ST3 Ace Security Target

Common Criteria: EAL2

Version 1.1

21-NOV-13

Document management

Document identification

Document ID	ACE_EAL2_ST
Document title	ST3 Ace Security Target
Document date/version	1.1, 21-NOV-13

Document history

Version	Date	Description
0.1	20-MAY-13	Released for internal review
0.2	15-JUL-13	Updated to address EOR-ASE v1.0
0.3	1-AUG-13	Released to evaluator
0.4	27-SEP-13	Updated to address EOR-ASE v2.0
0.5	1-OCT-13	Released to evaluator and MyCB
0.6	5-OCT-13	Added notes in Section 5.2.2, 5.2.3 and 5.2.6
1.0	17-OCT-13	Final Release
1.1	21-NOV-13	Updated to address comments from MyCB.

Table of Contents

1 Security Target introduction (ASE_INT.1)

1.1 ST reference

ST Title	ST3 Ace Security Target
ST Identifier	ACE_EAL2_ST
ST Version/Date	1.1 /21-NOV-13

1.2 TOE reference

TOE Title	ST3 Ace
TOE Version	The TOE consists of the following components: • ST3 Ace Token Manager (v1.0.13.927) • ST3 Ace Middleware (v1.0.13.910), and • SecureCOS Firmware (v5.2)

1.3 Document organization

This document is organized into the following major sections:

- Section 1 provides the introductory material for the ST as well as the TOE description (ASE_INT.1).
- Section 2 provides the conformance claims for the evaluation (ASE_CCL.1).
- Section 3 provides the definition of the security problem that the TOE has been designed to address (ASE_SPD.1).
- Section 4 defines the security objectives for the TOE and the environment (ASE_OBJ.2).
- Section 5 contains the security functional and assurance requirements derived from the Common Criteria, Part 2 and 3 respectively, which are in turn satisfied by the TOE and the development lifecycle (ASE_REQ.2).
- Section 6 provides a summary of the TOE specification, identifying the IT security functions provided by the TOE (ASE_TSS.1).

1.4 Defined terms

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

Term	Description
Authentication	It is information used to verify the claimed identity of a user.
Cipher-block chaining (CBC)	In the cipher-block chaining (CBC) mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted.
Electronic codebook (ECB)	The message is divided into blocks and each block is encrypted separately.
FIPS 46-3	FIPS-46-3 is the National Institute of Standards and Technology (NIST) technical publication defining Data Encryption standards.
FIPS 180-2	FIPS-180-2 is the National Institute of Standards and Technology (NIST) technical publication defining Secure Hash standards.
Integrated circuit (IC)	Electronic component(s) designed to perform processing and/or memory functions.
MD5	MD5 (Message-Digest algorithm 5) is a cryptographic hash function with a 128-bit hash value.
PIN	A numeric password shared between a user and a system that can be used for user authentication.
PKCS#11	The first in the Public-Key Cryptography Standards (PKCS) standards library published by RSA Laboratories. This document provides the basic definitions of, and recommendations for, implementing the RSA algorithm for public-key cryptography.
RSA	An algorithm for public-key cryptography published by RSA Laboratories
Smart Card	A credit card-sized chip card with embedded integrated circuits. Often used to store keys for authentication.
Security Officer (SO)	A user authorized to perform TOE configuration or other TOE Security Officer functions.

Term	Description
Secure Code	Unique code is sent through to the user's registered mobile number that will be used to decrypt the SO PIN for the user to unblock the token.
SHA-1	A cryptographic hash function designed by the National Security Agency (NSA) and published by NIST as a U.S. Federal Information Processing Standard. SHA stands for Secure Hash Algorithm. The function produces a 160-bit hash value.
TDES	A block cipher which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block.
TSF data	Data created by and for the TOE, that might affect the operation of the TOE
TMS RA	Token Management System Registration Authority
User	It means any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
User data	Data created by and for the user, that does not affect the operation of the TSF

1.5 TOE overview

1.5.1 TOE usage and major security functions

The Target of Evaluation (TOE) is the ST3 Ace. The TOE provides secure storage to store digital certificate(s) and cryptographic keys. The ST3 Ace follows the PKCS#11 standard and implements authentication via PIN to prevent unauthorized access to the token.

The TOE is comprised of the three following core components of the ST3 Ace product:

- **SecureCOS Operating System**: The operating system (firmware) embedded in a microprocessor smart chip based USB token. The firmware provides the core cryptographic functionality of the TOE.

- **ST3 Ace Middleware**: Two compiled binaries that utilise exported APIs to provide an interface to the core cryptographic security functionality of the TOE, providing developers with an easily accessible method for engaging PKI-related functionality to support the development of enterprise authentication and integrity solutions.

- **ST3 Ace Token Manager:** The TOE provides an application for the user to manage the cryptographic key security of the TOE.

The above TOE components provide the following functionalities:

- **Data encryption and decryption.** Perform digital data encryption and decryption (such as email encryption and decryption);
- **Digital signing**. Perform digital signing on documents (such as PDFs and word documents) and emails;
- **Secure Storage**: Provide storage to store certificate(s) and cryptographic keys. Private data cannot be exported from the TOE. Users must enter their token PIN to access private data or cryptographic keys, or to perform signing or decryption;
- **Cryptographic key generation:** Generate cryptography keys, such as an RSA key pair, that can be used to perform signing, encryption and decryption; and
- **Token management:** Provide a token manager tool to manage ST3 Tokens and perform management functions, such as changes in Token name, changes in token PIN, data deletion and data object management.

The following table highlights the range of security functions and features implemented by the TOE.

Security function	Description
Cryptographic Operations	The TOE provides a cryptographic library for cryptographic operations that can be used by third party applications outside the TOE, such as the encryption and decryption of email. The TOE also provides the functionality to digitally sign documents and files.
User Authentication	Access to the TOE management functions requires users to authenticate using their PIN. After 6 failed authentication attempts, the token will be locked and a SO PIN is needed to unblock the token and reset the user PIN. The integration with the Token Management System Registration Authority (TMS RA) will allow a user to unblock the TOE and reset the User PIN, without the need to deliver the physical token to the token management team. This function must first be enabled via the TMS RA by the token manager. Users must request a Secure Code that will be sent to the users registered smartphone via SMS. On the Token Manager, users must enter the Secure Code and submit a request to the TMS RA in order to unblock or reset User PIN. **Note:** The TMS RA is outside the scope of this evaluation
Security Management	The TOE provides management functions such as such as token management (name change, PIN change, unblock token) and object management (view object, export/import object).

1.5.2 TOE Type

The TOE is categorised within the ***PKI-related security solutions*** category as identified on the Common Criteria Portal for all Certified Products.

1.5.3 Supporting Hardware, software and/or firmware

The underlying hardware and software that is used to support the TOE are:

Minimum Requirements	
Host Computer	
Processor	x86/x64 architecture
Supported Operating Systems	• Windows Server 2000; • Windows XP SP3;

	• Windows 2003; • Windows Vista; • Windows 7; and • Windows 8
Third Party Application (Hot-pluggable)	• Internet Explorer 8; • Microsoft Office 2003/2007/2010; • Microsoft Outlook 2003/2007/2010 ; • Mozilla Firefox v10; • Mozilla Thunderbird v10; and • Adobe Acrobat 8.
USB Version	USB 1.0, USB 2.0 and USB 3.0
USB Token (ST3 Ace)	
Card Operating System	ST3 Ace hardware token with pre-installed SecureCOS
Memory Size	2MB on Flash for Middleware & CSP

1.6 TOE description

1.6.1 Physical scope of the TOE

The TOE is comprised of the ST3 Ace operating system **SecureCOS**, the **ST3 Ace middleware** and the **ST3 Ace Token Manager**. SecureCOS is embedded into the memory of the secure microprocessor smart chip-based USB token. The middleware and the Token Manager are installed onto a host computer for third party applications to communicate with SecureCOS and to manage the key security functionality of the TOE. The TOE components are shown in figure 2 below.

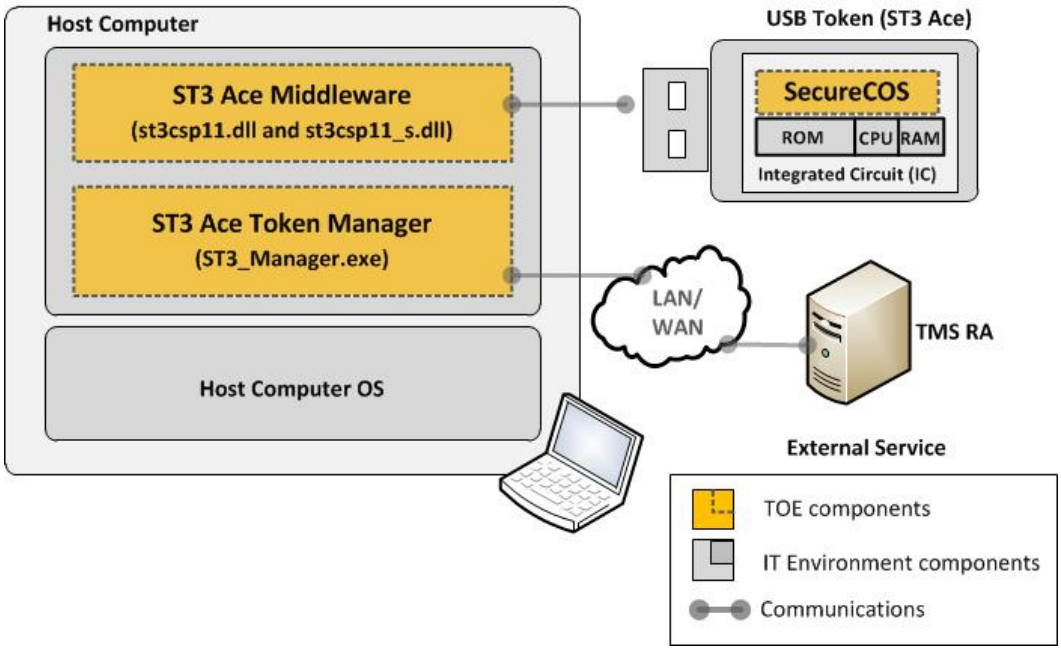

Figure 2: TOE Components

The table below shows the versions and descriptions of the TOE components.

TOE Component	Description
SecureCOS	The operating system (firmware) embedded on the token IC. The operating system provides the core cryptographic functionality of the TOE. Note: SecureCOS is embedded into the memory of the secure microprocessor smart chip-based USB token.
ST3 Ace Middleware (st3csp11.dll and st3csp11_s.dll)	Compiled binaries that provide an interface to the core cryptographic security functionality of the TOE - therefore providing developers with an easily accessible method for engaging PKI-related functionality to support the development of enterprise authentication and integrity solutions.
ST3 Ace Token Manager (ST3_Manager.exe)	The TOE provides an application for the user to manage the key security functionality of the TOE and perform functions such as changing User PINs and Certificate Management (e.g. view, import, export, unblock token, deletion, renewal of certificate)

1.6.2 Logical scope of the TOE

The logical boundary of the TOE is summarized below.

- **User Authentication:** Users and the security officer of the token authenticate themselves to the TOE using their PINs. The user will only be allowed access to the TOE functions and resources after a successful authentication. After 6 failed authentication attempts, access to the token will be blocked. Users must then request a Secure Code that will be send to user's registered phone via SMS. On the Token Manager, users must enter this Secure Code and submit a request to TMS RA in order to unblock or reset the token and User PIN.

- **Cryptographic Operation.** The TOE cryptographic library includes 3-DES, RSA, MD5 and SHA-1 implementations that applications can use for their specific purpose. The TOE performs key generation, encryption and decryption and digital signing.

- **Security Management.** The TOE maintains two (2) distinct roles; **user** and **Security Officer** (SO PIN) to ensure that functions are restricted to those who have the privilege to access them. On the token Manager, the **user** is able to perform the key security functionality of the TOE. On the Token Manager, users must enter the Secure Code and submit a request to the TMS RA in order to unblock or reset User PIN.The SO PIN is used at the back-end process to unblock a token for the user.

2 Conformance Claim (ASE_CCL.1)

The ST and TOE are conformant to version 3.1 (REV 4) of the Common Criteria for Information Technology Security Evaluation.

The following conformance claims are made for the TOE and ST:

- **Part 2 conformant.** Conformant with Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, version 3.1 (REV 4), September 2012
- **Part 3 conformant, EAL2.** Conformant with Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, version 3.1 (REV 4). Evaluation is EAL2, September 2012.

3 Security problem definition (ASE_SPD.1)

3.1 Overview

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

a) a series of ***threats*** that the TOE has been designed to mitigate,

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

c) any relevant ***organisational security policies*** are any statements made in terms of rules or guidelines that must be followed by the TOE and/or the operational environment.

3.2 Threats

Identifier	Threat statement
T.COMINT	An unauthorised user may attempt to compromise the integrity of the data collected, processed and transmitted by the TOE by bypassing a security mechanism.
T.DISCLOSURE	An unauthorised user may attempt to compromise the integrity of the protected resource on the TOE i.e. private cryptographic keys.
T.IMPERSON	An attacker may gain unauthorised access to information or resources by impersonating an authorised user of the TOE.
T.MODIFY	An unauthorised user may attempt to modify the TOE memory to compromise the confidentiality or integrity of the protected resources on the TOE. This may include the unauthorised loading of software onto the TOE.

3.3 Organisational security policies

No organisational security policies have been defined regarding the use of the TOE.

3.4 Assumptions

Identifier	Assumption statement
A.USER	The user is not careless, wilfully negligent or hostile and possesses the necessary privileges to access the information managed by the TOE.
A.HOST	The host computer must be able to communicate correctly. This includes ensuring that TOE is installed and configured correctly, patched and hardened to protect against unauthorized access, modification or deletion.

4 Security objectives (ASE_OBJ.2)

4.1 Overview

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

4.2 Security objectives for the TOE

Identifier	Objective statements
O.AUTHENTICATE	The TOE must ensure that all users are authenticated before they access a protected resource or functions.
O.KEYPROTECT	The TOE shall ensure that all cryptographic keys stored within the TOE are protect sufficiently to prevent their disclosure to a malicious entity.
O.CRYPT	The TOE implements cryptographic functions (RSA, TDES, SHA-1 and MD5) compliant to the relevant industry standards.
O.MODIFY	The TOE shall ensure that protected resources that are stored in the memory are protected against unauthorised modification.

4.3 Security objectives for the environment

Identifier	Objective statements
OE.AUTHDATA	Users of the token must not disclose their PIN.
OE.INSTALL	The TOE shall be delivered, installed, configured and set up in accordance with documented delivery and installation/setup procedures.

4.4 TOE security objectives rationale

This section provides the summary that all security objectives are traced back to aspects of the addressed assumptions and threats.

THREATS/ ASSUMPTIONS / OBJECTIVES	T.COMINT	T.DISCLOSURE	T.IMPERSON	T.MODIFY	A.USER	A.HOST
O.AUTHENTICATE		✓	✓			
O.KEYPROTECT	✓	✓				
O.CRYPT	✓					
O.MODIFY				✓		
OE.AUTHDATA					✓	
OE.INSTALL						✓

4.4.1 Rationale for Security objectives of the TOE

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

Objectives	Rationale
T.COMINT	This threat is traced to: • O.CRYPT will ensure that the data collected and transmitted by TOE using cryptographic algorithms is done so in compliance to standards and protected from attacks that attempt to bypass the security mechanisms of TOE; and • O.KEYPROTECT ensures that all cryptographic keys stored within the TOE are protected sufficiently to prevent their disclosure to a malicious entity.

T.DISCLOSURE	This threat is traced to: • O.AUTHENTICATE, which ensures that all users are authenticated before they access a protected resources or functions; and • O. KEYPROTECT, which ensures that the TOE supports cryptographic functions in secure manner. Cryptographic operations are conducted upon the TOE hardware, meaning the private key does not leave the token.
T.IMPERSON	This threat is traced to: • O.AUTHENTICATE, which ensures the likelihood of a successful impersonation is reduced by the authentication measures (PIN) and token blocking after a threshold of authentication attempts using an incorrect user PIN;
T.MODIFY	This threat is traced to: • O.MODIFY, which ensures the token, will protect resources in memory against unauthorised modification.
A.USER	This assumption is supported by • OE.AUTHDATA, which ensure that the user will know the PIN but not disclose it to anyone else.
A.HOST	This assumption is supported by • OE.INSTALL, which ensure that the installation of the appropriate hardware and software will enable users to access the data stored on the token and use the functions provided by the TOE.

5 Security requirements (ASE_REQ.2)

5.1 Overview

This section defines the security requirements satisfied by the TOE. Each requirement has been extracted from version 3.1 (REV 4) of the Common Criteria, part 2 providing functional requirements and part 3 providing assurance requirements.

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

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

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

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

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

5.2 Security functional requirements

5.2.1 Overview

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

Identifier	Title
FCS_CKM.1a	Cryptographic key generation (TDES)
FCS_CKM.1b	Cryptographic key generation (RSA)
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1a	Cryptographic Operation (TDES)
FCS_COP.1b	Cryptographic Operation (RSA)
FCS_COP.1c	Cryptographic Operation (SHA)
FCS_COP.1d	Cryptographic Operation (MD5)
FIA_AFL.1	Authentication failure handling
FIA_ATD.1	User attribute definition
FIA_UAU.2	User authentication before any action
FMT_MTD.1a	Management of TSF data (User PIN)
FMT_MTD.1b	Management of TSF data (SO PIN)
FMT_MTD.1c	Management of TSF data (block status of user)
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security Roles

5.2.2 FCS_CKM.1a Cryptographic key generation (TDES)

Hierarchical to:	No other components.
FCS_CKM.1a.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**TDES**] and specified cryptographic key sizes [**128 bits**] that meet the following: [**PKCS#11**]
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
Notes:	Note: Key will be generated with a random value and stored in the session memory which can be exported out as describe in pkcs#11 standard

5.2.3 FCS_CKM.1b Cryptographic key generation (RSA)

Hierarchical to:	No other components.
FCS_CKM.1b.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**RSA key generation**] and specified cryptographic key sizes [**1024 and 2048 bits**] that meet the following: [**RSA PKCS#11**].
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
Notes:	Note: RSA Key will be generated inside the token and private key will be stored into the secure memory that cannot be exported out from token.

5.2.4 FCS_CKM.4 Cryptographic key destruction

Hierarchical to:	No other components.
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method: [**overwrite the keys**] that meets the following: [**no standard**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
Notes:	Note: The cryptographic key destruction can be done by overwrite the key.

5.2.5 FCS_COP.1a Cryptographic Operation (TDES)

Hierarchical to:	No other components.
FCS_COP.1a.1	The TSF shall perform [**TDES encryption and decryption**] in accordance with a specified cryptographic algorithm [**TDES-CBC, TDES-ECB**] and cryptographic key sizes [**112 bits for TDES 2 keys, 168 bits for TDES 3 keys**] that meet the following: [**FIPS 46-3**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	None.

5.2.6 FCS_COP.1b Cryptographic Operation (RSA)

Hierarchical to:	No other components.
FCS_COP.1b.1	The TSF shall perform [**RSA encryption and decryption**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**1024 and 2048 bits**] that meet the following: [**RSA PKCS#11**]**.**
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	Note: RSA Signing /Decryption will perform a hardware based encryption that is required to login.

5.2.7 FCS_COP.1c Cryptographic Operation (SHA)

Hierarchical to:	No other components.
FCS_COP.1c.1	The TSF shall perform [**hashing**] in accordance with a specified cryptographic algorithm [**SHA-1**] and cryptographic key sizes [**none**] that meet the following: [**FIPS 180-2**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	None.

5.2.8 FCS_COP.1d Cryptographic Operation (MD5)

Hierarchical to:	No other components.
FCS_COP.1d.1	The TSF shall perform [**hashing**] in accordance with a specified cryptographic algorithm [**MD5**] and cryptographic key sizes [**none**] that meet the following: [**FIPS 180-2**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	None.

5.2.9 FIA_AFL.1 Authentication failure handling

Hierarchical to:	No other components.
FIA_AFL.1.1	The TSF shall detect when [***6***] unsuccessful authentication attempts occur related to [**user entering their passphrase (PIN) for authentication to the TOE**].
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been [***met***], the TSF shall [**block the user usage of the TOE**].
Dependencies:	FIA_UID.1 Timing of identification
Notes:	None.

5.2.10 FIA_ATD.1 User attribute definition

Hierarchical to:	No other components.
FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users: [• **Token Name** • **PIN** • **Certificate name].**
Dependencies:	No dependencies.
Notes:	User PINs are the only attributes maintained by the TOE in the context of this requirement.

5.2.11 FIA_UAU.2 User authentication before any action

Hierarchical to:	FIA_UAU.1 Timing of authentication
FIA_UAU.2.1	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	FIA_UID.1 Timing of identification
Notes:	None.

5.2.12 FMT_MTD.1a Management of TSF data (User PIN)

Hierarchical to:	No other components
FMT_MTD.1a.1	The TSF shall restrict the ability to [***modify***] the [**User PIN**] to [**User**].
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
Notes:	None.

5.2.13 FMT_MTD.1b Management of TSF data (SO PIN)

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

5.2.14 FMT_MTD.1c Management of TSF data (block status of user)

Hierarchical to:	No other components
FMT_MTD.1c.1	The TSF shall restrict the ability to [***modify***] the [**userBlock**] to [**Security Officer**].
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
Notes:	The Security Officer can reset the userBlock so that user can use the TOE again after being unblocked.

5.2.15 FMT_SMF.1 Specification of Management Functions

Hierarchical to:	No other components.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: [**a) changing of PIN** **b) unblock token** **c) import, export, delete digital certificate**]
Dependencies:	No dependencies.
Notes:	None.

5.2.16 FMT_SMR.1 Security Roles

Hierarchical to:	No other components.
FMT_SMR.1.1	The TSF shall maintain the roles [**Security Officer, Users**].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.
Dependencies:	FIA_UID.1 Timing of identification
Notes:	The “Security Officer” stated here is a role that is bounded to the SO PIN upon creation by the developer inside the TOE. The “users” stated here is a role that is bounded to the User PIN upon creation by the developer inside the TOE.

5.3 TOE Security assurance requirements

EAL2 requires evidence relating to the design information and test results, but does not demand more effort on the part of the developer than is consistent with good commercial practice.

EAL2 provides assurance by a full security target and an analysis of the SFRs in that ST, using a functional and interface specification, guidance documentation and a basic description of the architecture of the TOE, to understand the security behaviour.

The analysis is supported by independent testing of the TSF, evidence of developer testing based on the functional specification, selective independent confirmation of the developer test results, and a vulnerability analysis (based upon the functional specification, TOE design, security architecture description and guidance evidence provided) demonstrating resistance to penetration attackers with a basic attack potential.

EAL2 also provides assurance through use of a configuration management system and evidence of secure delivery procedures.

Assurance class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life cycle support	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_CMC.2 Use of a CM system
	ALC_DEL.1 Delivery procedures
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST Introduction
	ASE_OBJ.2 Security objectives

Assurance class	Assurance components
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security Problem Definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_IND.2 Independent testing - sample
	ATE_FUN.1 Functional testing
	ATE_COV.1 Evidence of coverage
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

5.4 Security requirements rationale

5.4.1 Dependency rationale

The table below demonstrates the mutual supportiveness of the SFRs for the TOE by demonstrating how the SFR dependencies are fulfilled by the TOE and by justifying those dependencies that are not fulfilled.

The SARs relevant to the TOE constitute an evaluation assurance level of EAL2, as defined in the Common Criteria and include no extensions or augmentations. Therefore, as a complete evaluation assurance level, they are a mutually supportive set and require no further justification.

SFR	Dependency	Inclusion
FCS_COP.1a	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1a FCS_CKM.4
FCS_COP.1b	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1b FCS_CKM.4

SFR	Dependency	Inclusion
FCS_COP.1c	[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	None. No keys are needed for hashing
FCS_COP.1d	[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	None. No keys are needed for hashing
FCS_CKM.1a	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1a FCS_CKM.4
FCS_CKM.1b	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1b FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1a FCS_CKM.1b
FIA_AFL.1	FIA_UAU.1 Timing of authentication	FIA_UAU.2
FIA_ATD.1	No dependencies	NA
FIA_UAU.2	FIA_UID.1 Timing of identification	None. There are only 2 users for the TOE. As Security Officer and a user. There are identified by the interfaces where the users use for authentication.
FMT_MTD.1a	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1b	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1

SFR	Dependency	Inclusion
FMT_MTD.1c	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
FMT_SMF.1	No dependencies	N/A
FMT_SMR.1	FIA_UID.1 Timing of identification	None. There are only 2 users for the TOE. As Security Officer and a user. There are identified by the interfaces where the users use for authentication.

5.4.2 Mapping of SFRs to security objectives for the TOE

Security objective	Mapped SFRs	Rationale
O.AUTHENTICATE	FIA_UAU.2	The requirement helps meet the objective by authenticating user before any TSF mediated actions.
O.CRYPT	FCS_COP.1	Perform cryptographic operation in accordance with a specified cryptographic algorithm.
	FCS_CKM.1	Generate cryptographic keys in accordance with a specified cryptographic key generation algorithm
	FCS_CKM.4	The requirement helps meet the objective by destroying cryptographic keys in accordance with a specified cryptographic key destruction method.
O.MODIFY	FIA_ATD.1	The requirement helps meet the objective by ensuring user security attributes are maintained.
	FMT_MTD.1	The requirement helps meet the objective by restricting the ability to modify the SO PIN.
O.DISCLOSURE	FIA_AFL.1	The requirement helps meet the objective by blocking authentication failure after number of attempt.
	FMT_SMR.1	The requirement helps meet the objective by providing user timing of identification.
	FMT_SMF.1	The requirement helps meet the objective by providing management functions of the TOE for authenticated user.

5.4.3 Explanation for selecting the SARs

The assurance package for the evaluation of the TOE is Evaluation Assurance Level 2 (EAL2).

The TOE is intended to provide a number of capabilities, which are designed to support organisations to rapidly develop and deploy enterprise PKI-related security solutions. EAL2 is sufficient to demonstrate that the TOE is resistant to attackers with a Basic attack potential.

6 TOE summary specification (ASE_TSS.1)

6.1 Overview

This section provides the TOE summary specification, a high-level description of how the TOE actually implements the claimed security functional requirements.

The TOE security functions include the following:

- **User Authentication:** The TOE requires that each user is successfully identified and authenticated before any interaction with protected resources is permitted;
- **Cryptographic Operation:** The TOE provides key generation, key destruction and operations; and
- **Security Management.** The TOE provides functions that allow management of the TOE and its security functions.

6.2 Cryptographic Operation

The TOE performs RSA, Triple DES, SHA-1 and MD5 operations. (FCS_COP.1a, FCS_COP.1b, FCS_COP.1c, FCS_COP.1d). It provides key generation for RSA and Triple DES (FCS_CKM.1a, FCS_CKM.1b). The TOE also performs key destruction by overwriting the memory space of the keys. This function overwrites the previous keys when the user deletes the expired certificate(s) and key(s) (FCS_CKM.4).

These operations are used by applications on the host system. The applications must comply with PKCS#11 and CSP standards and use the middleware of the TOE to access these functionalities provided by the TOE.

6.3 User Authentication

User of the TOE can authenticate to the TOE through the Token Manager application, which is outside the scope of evaluation. The Token Manager provides an interface to access management functions.

User must enter their PIN for authentication. Only after a successful authentication attempt will the TOE allows the users to access the TOE functions. (FIA_UAU.2)

The TOE maintains the status of user authentication by changing the authenticity value of a user to true if the Security Officer or user is successfully authenticated. The default value of the authenticity value is always false. (FIA_ATD.1)

Only the SO PIN is able to unblock a token when the token has been blocked due to failed authentication attempts. When this happens, the TOE will prevent all access to the TOE and TOE functionality. To unblock the token, the user must request a Secure Code from TMS RA. The TMS RA will send a Secure Code to user's registered mobile phone via SMS. On the Token Manager platform, a user must enter the Secure Code in order to decrypt the SO PIN. The Token Manager unblocks the token by calling PKCS#11 C_InitPIN function (FIA_AFL.1).

6.4 Security Management

The TOE maintains 2 roles - User and Security Officer (FMT_SMR.1). The list below describes the management functions available to user and Security Officer (FMT_SMF.1, FMT_MTD.1a, FMT_MTD.1b and FMT_MTD.1c).

- token with default passwords (Security Officer);
- changing of user PIN (user);
- unblock token (Security Officer) - Secure Code is used to decrypt SO PIN; and
- import, export, delete digital certificate, enrolment and renewal (user)