data_Protect	FDP_DAU.1, FDP_DAU.2, FDP_RIP.1, FDP_RIP.2, FPT_RCV.1, FRU_FLT.1	This combination of SFRs satisfies the objective by requiring that the TOE provide mechanisms to protect the confidentiality and integrity of user objects within the TOE and provide the means for the user to verify the integrity of the user object data.

Table 8-6: Dependency Rationale for Security Functional Requirements

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 FMT_MSA.2 Secure security attributes	Met by inclusion of FCS_COP.1, FCS_CKM.4 and FMT_MSA.2 as SFRs
FCS_CKM.2	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	Met by inclusion of FDP_ITC.1, FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2 as SFRs.
FCS_CKM.3	[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 FMT_MSA.2 Secure security attributes	Met by inclusion of FDP_ITC.1 and FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2 as SFRs
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] FMT_MSA.2 Secure security attributes	Met by inclusion of FDP_ITC.1 and FCS_CKM.1 and FMT_MSA.2 as SFRs
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	Met by inclusion of FDP_ITC.1 and FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2 as SFRs
FDP_ACC.1	FDP_ACF.1 Security attribute based access control	Met by inclusion of FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	Met by inclusion of FDP_ACC.1 and FMT_MSA.3
FDP_BKP.1	[FCS_CKM.1 Cryptographic key generation or FCS_CKM.2 Cryptographic key distribution or FDP_ITC.1 Import of user data without security attributes] FCS_COP.1 Cryptographic operation	Met by inclusion of FCS_CKM.1, FCS_CKM.2 (BACKUP), FCS_COP.1(TDES ENC/DEC)
FDP_DAU.1	No dependencies	No dependencies
FDP_DAU.2	FIA_UID.1	Met by inclusion of FIA_UID.1 as SFR

Security Functional Requirement	Dependencies	Rationale
FDP_ETC.1	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	Met by inclusion of FDP_ACC.1
FDP_ITC.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialization	Met by inclusion of FDP_ACC.1 and FMT_MSA.3
FDP_RIP.1	No dependencies	No dependencies
FDP_RIP.2	No dependencies	No dependencies
FIA_AFL.1	FIA_UAU.1 Timing of authentication	Met by inclusion of FIA_UAU.1
FIA_ATD.1	No dependencies	No dependencies
FIA_SOS.1	No dependencies	No dependencies
FIA_SOS.2	No dependencies	No dependencies
FIA_UAU.1	FIA_UID.1 Timing of identification	Met by inclusion of FIA_UID.1
FIA_UAU.4	No dependencies	No dependencies
FIA_UAU.5	No dependencies	No dependencies
FIA_UID.1	No dependencies	No dependencies
FIA_USB.1	FIA_ATD.1 User attribute definition	Met by inclusion of FIA_ATD.1
FMT_MOF.1	FMT_SMF.1 Specification of management functions, FMT_SMR.1 Security roles	Met by inclusion of FMT_SMF.1 and FMT_SMR.1
FMT_MSA.1	[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	Met by inclusion of FDP_ACC.1 and 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	Met by inclusion of FDP_ACC.1, FMT_MSA.1, FMT_SMR.1
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Met by inclusion of FMT_MSA.1 and FMT_SMR.1
FMT_MTD.1	FMT_SMF.1 Specification of Management Functions FMT_SMR.1 Security roles	Met by inclusion of FMT_SMR.1
FMT_SMF.1	No dependencies	No dependencies
FMT_SMR.1	FIA_UID.1 Timing of identification	Met by inclusion of FIA_UID.1
FPT_FLS.1	No dependencies	No dependencies
FPT_ITC.1	No dependencies	No dependencies
FPT_ITI.1	No dependencies	No dependencies
FPT_PHP.1	No dependencies	No dependencies
FPT_RCV.1	AGD_OPE.1 Operational user guidance	Met by provision of TOE User/Administrative Guidance
FPT_TST.1	No dependencies	No dependencies
FRU_FLT.1	FPT_FLS.1 Failure with preservation of secure state	Met by inclusion of FPT_FLS.1.
FTP_ITC.1	No dependencies	No dependencies
FTP_TRP.1	No dependencies	No dependencies

Table 8-7 – Assurance Measures

Assurance Measures	Document Title	Document Description
Configuration Management	Configuration Management Manual for Luna® Product	This document describes the CM procedure for Luna® product design, development, testing, release and manufacture and maintenance.
	Configuration Management Plan for Luna® CA4 2.6 with Firmware 4.8.7	This document defines CM procedures to be used during development of the Luna® CA4.
	SafeNet Engineering Change Request Procedure	This procedure establishes the requirements for requesting action that may initiate a release or change to released hardware, software, or controlled documentation during the various product life cycles.
	SafeNet Engineering Change Procedure	This procedure establishes the requirements for documenting and approving Engineering Change Notices (ECNs) to release new items as well as for released hardware, software, or controlled documentation during the various product life cycles.
	Luna® Product Identification and Traceability	This document describes product identification and marking on all Luna® products and specifies the process of maintaining product information for traceability purposes.
	Luna® Critical Security Material Handling for Contract Manufacturers	This document describes the various administrative procedures and security policies for handling critical key material that is used during the manufacture of Luna® devices.
	Luna® Product Process Flow for Contract Manufacturers	The checklist is intended to show the process flow of Luna® finished goods through the Contract Manufacturer's facility.
	Luna® Board Level Development Process	This document describes the design procedure for board level development of Luna® products. This document does not cover mechanical or system level design procedures.
	Luna® Products Life Cycle – Assuring Integrity	This document provides an overview of the procedures and steps taken to assure the integrity of Luna® products from development through to delivery.
	Luna® PCM/CA4 2.6 SOW	This document defines the product requirements for the release of Luna® PCM and Luna® CA4 2.6.
Delivery and operation documentation	Luna® CA4 Quick Start Guide / Content Sheet PED-Auth	Quick Start Guide for Luna® CA4 Release 2.6. Describes the procedures for secure initialization of the product. The Content Sheet identifies the TOE components that the customer should expect to find in the delivered product.
	Luna® Critical Security Material Handling for Contract Manufacturers	This document describes the various administrative procedures and security policies for handling critical key material that is used during the manufacture of Luna® devices.
	Luna® Product Process Flow for Contract Manufacturers	The checklist is intended to show the process flow of Luna® finished goods through the Contract Manufacturer's facility.

Assurance Measures	Document Title	Document Description
Development Documentation	Security Architecture Description	Presents an architectural description of the Luna® CA4 with particular emphasis on the security architecture elements that ensure the non-bypassability, self-protection and isolation of the security-enforcing components.
	Luna® Functional Specification	Presents a high-level overview of the Luna® CA4 and describes its functions in detail, including its cryptographic capabilities and security features with a full definition of all relevant external interfaces.
	PKCS #11: Cryptographic Token Interface Standard, V 2.20	This document is offered by RSA Laboratories to developers of computer systems employing public-key and related technology.
	Extensions to PKCS#11, Cryptographic Token Interface Standard	Describes a set of extensions to the standard application programming interface (API), called Cryptoki. Specifies the data types and functions available to an application requiring cryptographic services using the ANSI C programming language.
	Luna® High-Level Design	Provides an overview of the cryptographic module's firmware architecture and a foundation for further design documents that address each module in more detail.
	Luna® Interface Control Document (ICD)	Defines the command set and associated parameters that are used to communicate to the Luna® cryptographic module.
	Luna® Memory Management Subsystem Design	Identifies the set of requirements met by the Luna® Memory Management Subsystem Design and provides detailed notes on its implementation.
	Luna® Session Manager Subsystem Design	Presents the Luna® Session Manager Subsystem capabilities and describes the implementation, internal interface and external command processing details.
	Luna® Object Management Subsystem Design	Identifies the set of requirements to be met by the Luna® Object Management Subsystem Design and provides detailed notes on its design and implementation.
	Luna® Boot Block Subsystem Design	Identifies the set of requirements met by the Luna® boot block and provides detailed notes on its implementation.
	Luna® Communication Subsystem Design	Identifies the set of requirements to be met by the Luna® Communication Subsystem design and provides detailed notes on its implementation.
	Luna® Param Subsystem Design	Describes the Luna® Param Subsystem Design implementation and defines the internal interface.
	Luna® Cryptographic Module Self Tests	Identifies the set of requirements to be met by the Luna® CA4 self tests and provides detailed notes on their implementation.
	Luna® CA4 User Subsystem Design	Design and validation document for Luna® firmware user subsystem design.

Assurance Measures	Document Title	Document Description
	Luna® Cryptographic Algorithms Subsystem Design	Describes the generation, distribution, loading, storing, use, updating and destruction of the cryptographic material – keys and vectors – necessary for all of the cryptographic operations performed by the cryptographic module.
	Luna® Key Cloning Protocol	Describes the key cloning protocol.
	Luna® Key Management Subsystem Design	Describes the generation, distribution, loading, storing, use, updating and destruction of the cryptographic material – keys and vectors – necessary for all of the cryptographic operations performed by the cryptographic module.
	Luna® M of N Activation Protocol	Describes the M of N activation capability implemented on the Luna® cryptographic modules.
	Luna® Main Subsystem Design	This document presents the Luna® Main Subsystem low-level design in terms of subordinate modules, their roles within the subsystem and their interactions to provide the Main Subsystem services.
	Luna® Random Number Generation (RNG) process	This document describes the Random Number Generation (RNG) process used to generate the random bits required by the cryptographic functions running on the Luna® family of cryptographic hardware modules hereinafter referred to as Luna® modules.
	Luna® Firmware (F/W) Update High Level Design	Defines the process used to perform a secure update to the firmware on cryptographic modules in the field. The firmware update process maintains all user information that exists on the cryptographic module prior to the update.
	Luna® Serial Communication Protocol for Luna® CA4	This document describes the Luna® Serial Communication Protocol (SCP) used to transfer data over the serial communication port interface defined for the Luna® CA4 cryptographic module
	Luna® Rule Subsystem Design	This document describes the Luna® Cryptographic Module Rule Subsystem implementation and defines the internal interface.
	Physical Design	This document describes the effective physical security mechanisms for the Luna® G4 cryptographic module, a PCMCIA form factor cryptographic asymmetric accelerator, also known as Luna® PCM or Luna® CA4.
	Luna® CA4 Key Card HW Assembly Drawings	
	Top Assembly, Luna® CA4 Drawings	
Implementation of the TSF		Firmware code, hardware schematics, FPGA code to be provided in accordance with evaluator’s request.

Assurance Measures	Document Title	Document Description
Informal correspondence demonstration		Correspondence mappings for TSS to Functional Specification, Functional Specification to HLD and HLD to LLD are provided as part of the appropriate documents.
Informal TOE security policy model	Luna® CA4 Security Policy	The Security Policy describes the security behaviour of the Luna® CA4.
Guidance documents	Guidance documents provided with the TOE are primarily intended as Administrator guidance. The administration functions are normally carried out by the Security Officer, or possibly a designated User, using the CLI software as the interface. In most cases, these functions will be performed very infrequently. User guidance documents are not provided because the normal user of the TOE is an application program making function calls to the TOE via the PKCS #11 Cryptographic API. Direct access to the TOE's functions by a human user only occurs in the course of performing administration functions.	
	Luna® CA4 2.6 Quick Start Guide / Content Sheet Trusted Path Authentication	Quick Start Guide for Luna® CA4 Release 2.6.
	Luna® CA4 2.6 Online Help	The Online Help system provides the detailed Administrator/User guidance for the operation of the product.
	Luna® CA4 2.6 Release Notes	The Release Notes identify major issues from previous releases that have been fixed in the current release and any outstanding known issues.
Life cycle support	Luna® Software Development Process	This document defines the high-level software development process used for Luna® product.
	Luna® Product Identification and Traceability	This document describes product identification and marking on all Luna® products and specifies the process of maintaining product information for traceability purpose
	Luna® Board Level Development Process	This document describes the design procedure for board level development of Luna® product. This document does not cover mechanical or system level design procedures.
	Return Material Authorization Process	RMA Process for Sales, Service and Logistics
	Luna® Life Cycle – Assuring Integrity	This document provides an overview of the procedures and steps taken to assure the integrity of Luna® products from development through to delivery.
	Luna® Development Tools	This document identifies and explains the use of the various tools used within Luna® development environment.
	Luna® Critical Security Material Handling for Contract Manufacturers	This document describes the various administrative procedures and security policies for handling critical key material that is used during the manufacture of Luna® devices.
	Luna® Product Process Flow for Contract Manufacturers	The checklist is intended to show the process flow of Luna® finished goods through the Contract Manufacturer's facility.

Assurance Measures	Document Title	Document Description
	SafeNet Canada Security Policies	This document presents the security policies to be followed by all SafeNet Canada full-time, part-time and contract employees.
	Luna® Problem Reporting Process	This document discusses the problem reporting process employed for Software, Firmware and Hardware related problems.
Developer Tests	ScriptHelp.txt	Help text describing the use of the Scripter tool.
	List of Scripts Used in Engineering Testing and run by Scripter	List of Scripts Used in PV Testing and run by Scripter. Demonstrates tests coverage.
	Cloning Scripts	Script used by Engineering Test to test the cloning functionality. Demonstrates tests coverage
	Test Plan for Luna® CA4 Release 2.6	This document defines the preparation, testing strategy and exit criteria for the testing of Luna® CA4 with software 2.6.
	Test Cases for Luna® CA4 Release 2.6	This document will give detailed information on Luna® CA4 2.6 installation, cryptoki toolkit and cryptographic module testing for Luna® CA4.
	Release Notice and Test Report for Luna® CA4 Release 2.6	This document provides the test results and release authorization for Luna® CA4 2.6.
	Testing Coverage and Depth Analysis	This document has been prepared for the Common Criteria evaluation of the Luna® CA4 (K5) and describes the functional coverage of the test plan and the depth of testing related to the high-level design interfaces.
	Independent testing - sample	To be performed by evaluator.
Vulnerability assessment	Vulnerability Analysis	This document presents the developer's analysis of the vulnerabilities to which the Luna® CA4 may be subject.

APPENDIX A – REFERENCES

Reference Number	Document Number	Revision	Author	Title
[1]	ISO/IEC 15408-1	V3.1		Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and General Model
[2]	ISO/IEC 15408-2	V3.1		Common Criteria for Information Technology Security Evaluation – Part 2: Security Functional Requirements
[3]	ISO/IEC 15408-3	V3.1		Common Criteria for Information Technology Security Evaluation – Part 3: Security Assurance Requirements
[4]	FIPS 140-2	December 2002	National Institute of Standards and Technology	Security Requirements for Cryptographic Modules
[5]		Version 2.20, June 2004	RSA Laboratories	PKCS #11: Cryptographic Token Interface Standard
[6]		Version 2.1, June 2002	RSA Laboratories	PKCS #1: RSA Cryptography Standard
[7]	CR-2952	1	SafeNet, Inc.	Overview of Documentation Required to Support Luna® CA4 Common Criteria Evaluation
[8]	CR-2877	3	SafeNet, Inc.	This document defines the logical interface specification for the Luna® CA4 Cryptographic Module.