Public Key Infrastructure Framework (PKIF)

Version 1.2

PKIF Security Target

Version 1.63
Date: December 6, 2005

Prepared for: US Marine Corps

Authors

Jean Petty, CygnaCom Solutions, Inc
Peter Kukura, CygnaCom Solutions, Inc
Santosh Chokhani, CygnaCom Solutions, Inc
Carl Wallace, CygnaCom Solutions, Inc

Foreword

This Security Target (ST) defines the PKI Framework (PKIF), a C++ software library designed to simplify the task of adding PKI support to applications. This ST is conformant with a PP that is validated under the Common Criteria Evaluation and Validation Scheme, Version: 2.61, of *U.S. Government PKE PP with:*

1. *Certification Path Validation (CPV) – Basic Package,*
2. *CPV – Basic Policy Package,*
3. *CPV – Policy Mapping Package,*
4. *CPV – Name Constraints Package,*
5. *PKI Signature Generation Package,*
6. *PKI Signature Verification Package,*
7. *PKI Encryption using Key Transfer Algorithms Package,*
8. *PKI Decryption using Key Transfer Algorithms Package,*
9. *Online Certificate Status Protocol (OCSP) Client Package, and*
10. *Certificate Revocation List (CRL) Validation Package*

at EAL4 with augmentation. (Date: July 31, 2004; prepared for: US Marine Corps)

Revision History

Version	Date	ST Author	Description
0.1	July 14, 2003		Initial version of the Security Target.
0.2	July 25, 2003		In-progress version of the Security Target
1.0	February 3, 2004		Draft at EAL4
1.1	April 14, 2004		Incorporation of EOR comments and additional review comments provided by Carl Wallace.
1.2	August 11, 2004		Updates based on EORs.
1.3	February 28, 2005		Minor revisions, updated TSFI definition
1.4	May 8, 2005	Chokhani	Accepted past changes, aligned the list of PKIF functions in Sections 1, 2, and 6; listed TSF data and user data in Section 2; included SFR mapping to SF in TSS section.
1.5	May 16, 2005	Chokhani	Added the SFR to SF mapping in Section 8.3
1.6	June 15, 2005	Kukura	Updated text throughout to match v2.61 PP
1.61	June 30, 2005	Wallace	Accepted many suggested changes, clarified and added text per comments
1.62	July 21, 2005	Chokhani	Aligned OCSP time checks and CRL time checks
1.63	December 6, 2005	Galustyan	Added CPKIFSignedDate::Decode function to the table in section 6.3.

Table of Contents

Page

1 Introduction 1
1.1 Identification 1
1.2 Overview 1
1.3 Related Documents 1
1.4 Organization 2
1.5 Common Criteria Conformance 3
2 TOE Description 5
2.1 Overview 5
2.2 TOE Description 5
2.2.1 Certification Path Processing 5
2.2.2 Signature Generation Functionality 6
2.2.3 PKI Signature Verification Functionality 6
2.2.4 PKI Encryption using Key Transfer Algorithms Functionality 6
2.2.5 PKI Decryption using Key Transfer Algorithms Functionality 6
2.2.6 Online Certificate Status Protocol Client Functionality 6
2.2.7 Certificate Revocation List functionality 6
2.2.8 Symmetric key encryption and decryption 7
2.2.9 ASN.1 encoding/decoding 7
2.3 Roles, User Data, and TSF Data 7
2.4 TOE Environment Description 7
2.5 PP Conformance 8
2.6 Assurance Requirements 9
3 TOE Security Environment 10
3.1 Secure Usage Assumptions for the IT Environment 10
3.2 Threats to Security for the TOE 11
3.3 Threats to Security for IT Environment 13
4 Security Objectives 14
4.1 Security Objectives for the TOE 14
4.2 Security Objectives for the IT Environment 15
5 IT Security Requirements 17
5.1 Security Functional Requirements for the TOE 18
5.1.1 Class FDP – User Data Protection 19
5.2 Security Functional Requirements for the IT Environment 26
5.2.1 Class FCS – Cryptographic Support 27
5.2.2 Class FDP – User Data Protection 27
5.2.3 Class FIA – Identification and Authentication 28
5.2.4 Class FMT – Security Management 29
5.2.5 Class FPT – Protection of the TOE Security Functions 30

5.2.6 Strength of Function Requirement 31
5.3 Assurance Requirements 32
6 TOE Summary Specification...33
6.1 Certification Path Processing, CRL Processing and OCSP Processing 33
6.2 Signature Generation Functionality 35
6.3 PKI Signature Verification Functionality 35
6.4 PKI Encryption using Key Transfer Algorithms Functionality 36
6.5 PKI Decryption using Key Transfer Algorithms Functionality 36
6.6 Supporting Functionality...37
6.7 Assurance Measures...40
7 PP Conformance 44
7.1 Conformance with PP Requirements 44
7.2 Conformance with PP Assumptions 44
7.3 Conformance with PP Threats..46
7.3.1 Conformance with PP Threats to TOE Security 46
7.3.2 Conformance with PP Threats to IT Environment Security 48
7.4 Conformance with PP Objectives 49
7.4.1 Conformance with PP Objectives for IT Environment 49
7.4.2 Conformance with PP Objectives for TOE 51
8 Rationale...53
8.1 Security Objectives Rationale...53
8.1.1 Base and Environmental Security Objectives Rationale for TOE.......................53
8.1.2 Security Objectives Rationale for the TOE..56
8.2 Security Requirements Rationale 64
8.2.1 Functional Security Requirements Rationale 64
8.2.2 Assurance Requirement Rationale...76
8.2.3 Strength of Function Rationale...76
8.2.4 Security Functional Requirements Dependencies Rationale 76
8.3 TOE Summary Specification Rationale 78
References...81
Glossary of Terms..82
List of Acronyms 85

List of Tables

Page

Table 3.1 – Assumptions for the IT Environment 10
Table 3.2 – Threats for the TOE 11
Table 3.3 – Threats to Security for the IT Environment 13
Table 4.1 – Security Objectives for the TOE 14
Table 4.2 – Security Objectives for the IT Environment 15
Table 5.1 – Part 2 or Part 2 Extended Requirements 17
Table 5.2 – Security Functional Requirements for the TOE 18
Table 5.3 – Security Functional Requirements for the IT Environment 26
Table 5.4 – EAL4 with Augmentation Assurance Requirements 32
Table 6.1 Primary Path Processing and Revocation Status-related Interfaces 34
Table 6.2 Primary Signature Generation-related Interfaces 35
Table 6.3 Primary Signature Verification-related Interfaces 35
Table 6.4 Primary PKI Encryption-related Interfaces 36
Table 6.5 Primary PKI Decryption using Key Transfer Algorithms-related Interfaces 37
Table 6.6 Assurance Measures and How Satisfied 40
Table 7.1 – Conformance with PP Base Assumptions for IT Environment 44
Table 8.1 – Mapping the Base Assumptions and Threats to Objectives 53
Table 8.2 – Mapping of Base TOE and Environmental Objectives to Threats and Assumptions 56
Table 8.3 – Mapping of TOE Security Threats to Objectives 57
Table 8.4 – Mapping of TOE Security Objectives to Threats 61
Table 8.5 – Security Objective to Functional Component Mapping 64
Table 8.6 – Functional Requirements Dependencies 76
Table 8.7 – Mapping from SFR to IT Security Function 78

1 Introduction

This section contains document management and overview information. The Security Target (ST) Identification provides the labeling and descriptive information necessary to identify, catalogue, register, and cross-reference an ST. The Overview summarizes the ST in narrative form and provides sufficient information for a potential user to determine whether the ST is of interest. The overview can also be used as a standalone abstract for ST catalogues and registers.

1.1 Identification

TOE Identification: Public Key Infrastructure Framework

TOE Version Number: Version 1.2

ST Title: Public Key Infrastructure Framework (PKIF) Security Target

ST Version Number: Version 1.63

ST Date: December 6, 2005

ST Authors: Jean Petty, CygnaCom Solutions, Inc.; Peter Kukura, CygnaCom Solutions, Inc.; Santosh Chokhani, CygnaCom Solutions, Inc.; Carl Wallace, CygnaCom Solutions, Inc.

Assurance Level: EAL4, augmented with ALC_FLR.1, Basic flaw remediation

Strength of Function: Not Applicable

Sponsoring Organization: United States Marine Corps (USMC)

Registration: <To be filled in upon registration>

Keywords: Public Key Enabled (PKE), PKE, Public Key Infrastructure (PKI), PKI

1.2 Overview

This Security Target (ST) describes the PKIF, a C++ software library designed to simplify the task of adding PKI support to applications. PKIF provides application developers a set of extensible classes, packaged as a Windows dynamic link library (DLL), that perform a variety of PKI-related functions including:

- Certification Path Processing
- CMS based Signature Generation
- Verification of signatures on CMS messages using PKI
- PKI Encryption using Key Transfer Algorithms functionality
- PKI Decryption using Key Transfer Algorithms functionality
- Online Certificate Status Protocol Client functionality
- Certificate revocation list processing functionality
- ASN.1 encoding/decoding functionality
- Cryptographic message creation and processing (CMS format)

1.3 Related Documents

- International Organization for Standards/Internet Electrotechnical Committee (ISO/IEC) 9594-8:2001"Information Technology- Open Systems Interconnection-The Directory: Public Key and Attribute Certificate Frameworks" (X.509 Standard)

- X.509 Internet Public Key Infrastructure Certificate and CRL Profile, RFC 3280, April 2002
- X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP), RFC 2560 June 1999.
- ISO/IEC 15408:2004 Information technology — Security techniques — Evaluation criteria for IT security
- FIPS 140-2, Security Requirements for Cryptographic Modules, 25 May 2001 http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
- Cryptographic Message Syntax (CMS), RFC 3369, August 2002

1.4 Organization

The main sections of the ST are the TOE Description, TOE Security Environment, Security Objectives, IT Security Requirements, TOE Summary Specification, PP Conformance, and Rationale.

Section 2, the TOE Description, provides general information about the TOE, serves as an aid to understanding its security requirements, and provides context for the ST's evaluation.

The TOE Security Environment in Section 3 describes security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes:

a) Assumptions regarding the TOE's intended usage and environment of use
b) Threats relevant to secure TOE operation
c) Organizational security policies with which the TOE must comply

Section 4 contains the security objectives that reflect the stated intent of the ST. The objectives define how the TOE will counter identified threats and how it will cover identified organizational security policies and assumptions. Each security objective is categorized as being for the TOE or for the environment.

Section 5 contains the applicable Security Requirements taken from the Common Criteria, with appropriate refinements. The requirements are provided in separate subsections for the TOE and its environment. The IT security requirements are subdivided as follows:

a) TOE Security Functional Requirements
b) TOE Security Assurance Requirements

Section 6 contains the TOE Summary Specification.

Section 7 contains the PP Conformance.

The Rationale in Section 8 presents evidence that the ST is a complete and cohesive set of requirements and that a conformant TOE would provide an effective set of IT security countermeasures within the security environment. The Rationale is in three main parts. First, a Security Objectives Rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them. Then, a Security Requirements Rationale demonstrates that the security requirements (TOE and environment) are traceable to the security objectives and are suitable to meet them. Finally, a PP Rationale shows how the assumptions, threats, objectives and requirements in the ST map to those in the PP.

A glossary of PKI-related terms used in this ST is provided in the Appendix followed by a list of acronyms.

1.5 Common Criteria Conformance

This Security Target has been built with Common Criteria (CC) Version 2.2 (ISO/IEC 15408 Evaluation Criteria for Information Technology Security; Part 1: Introduction and general model, Part 2: Security functional requirements, and Part 3: Security assurance requirements).

This Security Target is Common Criteria Version 2.2, Part 2 extended, and Part 3 conformant, at Evaluation Assurance Level 4 with Augmentation. The definition of Part 2 extended is found in the CC Part 1, section 5.1.3, “Part 2 extended - A PP or TOE is Part 2 extended if the functional requirements include functional components not in Part 2.”

This ST is conformant with the *U.S. Government PKE PP with:*

1. *Certification Path Validation (CPV) – Basic Package,*
2. *CPV – Basic Policy Package,*
3. *CPV – Policy Mapping Package,*
4. *CPV – Name Constraints Package,*
5. *PKI Signature Generation Package,*
6. *PKI Signature Verification Package,*
7. *PKI Encryption using Key Transfer Algorithms Package,*
8. *PKI Decryption using Key Transfer Algorithms Package,*
9. *Online Certificate Status Protocol (OCSP) Client Package, and*
10. *Certificate Revocation List (CRL) Validation Package*

at EAL4 with augmentation. (Version: 2.61; Date: July 31, 2004; Prepared for: US Marine Corps)

2 TOE Description

2.1 Overview

PKIF is a C++ software library designed to simplify the task of adding PKI support to applications. PKIF provides application developers a set of extensible classes that perform a variety of PKI-related functions including:

- Certification Path Processing
- CMS based Signature Generation
- Verification of signatures on CMS messages using PKI
- PKI Encryption using Key Transfer Algorithms functionality
- PKI Decryption using Key Transfer Algorithms functionality
- Online Certificate Status Protocol Client functionality
- Certificate revocation list processing functionality
- ASN.1 encoding/decoding functionality
- Cryptographic message creation and processing (CMS format)

Note, for cryptographic functions, cryptographic key lengths supported by PKIF are not a function of the PKIF DLL, but rather, are determined by the capabilities of the relevant CSP.

2.2 TOE Description

2.2.1 Certification Path Processing

PKIF performs X.509 certification path processing, including certification path development and certification path validation. Certification path validation consists of validating certificates starting with the one certified by a trust anchor and ending with the one issued to the subscriber of interest. PKIF supports X.509 version 3 Certificates and X.509 CRLs, versions 1 and 2. All processing is X.509 and PKIX RFC3280 compliant.

There are three types of public key certificates involved in certificate path validation:

- Trust anchor (TA) certificates: These are certificates containing public keys that do not require any validation. Trust anchors generally take the form of a self-signed certificate. TAs must be delivered to entities that rely on the TA's public key using trusted means. The primary purpose of the trust anchor is to provide a means of conveying a Distinguished Name (DN), public key, algorithm identifier, and the public key parameters (if applicable) for use in validating certification paths.
- Intermediate certificates: These are the certificates issued to CAs. All certificates in a certification path are intermediate certificates, except the trust anchor certificate and end entity certificate.
- End certificates: This is the last certificate in the certification path and is issued to the subscriber of interest. A subscriber certificate is also called an end-entity certificate (i.e., a certificate issued to an entity not functioning as a CA). Sometimes, the last

certificate can be a CA certificate, e.g., when the certification path is used to verify signature on a CRL.

PKIF processes the following security-related certificate extensions: ocsp-nocheck, keyUsage, extendedKeyUsage, and basicConstraints. PKIF performs the processing of the following certificate policy-related extensions: certificatePolicies, policyMapping, inhibitAnyPolicy, policyConstraints, and nameConstraints extensions

By default, PKIF assumes that the path validation is being done as of the current system time, as opposed to verification of signature relative to a point in time in the past. However, applications can specify a time other than the current time for use during path validation.

2.2.2 Signature Generation Functionality

PKIF enables applications to use a private key for signature generation and to specify information covered by that signature and packaged with the signature, e.g. using the CMS SignedData format.

2.2.3 PKI Signature Verification Functionality

PKIF enables applications to process signature information, e.g. using the CMS SignedData format, and to verify signatures using a public key.

2.2.4 PKI Encryption using Key Transfer Algorithms Functionality

PKIF enables applications to perform public key encryption using key transfer algorithms such as RSA.

2.2.5 PKI Decryption using Key Transfer Algorithms Functionality

PKIF enables applications to perform private key decryption using key transfer algorithms such as RSA.

2.2.6 Online Certificate Status Protocol Client Functionality

PKIF can generate Online Certificate Status Protocol (OCSP) requests and validate OCSP responses to determine the revocation status of public key certificates. PKIF verifies OCSP Responder as a trust anchor, as a CA, or as an end entity authorized to sign OCSP responses. PKIF establishes trust in the OCSP responder certificates by performing Certification Path Validation.

2.2.7 Certificate Revocation List functionality

PKIF provides Certificate Revocation List (CRL) validation functionality that enables applications to determine the revocation status of a certificate using a CRL. PKIF may be used to process CRLs obtained from a variety of sources including: locations indicated by a CRL Distribution Point (CRLDP) extension in a certificate, local storage facilities or LDAP-accessible directories.

PKIF permits the use of the same public key for CRL signature verification as the one used for verifying the signature on the certificate, but does not mandate it. In other words, a PKIF will develop and validate certification paths to CRL signers where necessary.

2.2.8 Symmetric key encryption and decryption

PKIF provides functionality to perform symmetric key encryption and decryption using algorithms including DES and Triple DES.

2.2.9 ASN.1 encoding/decoding

PKIF performs decoding of objects in support of processing related to X.509, RFC3280, OCSP and CMS. PKIF performs encoding of objects in support of processing related to OCSP and CMS.

2.3 Roles, User Data, and TSF Data

PKIF is a toolkit used by application developers to incorporate secure PKI functionality into an application; PKIF has only one role: user. The user is considered to be the application using PKIF, or, to provide a human definition, the application developer.

TOE user data is defined as any data that is passed to or returned from PKIF. This includes data that is encrypted, decrypted, signed, and verified or information used in support operations on such data. Trust anchors, certificates, CRLs, OCSP requests and responses are also user data.

Note that, for PKIF, the TOE environment performs the identification and authentication (I&A) functions. Therefore, data associated with I&A is not considered TSF data, since it is not within the TOE boundary. Similarly, private keys are managed by FIPS 140-2 validated cryptographic modules present in the environment and are not considered TSF data. Thus, there are no TSF data in PKIF.

2.4 TOE Environment Description

PKIF is intended for use with Microsoft Visual C++ .NET 2002. All references to IDE dialogs, property pages, fields, etc. assume use of Microsoft Visual C++ .NET 2002.

PKIF is designed to operate with any CAPI-compatible cryptographic module, including middleware that interacts with Common Access Cards (CAC). CACs are cryptographic modules that are validated at FIPS 140 series Level 1 or greater. Cryptographic modules may perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, Hash-based Message Authentication Code (HMAC) and/or other required cryptographic functions.

Certificates and revocation status information, i.e., CRLs or OCSP responses, are included in the environment and are available as part of the interface to a PKI.

PKIF is intended for use on PCs running Windows 2000. Windows 2000 includes LDAP and HTTP client functionality. Windows 2000 includes a CAPI-compatible FIPS 140 Level 1 validated cryptographic module. In addition, the configuration includes the ActivCard CAC CSP.

The hardware configuration includes any PC with at least 128MB RAM, 20 GB hard drive, display, keyboard, mouse and, optionally, a smart card reader and CAC.

PKIF will build and validate certification paths to any trust anchor. For example, in order to use PKIF with a DoD-issued CAC, the DoD Class 3 Root needs to be included as one of the trust anchors in CAPI or otherwise made available to PKIF as a trust anchor. While operational DoD systems have the requirements to delete various trust anchors except for those required by Microsoft, the evaluated configuration does not depend on that requirement.

When using a CAC, the user certificates associated with the private keys stored on the CAC must be imported into a CAPI certificate store and associated with the CAC.

PKIF can be configured to search an application specified LDAP-accessible directory or to retrieve certificates and CRLs from HTTP or LDAP URLs included in certificates. To obtain information via HTTP or LDAP, the workstation must have network connectivity and access to the servers of interest. The evaluated configuration permits sufficient network connectivity.

Windows Registry Settings

Release builds of PKIF do not utilize any PKIF-specific Windows registry entries.

ActivCard CAC CSP Installation

ActivCard CAC CSP installation is done from an administrator account using the ActivCard installation application.

PKIF Installation

PKIF is installed by any person with administrative privileges on the workstation on which PKIF is being installed. PKIF is installed using the PKIF installation application.

Identification and Authentication

Windows 2000 provides I&A. I&A is useful for access control of resources managed by Windows including files, folders, CAPI certificate stores, private keys, and audit logs (audit logs are maintained in a specific folder in the file system hierarchy). Windows 2000 I&A is used for identifying the event-causing subject and for identification of roles.

2.5 PP Conformance

This ST is conformant with the *U.S. Government PKE PP with:*

1. *Certification Path Validation (CPV) – Basic Package,*
2. *CPV – Basic Policy Package,*
3. *CPV – Policy Mapping Package,*
4. *CPV – Name Constraints Package,*
5. *PKI Signature Generation Package,*
6. *PKI Signature Verification Package,*
7. *PKI Encryption using Key Transfer Algorithms Package,*
8. *PKI Decryption using Key Transfer Algorithms Package,*
9. *Online Certificate Status Protocol (OCSP) Client Package, and*
10. *Certificate Revocation List (CRL) Validation Package*

at EAL4 with augmentation. (Version: 2.61; Date: July 31, 2004; Prepared for: US Marine Corps)

The PP to which this ST conforms defines functionality in terms of "packages." A package, as defined by the CC, is an intermediate combination of functional or assurance components that define requirements that meet an identifiable set of security objectives. A package may be thought of as a set of defined security requirements for a specific function. Note that in this ST, all requirements are defined as either in the TOE or in the environment and there are no package distinctions made. To make it easier for the ST evaluator, however, cross references to PP packages and ST components are provided in Section 8, Rationale.

2.6 Assurance Requirements

The assurance level selected for PKIF is EAL4 with augmentation; EAL4 was selected because PKIF users require a moderate to high level of independently assured security.

EAL4 provides added assurance from positive security engineering based on good commercial development practices, which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest assurance level at which it is likely to be economically feasible to retrofit to an existing product line. ALC_FLR.1 is added to provide basic flaw remediation.

3 TOE Security Environment

3.1 Secure Usage Assumptions for the IT Environment

Table 3.1 lists the Secure Usage Assumptions for the IT environment.

Table 3.1 – Assumptions for the IT Environment

#	Assumption Name	Description
1	AE.Authorized_Users	Authorized users are trusted to perform their assigned functions.
2	AE.Configuration	The TOE will be properly installed and configured.
	AE.Crypto_Module	The TOE environment is assumed to include one or more cryptographic module(s) that are all validated at FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, Hash based Message Authentication Code (HMAC) and/or other required cryptographic functions. In summary, all cryptographic modules in the TOE shall be validated at FIPS 140 series Level 1.
4	AE.Low	The attack potential on the TOE is assumed to be low.
5	AE.Physical_Protection	Physical protection is assumed to be provided by the environment. The TOE hardware and software is assumed to be protected from unauthorized physical access.
6	AE.PKI_Info	The certificate and certificate revocation status information is available to the TOE for the time of interest (TOI).
7	AE.Time	Accurate system time with required precision in GMT format is assumed to be provided by the environment.

3.2 Threats to Security for the TOE

Table 3.2 defines security threats for the TOE. The asset under attack is the information transiting the TOE. In general, the threat agent includes, but is not limited to: 1) people with TOE access who are expected to possess "average" expertise, few resources, and moderate motivation, or 2) failure of the TOE.

Table 3.2 – Threats for the TOE

#	Threat Name	Threat Description
1	T.Certificate_Modi	An untrusted user may modify a certificate resulting in using a wrong public key.
2	T.DOS_CPV_Basic	Revocation information or access to revocation information could be made unavailable, resulting in loss of system availability.
3	T.Expired_Certificate	An expired (and possibly revoked) certificate as of TOI could be used for signature verification.
4	T.Masquarade	An untrusted entity (e.g. acting as a CA) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users.
5	T.No_Crypto	The user public key and related information may not be available to carry out the cryptographic function.
6	T.Path_Not_Found	A valid certification path is not found due to lack of system functionality.
7	T.Revoked_Certificate	A revoked certificate could be used as valid, resulting in security compromise.
8	T.User_CA	A user could act as a CA, issuing unauthorized certificates.
9	T.Unknown_Policies	The user may not know the policies under which a certificate was issued.
10	T.Mapping	The user may accept unacceptable certificates or reject acceptable certificates due to improper certificate policy mapping.
11	T.Wrong_Policy_Dec	The user may accept certificates that were not generated with the diligence and security acceptable to the user. The user may reject certificates that were generated with the diligence and security acceptable to the user.
12	T.Name_Collision	The user may accept certificates from CA where the CA's understanding and the user's understanding of the names differ, i.e., user and CA associate different identity with the same name.
13	T.Clueless_PKI_Sig	The user may try only inappropriate certificates for signature verification because the signature does not include a hint.
14	T.Assumed_Identity_PKI_Ver	A user may assume the identity of another user in order to verify a PKI signature.

#	Threat Name	Threat Description
15	T.Clueless_PKI_Ver	The user may try only inappropriate certificates for signature verification because hints in the signature are ignored.
16	T.Assumed_Identity_WO_En	A user may assume the identity of another user in order to perform encryption using Key Transfer algorithms.
17	T.Clueless_WO_En	The user may try only inappropriate certificates for encryption using Key Transfer algorithms in absence of hint.
18	T.Garble_WO_De	The user may not apply the correct key transfer algorithm or private key, resulting in garbled data.
19	T.DOS_OCSP	The OCSP response or access to the OCSP response could be made unavailable, resulting in loss of system availability.
20	T.Replay_OCSP_Info	The user may accept an OCSP response from before TOI resulting in acceptance of a revoked certificate.
21	T.Wrong_OCSP_Info	The user may accept a revoked certificate or reject a valid certificate due to a wrong OCSP response.
22	T.DOS_CRL	The CRL or access to CRL could be made unavailable, resulting in loss of system availability.
23	T.Replay_Revoc_Info_CRL	The user may accept a CRL issued before TOI resulting in acceptance of a revoked certificate.
24	T.Wrong_Revoc_Info_CRL	The user may accept a revoked certificate or reject a valid certificate due to a wrong CRL.

3.3 Threats to Security for IT Environment

This subsection defines the threats to the IT Environment, included in Table 3.3, below. The asset under attack is the information transiting the TOE. In general, the threat agent includes, but is not limited to: 1) people with TOE access who are expected to possess “average” expertise, few resources, and moderate motivation, or 2) failure of the TOE.

Table 3.3 – Threats to Security for the IT Environment

#	Threat Name	Threat Description
1E	TE.Attack	An undetected compromise of the TOE assets may occur as a result of an attacker (whether an insider or outsider) attempting to perform actions that the individual is not authorized to perform.
2E	TE.Bypass	An unauthorized individual or user may tamper with security attributes or other data in order to bypass TOE security functions and gain unauthorized access to TOE assets.
3E	TE.Imperson	An unauthorized individual may impersonate an authorized user of the TOE and thereby gain access to TOE data, keys, and operations.
4E	TE.Modify	An attacker may modify TSF or user data, e.g., stored security attributes or keys, in order to gain access to the TOE and its assets.
5E	TE.Object_Init	An attacker may gain unauthorized access to an object upon its creation, if the security attributes are not assigned to the object or any one can assign the security attributes upon object creation.
6E	TE.Private_Key	An attacker may assume the identity of a user by generating or using the private key of the user.
7E	TE.Role	A user may assume more privileged role than permitted and use the enhanced privilege to take unauthorized actions.
8E	TE.Secure_Attributes	A user may be able to change the security attributes of an object and gain unauthorized access to the object.
9E	TE.Shoulder_Surf	An authorized user may look over the shoulder of the authorized user while authentication is in progress and read the authentication information.
10E	TE.Tries	An unauthorized individual may guess the authentication information using trial and error.

4 Security Objectives

4.1 Security Objectives for the TOE

Table 4.1 defines the security objectives for TOE.

Table 4.1 – Security Objectives for the TOE

#	Objective Name	Objective Description
1	O.Availability	The TSF shall continue to provide security services even if revocation information is not available.
2	O.Correct_Temporal	The TSF shall provide accurate temporal validation results.
3	O.Current_Certificate	The TSF shall only accept certificates that are not expired as of TOI.
4	O.Get_KeyInfo	The TSF shall provide the user public key and related information in order to carry out cryptographic functions.
5	O.Path_Find	The TSF shall be able to find a certification path from a trust anchor to the subscriber.
6	O.Trusted_Keys	The TSF shall use trusted public keys in certification path validation.
7	O.User	The TSF shall only accept certificates issued by a CA.
8	O.Verified_Certificate	The TSF shall only accept certificates with verifiable signatures.
9	O.Valid_Certificate	The TSF shall use certificates that are valid, i.e., not revoked.
10	O.Provide_Policy_Info	The TSF shall provide certificate policies for which the certification path is valid.
11	O.Map_Policies	The TSF shall map certificate policies in accordance with user and CA constraints.
12	O.Policy_Enforce	The TSF shall validate a certification path in accordance with certificate policies acceptable to the user.
13	O.Authorised_Names	The TSF shall validate a certificate only if the CA is authorized to issue a certificate to the subject.
14	O.Give_Sig_Hints	The TSF shall provide hints for selecting correct certificates for signature verification.
15	O.Use_Sig_Hints	The TSF shall use hints for selecting correct certificates for signature verification.
16	O.Linkage_Sig_Ver	The TSF shall use the correct user public key for signature verification.
17	O.Hints_Enc_WO	The TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer Algorithms.

#	Objective Name	Objective Description
18	O.Linkage_Enc_WO	The TSF shall use the correct user public key for key transfer.
19	O.Correct_KT	The TSF shall use appropriate private key and key transfer algorithm.
20	O.Accurate_OCSP_Info	The TSF shall accept only accurate OCSP responses.
21	O.Auth_OCSP_Info	The TSF shall accept the revocation information from an authorized source for OCSP transactions.
22	O.Current_OCSP_Info	The TSF accept only OCSP responses current as of TOI.
23	O.User_Override_Time_OCSP	The TSF shall permit the user to override the time checks on the OCSP response.
24	O.Accurate_Rev_Info	The TSF shall accept only accurate revocation information.
25	O.Auth_Rev_Info	The TSF shall accept the revocation information from an authorized source for CRL.
26	O.Current_Rev_Info	The TSF shall accept only CRL that are current as of TOI.
27	O.User_Override_Time_CRL	The TSF shall permit the user to override the time checks on the CRL.

4.2 Security Objectives for the IT Environment

Security objectives for the IT Environment are defined in Table 4.2, below.

Table 4.2 – Security Objectives for the IT Environment

#	Objective Name	Objective Description
1E	OE.DAC	The IT environment shall control and restrict user access to the TOE assets in accordance with a specified access control policy.
2E	OE.I&A	The IT environment shall uniquely identify all users, and shall authenticate the claimed identify before granting a user access to the TOE facilities.
3E	OE.Init_Secure_Attr	The IT environment shall provide valid default security attributes when an object is initialized.
4E	OE.Invoke	The IT environment shall be invoked for all actions.
5E	OE.Limit_Actions_Auth	The IT environment shall restrict the actions a user may perform before the TSF verifies the identity of the user.
6E	OE.Limit_Tries	The IT environment shall restrict the number of consecutive unsuccessful authentication attempts.
7E	OE.No_Echo	The IT environment shall not echo the authentication information.

#	Objective Name	Objective Description
8E	OE.Protect_I&A_Data	The IT environment shall permit only authorized users to change the I&A data.
9E	OE.Secure_Attributes	The IT environment shall permit only the authorized users to change the security attributes.
10E	OE.Security_Roles	The IT environment shall maintain security-relevant roles and association of users with those roles.
11E	OE.Self_Protect	The IT environment shall maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure.
12E	OE.Trust_Anchor	The IT environment shall permit only authorized users to manage the trust anchors.
13E	OE.TSF_Data	The IT environment shall permit only authorized users to modify the TSF data.
14E	OE.Authorized_Users	Authorized users are trusted to perform their authorized tasks.
15E	OE.Configuration	The TOE shall be installed and configured properly for starting up the TOE in a secure state.
16E	OE.Crypto	The IT environment shall include one or more cryptographic modules that are validated at FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, HMAC and/or other required cryptographic functions. In summary, all cryptographic modules within the TOE shall be FIPS 140 series level 1 validated.
17E	OE.Low	The Identification and Authentication functions in the TOE shall be designed and implemented for a minimum attack potential of low as validated by the vulnerability assessment and Strength of Function analyses.
18E	OE.Physical_Security	The IT environment shall provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.
19E	OE.PKI_Info	The IT environment shall provide the TOE certificate and certificate revocation information for the time of interest (TOI).
20E	OE.Time	The IT environment shall provide access to accurate current time with required precision, translated to GMT.

5 IT Security Requirements

This section defines the TOE security functional requirements and assurance requirements. Requirements are drawn from the CC Parts 2 and 3 and have been written as required as Part 2 extended requirements. Selections and assignments made by the ST author in Part 2 and Part 2 extended requirements are enclosed in [square brackets] and text is in *italics*. Where refinements have been made in Part 2 requirements, the text is indicated by ***bold italics***. Iterations of requirements are indicated by a semicolon and number following the requirement number, e.g., FIA_UAU.1.1;1. In addition, the iterated requirement titles are indicated using a colon, e.g., FIA_UAU.1:1. Application Notes are denoted by "*Application Note:*" and the text following is in *italics*.

The TOE is Part 2 extended. The definition of Part 2 extended is found in the CC Part 1, section 5.1.3, "Part 2 extended - A PP or TOE is Part 2 extended if the functional requirements include functional components not in Part 2." All functional requirements included in this ST are listed in Table 5.1, below. Part 2 extended requirements are explicitly identified as "Part 2 extended."

Table 5.1 – Part 2 or Part 2 Extended Requirements

Requirement	**Part 2 or Extended**
FDP_ACC.1	Part 2
FDP_ACF.1	Part 2
FIA_AFL.1	Part 2
FIA_ATD.1	Part 2
FIA_UAU.1	Part 2
FIA_UAU.7	Part 2
FIA_UID.1	Part 2
FMT_MSA.1	Part 2
FMT_MSA.3	Part 2
FMT_MTD.1	Part 2
FMT_SMF.1	Part 2
FMT_SMR.2	Part 2
FPT_RVM.1	Part 2
FPT_SEP.1	Part 2
FPT_STM.1	Part 2
FCS_CRM_FPS.1	Part 2 Extended
FDP_CPD.1	Part 2 Extended
FDP_DAU_CPV_CER.1	Part 2 Extended
FDP_DAU_CPV_CER.2	Part 2 Extended
FDP_DAU_CPV_CER.3	Part 2 Extended
FDP_DAU_CPV_CER.4	Part 2 Extended

Requirement	Part 2 or Extended
FDP_DAU_CPV_CER.5	Part 2 Extended
FDP_DAU_CPV_INI.1	Part 2 Extended
FDP_DAU_CPV_INI.2	Part 2 Extended
FDP_DAU_CPV_INI.3	Part 2 Extended
FDP_DAU_CPV_INI.4	Part 2 Extended
FDP_DAU_CPV_OUT.1	Part 2 Extended
FDP_DAU_CPV_OUT.2	Part 2 Extended
FDP_DAU_CPV_OUT.3	Part 2 Extended
FDP_DAU_CRL.1	Part 2 Extended
FDP_DAU_ENC.1	Part 2 Extended
FDP_DAU_OCS.1	Part 2 Extended
FDP_DAU_SIG.1	Part 2 Extended
FDP_ETC_ENC.1	Part 2 Extended
FDP_ETC_SIG.1	Part 2 Extended
FDP_ITC_ENC.1	Part 2 Extended
FDP_ITC_PKI_INF.1	Part 2 Extended
FDP_ITC_SIG.1	Part 2 Extended

5.1 Security Functional Requirements for the TOE

The security functional requirements for the TOE are listed in Table 5.2 and the complete text of the requirements is provided below.

Table 5.2 – Security Functional Requirements for the TOE

#	Functional Requirement	Title
1	FDP_CPD.1	Certification path development
2	FDP_DAU_CPV_INI.1	Certification path initialisation -- basic
3	FDP_DAU_CPV_CER.1	Certificate processing -- basic
4	FDP_DAU_CPV_CER.2	Intermediate certificate processing -- basic
5	FDP_DAU_CPV_OUT.1	Certification path output -- basic
6	FDP_DAU_CPV_INI.2	Certification path initialisation – basic policy
7	FDP_DAU_CPV_OUT.2	Certification path output – basic policy
8	FDP_DAU_CPV_INI.3	Certification path initialisation – policy mapping
9	FDP_DAU_CPV_CER.3	Intermediate certificate processing – policy mapping
10	FDP_DAU_CPV_OUT.3	Certification path output – policy mapping
11	FDP_DAU_CPV_INI.4	Certification path initialisation – names

#	Functional Requirement	Title
12	FDP_DAU_CPV_CER.4	Certificate processing – name constraints
13	FDP_DAU_CPV_CER.5	Intermediate Certificate processing – name constraints
14	FDP_ETC_SIG.1	Export of PKI Signature
15	FDP_ITC_SIG.1	Import of PKI Signature
16	FDP_DAU_SIG.1	Signature Blob Verification
17	FDP_ETC_ENC.1	Export of PKI Encryption – Key Transfer Algorithms
18	FDP_DAU_ENC.1	PKI Encryption Verification – Key Transfer
19	FDP_ITC_ENC.1	Import of PKI Encryption – Key Transfer Algorithms
20	FDP_DAU_OCS.1	Basic OCSP Client
21	FDP_DAU_CRL.1	Basic CRL Checking

5.1.1 Class FDP – User Data Protection

FDP_CPD.1 Certification path development

Hierarchical to: No other components.

FDP_CPD.1.1 The TSF shall develop a certification path from a trust anchor provided by [*user*] to the subscriber using matching rules for the following subscriber certificate fields or extensions: [*distinguished name, subject alternative names, subject key identifier, subject public key algorithm, certificate policies*]*.*

FDP_CPD.1.2 The TSF shall develop the certification path using the following additional matching rule: [*none*].

FDP_CPD.1.3 The TSF shall develop the certification path using the following additional matching rule: [*none*]*.*

FDP_CPD.1.4 The TSF shall bypass any matching rules except [*distinguished name*] if additional certification paths are required.

Dependencies: None

FDP_DAU_CPV_INI.1 Certification path initialisation -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_INI.1.1 The TSF shall use the trust anchor provided by [*user*].

FDP_DAU_CPV_INI.1.2 The TSF shall obtain the time of interest called “TOI” from a reliable source: [*local environment,* [*or application*]].

FDP_DAU_CPV_INI.1.3 The TSF shall perform the following checks on the trust anchor [*None*].

FDP_DAU_CPV_INI.1.4 The TSF shall derive from the trust anchor [*subject DN, subject public key, subject public key algorithm object identifier, subject public key parameters*].

Dependencies: FCS_COP.1, FPT_STM.1

FDP_DAU_CPV_CER.1 Certificate processing -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_CER.1.1 The TSF shall reject a certificate only if the following checks fails:

a) Use parent-public-key, parent-public-key-algorithm-identifier, and parent-public-key-parameters to verify the signature on the certificate

b) notBefore field in the certificate < = TOI

c) notAfter field in the certificate > = TOI

d) issuer field in the certificate = parent-DN; or

e) TSF is able to process all extensions marked critical

FDP_DAU_CPV_CER.1.2 The TSF shall bypass the revocation status check if the certificate contains no-check extension.

FDP_DAU_CPV_CER.1.3 The TSF shall bypass the revocation check if the [*user*] overrides revocation checking.

FDP_DAU_CPV_CER.1.4 The TSF shall accept a certificate if the revocation status using [*CRL, **or** OCSP **response***] demonstrates that the certificate is not revoked.

FDP_DAU_CPV_CER.1.5 The TSF shall update the public key parameters state machine using the following rules:

a) Obtain the parameters from the subjectPublickeyInfo field of certificate if the parameters are present in the field; else

b) Retain the old parameters state if the subject public key algorithm of current certificate and parent public key algorithm of current certificate belong to the same family of algorithms, else

c) Set parameters = “null”.

Dependencies: FCS_COP.1, FPT_STM.1

Application Note: *Note that for FDP_DAU_CPV_CER.1.2, bypassing the revocation status check only occurs as part of OCSP processing, i.e. nocheck extensions are not processed in any other context.*

FDP_DAU_CPV_CER.2 Intermediate certificate processing -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_CER.2.1 The TSF shall accept an intermediate certificate only if the following additional checks succeed:

a) basicConstraints field is present with cA = TRUE

b) pathLenConstraint is not violated

c) if a critical keyUsage extension is present, keyCertSign bit is set

Dependencies: FDP_DAU_CPV_CER.1

FDP_DAU_CPV_OUT.1 Certification path output -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_OUT.1.1 The TSF shall output certification path validation failure if any certificate in the certification path is rejected.

FDP_DAU_CPV_OUT.1.2 The TSF shall output the following variables from the end certificate: subject DN, subject public key algorithm identifier, subject public key, critical keyUsage extension.

FDP_DAU_CPV_OUT.1.3 The TSF shall output the following additional variables from the end certificate [*certificate, subject alternative names, extendedKeyUsage*].

FDP_DAU_CPV_OUT.1.4 The TSF shall output the subject public key parameters from the certification path parameter state machine.

Dependencies: None

FDP_DAU_CPV_INI.2 Certification path initialisation – basic policy

Hierarchical to: No other components.

FDP_DAU_CPV_INI.2.1 The TSF shall use the initial-certificate-policies provided by [*user*].

Dependencies: FDP_DAU_CPV_INI.1

FDP_DAU_CPV_OUT.2 Certification path output – basic policy

Hierarchical to: No other components.

FDP_DAU_CPV_OUT.2.1 The TSF shall output the certificate policies using the following rule: intersection of certificatePolicies extensions in all the certificates in certification path and initial-certificate-policies.

Dependencies: FDP_DAU_CPV_OUT.1

FDP_DAU_CPV_INI.3 Certification path initialisation – policy mapping

Hierarchical to: No other components.

FDP_DAU_CPV_INI.3.1 The TSF shall use the explicit-policy-indicator, policy-mapping-inhibit-indicator, inhibit-any-policy-indicator provided by [*user*].

Dependencies: FDP_DAU_CPV_INI.2

FDP_DAU_CPV_CER.3 Intermediate certificate processing – policy mapping

Hierarchical to: No other components.

FDP_DAU_CPV_CER.3.1 The TSF shall use the intermediate certificate to update the following state variables:

a) explicit-policy-indicator

b) policy-mapping-inhibit-indicator

c) inhibit-any-policy-indicator

Dependencies: FDP_DAU_CPV_CER.2

FDP_DAU_CPV_OUT.3 Certification path output – policy mapping

Hierarchical to: No other components.

FDP_DAU_CPV_OUT.3.1 The TSF shall map policies in the calculation of the policies intersection if and only if policy-mapping-inhibit-indicator is not set.

FDP_DAU_CPV_OUT.3.2 During the calculation of the policy intersection, the TSF shall match any-policy to all policies if and only if inhibit-any-policy-indicator is not set.

FDP_DAU_CPV_OUT.3.3 The TSF shall output certification path failure if the intersection of certificatePolicies (as modified by policy mapping and inhibit-any-policy) is null and explicit-policy-indicator is set.

FDP_DAU_CPV_OUT.3.4 The TSF shall output certification path failure if the intersection of certificatePolicies (as modified by policy mapping and inhibit-any-policy) and initial-certificate-policies is null and explicit-policy-indicator is set.

FDP_DAU_CPV_OUT.3.5 The TSF shall output policy mapping history.

FDP_DAU_CPV_OUT.3.6 The TSF shall output policy qualifiers applicable to output policies.

Dependencies: FDP_DAU_CPV_OUT.2

FDP_DAU_CPV_INI.4 Certification path initialisation – names

Hierarchical to: No other components.

FDP_DAU_CPV_INI.4.1 The TSF shall initialize the following: permitted-subtrees = ∞, excluded-subtrees = ∅

Dependencies: FDP_DAU_CPV_INI.1

FDP_DAU_CPV_CER.4 Certificate processing – name constraints

Hierarchical to: No other components.

FDP_DAU_CPV_CER.4.1 The TSF shall accept a certificate only if the following additional conditions are satisfied:

a) subject DN is in at least one of the permitted-subtrees for DN

b) subject DN is in none of the excluded-subtrees for DN

c) each hierarchical name form of type [*DN*] in the subjectAlternateName field is in at least one of the permitted-subtrees for that name form

d) each hierarchical name form of type [*DN*] in the subjectAlternateName field is in none of the excluded-subtrees for that name form

Dependencies: FDP_DAU_CPV_CER.1

FDP_DAU_CPV_CER.5 Intermediate Certificate processing – name constraints

Hierarchical to: No other components.

FDP_DAU_CPV_CER.5.1 The TSF shall use the intermediate certificate to update the following states:

a) permitted-subtrees

b) excluded-subtrees

Dependencies: FDP_DAU_CPV_CER.2

FDP_ETC_SIG.1 Export of PKI Signature

Hierarchical to: No other component

FDP_ETC_SIG.1.1 The TSF shall invoke the cryptographic module with the appropriate private key to generate a digital signature.

FDP_ETC_SIG.1.2 The TSF shall include the following information with the digital signature [*hashing algorithm, signature algorithm*].

Dependencies: FCS_CRM_FPS.1

FDP_ITC_SIG.1 Import of PKI Signature

Hierarchical to no other component

FDP_ITC_SIG.1.1 The TSF shall use the following information from the signed data: [*hashing algorithm, signature algorithm*] during signature verification.

Dependencies: None

FDP_DAU_SIG.1 Signature Blob Verification

Hierarchical to: No other components.

FDP_DAU_SIG.1.1 The TSF shall invoke the cryptographic module with the following information from Certification Path Validation to verify digital signature on signed data: subject public key algorithm, subject public key, subject public key parameters.

FDP_DAU_SIG.1.2 The TSF shall verify that the keyUsage extension output from the Certification Path Validation has the [*nonRepudiation or digitalSignature*] bit set.

FDP_DAU_SIG.1.3 The TSF shall apply the following additional checks: [*none*].

Dependencies: FCS_CRM_FPS.1, FDP_DAU_CPV_OUT.1

FDP_ETC_ENC.1 Export of PKI Encryption – Key Transfer Algorithms

Hierarchical to: No other component

FDP_ETC_ENC.1.1 The TSF shall include the following information with the encrypted data: [*key encryption algorithm, data encryption algorithm, decryption key identifier*].

FDP_ETC_ENC.1.2 The TSF shall invoke the cryptographic module with the following information from Certification Path Validation to create encrypted data:

subject public key algorithm, subject public key, subject public key parameters.

Dependencies: FCS_CRM_FPS.1, FDP_DAU_CPV_OUT.1

FDP_DAU_ENC.1 PKI Encryption Verification – Key Transfer

Hierarchical to: No other components.

FDP_DAU_ENC.1.1 The TSF shall verify that the keyUsage output from Certification Path Validation contains keyEncipherment bit set.

FDP_DAU_ENC.1.2 The TSF shall apply the following additional checks: [[*none*]].

Dependencies: FDP_DAU_CPV_OUT.1

Application Note: *This component is used to verify that the correct public key is used during encryption.*

FDP_ITC_ENC.1 Import of PKI Encryption – Key Transfer Algorithms

Hierarchical to: No other components

FDP_ITC_ENC.1.1 The TSF shall invoke the cryptographic module with the following information from the encrypted data: [*key encryption algorithm, data encryption algorithm, decryption key identifier*] to perform decryption.

Dependencies: FCS_CRM_FPS.1

FDP_DAU_OCS.1 Basic OCSP Client

Hierarchical to: No other component

FDP_DAU_OCS.1.1 The TSF shall formulate the OCSP requests in accordance with PKIX RFC 2560.

FDP_DAU_OCS.1.2 The OCSP request shall contain the following extensions: [*nonce*].

FDP_DAU_OCS.1.3 The TSF shall obtain the public key, algorithm, and public key parameters of the OCSP Responder from [*OCSP responder certificate*].

FDP_DAU_OCS.1.4 The TSF shall perform the following additional function:

[*establish trust in OCSP responder certificate using* [*certification path validation – basic, certification path validation – basic policy, certification path validation –policy mapping, certification path validation – name constraint*]].

FDP_DAU_OCS.1.5 The TSF shall invoke the cryptographic module to verify signature on the OCSP response using the trusted public key, algorithm, and public key parameters of the OCSP responder.

FDP_DAU_OCS.1.6 The TSF shall verify that if the OCSP responder certificate contains extendedKeyUsage extension, the extension contains the PKIX OID for ocsp-signing or the anyExtendedKeyUsage OID.

FDP_DAU_OCS.1.7 The TSF shall match the responderID in the OCSP response with the corresponding information in the responder certificate.

FDP_DAU_OCS.1.8 The TSF shall match the certID in a request with certID in singleResponse.

FDP_DAU_OCS.1.9 The TSF shall reject the OCSP response for an entry if all of the following are true:

a) time checks are not overridden;

b) TOI > thisUpdate + x where x is provided by [*user*]; and

c) TOI > nextUpdate for entry + x where x=0 and is provided by [*no one*].

FDP_DAU_OCS.1.10 The TSF shall permit [*user*] to override time checks.

FDP_DAU_OCS.1.11 The TSF shall reject OCSP response if the response contains “critical” extension(s) that TSF does not process.

FDP_DAU_OCS.1.12 The TSF shall perform the following additional:

a) request nonce = response nonce,

Dependencies: FCS_CRM_FPS.1, FPT_STM.1

FDP_DAU_CRL.1 Basic CRL Checking

Hierarchical to no other component

FDP_DAU_CRL.1.1 The TSF shall obtain the CRL from: [*local cache, repository, location point to by the CRL DP in public key certificate of interest, user*].

FDP_DAU_CRL.1.2 The TSF shall obtain the trusted public key, algorithm, and public key parameters of the CRL issuer.

FDP_DAU_CRL.1.3 The TSF shall invoke the cryptographic module to verify signature on the CRL using trusted public key, algorithm, and public key parameters of the CRL issuer.

FDP_DAU_CRL.1.4 The TSF shall verify that if a critical keyUsage extension is present in CRL issuer certificate, cRLSign bit in the extension is set in the certificate.

FDP_DAU_CRL.1.5 The TSF shall match the issuer field in the CRL with what it assumes to be the CRL issuer.

FDP_DAU_CRL.1.6 The TSF shall reject the CRL if all of the following are true:

a) Time check are not overridden;

b) TOI > thisUpdate + x where x is provided by [*user*]; and

c) TOI > nextUpdate + x where x = 0 and is provided by [*no one*].

FDP_DAU_CRL.1.7 The TSF shall permit [*user*] to override time checks.

FDP_DAU_CRL.1.8 The TSF shall reject CRL if the CRL contains “critical” extension(s) that TSF does not process.

FDP_DAU_CRL.1.9 The TSF shall perform the following additional checks: [*none*].

Dependencies: FCS_COP.1, FPT_STM.1

Application Note: *The trusted public key, algorithm, and public key parameters of the CRL issuer should normally be the same as those used for verifying signature on the certificate being checked for revocation. If not, at least certificate path validation – basic can be used to obtain the public key.*

5.2 Security Functional Requirements for the IT Environment

A list of the security functional requirements for the IT Environment are provided in Table 5.3. The full text of the security functional requirements is contained below. The security functional requirements for the IT environment specify the ability to manage multiple private keys, associated certificates, and identifying data and associations among them. The term “manage” means the ability to do one or more of the following: generate, destroy, delete, use, import, export, modify, etc. The identifying data and association between private key and public key certificates are useful in selecting the appropriate cryptographic keys for cryptographic operations and for CMS type information generation. The security requirements for the IT Environment also provide for the maintenance of secure storage of trust anchors.

Table 5.3 – Security Functional Requirements for the IT Environment

#	Functional Requirement	Title
1E	FCS_CRM_FPS.1	FIPS compliant cryptographic module
2E	FDP_ACC.1	Subset Access Control – PKI Credential Management
3E	FDP_ACF.1	Security attribute based access control – PKI Credential Management
4E	FDP_ITC_PKI_INF.1	Import of PKI information from outside the TSF
5E	FIA_AFL.1	Authentication failure handling
6E	FIA_ATD.1	User attribute definition
7E	FIA_UAU.1	Timing of authentication
8E	FIA_UAU.7	Protected authentication feedback
9E	FIA_UID.1	Timing of identification
10E	FMT_MSA.1	Management of security attributes
11E	FMT_MSA.3	Static attribute initialisation
12E	FMT_MTD.1	Management of TSF data
13E	FMT_SMF.1	Specification of management functions
14E	FMT_SMR.2	Restrictions on security roles
15E	FPT_RVM.1	Non-bypassability of the TSP
16E	FPT_SEP.1	TSF domain separation
17E	FPT_STM.1	Reliable time stamps

5.2.1 Class FCS – Cryptographic Support

FCS_CRM_FPS.1 FIPS compliant cryptographic module

Hierarchical to: No other components.

FCS_CRM_FPS.1.1 The ***IT Environment*** shall provide all cryptographic modules necessary for the TSF.

FCS_CRM_FPS.1.2 Each cryptographic module shall be FIPS 140 series Level 1 validated.

Dependencies: None.

5.2.2 Class FDP – User Data Protection

FDP_ACC.1 Subset access control – PKI Credential Management

Hierarchical to: No other components.

FDP_ACC.1.1 The ***IT Environment*** shall enforce the [*PKI credential management SFP*] on [*management of trust anchors and private key material*].

Application Note: *The terms object and subject refer to generic elements in the TOE. For a policy to be implemented, these entities must be clearly identified. For most systems there is only one type of subject, usually called a process or task, which needs to be specified in the ST. The ST author should specify the list of subjects, objects, and operations among subjects and objects covered by the SFP.*

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACF.1 Security attribute based access control – PKI Credential Management

Hierarchical to: No other components.

FDP_ACF.1.1 The ***IT Environment*** shall enforce the [*PKI credential management SFP*] to objects based on the *identity of the subject and the set of roles that the subject is authorized to assume*.

FDP_ACF.1.2 The ***IT Environment*** shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

a) ***Private keys may be generated, imported, exported, destroyed, used by*** **[[*user*]].**

b) ***Public key certificates may be imported, exported, deleted by*** **[[*user*]].**

c) ***Public key certificates may be used by anyone.***

d) **[*Trust anchors may be imported, exported, or deleted by user*].]**

FDP_ACF.1.3 The ***IT Environment*** shall explicitly authorize access of subjects to objects based on the following additional rules: [*users shall have* read *and read and execute privileges on PKIF.dll*].

FDP_ACF.1.4 The ***IT Environment*** shall explicitly deny access of subjects to objects based on the [*users shall have read and read and execute privileges for PKIF.dll and shall not be able delete or modify to the file*].

Dependencies: FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation

FDP_ITC_PKI_INF.1 Import of PKI information from outside the TSF

Hierarchical to: No other components.

FDP_ITC_PKI_INF.1.1 The ***IT Environment*** shall ensure the availability of [*certificates, CRLs, OCSP responses,* [*trust anchors*]], for the time of interest TOI, to the TOE [100%] given the following conditions [*availability of network connection, availability of information server, availability of information in the application protocol, availability of information to the IT environment*].

Dependencies None

5.2.3 Class FIA – Identification and Authentication

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components

FIA_AFL.1.1 The ***IT Environment*** shall detect when [*5*] unsuccessful authentication attempts occur related to [*Windows login*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the ***IT Environment*** shall [*lockout the account for 1 hour*].

Dependencies: FIA_UAU.1 Timing of authentication

FIA_ATD.1 User attribute definition

Hierarchical to: No other components

FIA_ATD.1.1 The ***IT Environment*** shall maintain the following list of security attributes belonging to individual users: [*role*].

Dependencies: None

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components

FIA_UAU.1.1 The ***IT Environment*** shall allow [*shutdown*] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The ***IT Environment*** 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

FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components

FIA_UAU.7.1 The ***IT Environment*** shall provide only [*dialog box*] to the user while the authentication is in progress.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UID.1 Timing of identification

Hierarchical to: No other components

FIA_UID.1.1 The ***IT Environment*** shall allow [shutdown] on behalf of the user to be performed before the user is identified.

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

Dependencies: None.

Application Note: Identification and authentication rules may vary between TOEs; those rules need to be specified in the ST.

5.2.4 Class FMT – Security Management

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components

FMT_MSA.1.1 The ***IT Environment*** shall enforce the [*PKI credential management SFP*] to restrict the ability to [*query, modify, delete,* [*import*]] the security attributes **[*user role, key identifier, association between private key and public key certificate*] to [*user*]**.

Dependencies: FMT_SMF.1 Specification of Management Functions, FMT_SMR.1 Security roles, FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components

FMT_MSA.3.1 The ***IT Environment*** shall enforce the [*PKI credential management SFP*] to provide [[*specific*]] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The ***IT Environment*** shall allow the **[*user*]** to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_SMR.1 Security roles, FMT_MSA.1 Management of security attributes

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components

FMT_MTD.1.1 The ***IT Environment*** shall restrict the ability to [*change_default, modify, delete, clear, **import, add***] the **[*trust anchors, identification data, authentication data, number of unsuccessful authentication attempts*]** to **[*user*]**.

Dependencies: FMT_SMF.1 Specification of Management Functions, FMT_SMR.1 Security roles

Application Note: The ST author may iterate the requirement as necessary. The ST author must select identification data and authentication data in order to meet the security objective OE.Protect_I&A_Data. The ST author must select trust anchors in order to meet the security objective OE.Trust_Anchor.

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components

FMT_SMF.1.1 The ***IT Environment*** shall be capable of performing the following security management functions: [*none*].

Dependencies: None

FMT_SMR.2 Restrictions on security roles

Hierarchical to: FMT_SMR.1

FMT_SMR.2.1 The ***IT Environment*** shall maintain the roles: **[*user*]**.

FMT_SMR.2.2 The ***IT Environment*** shall be able to associate users with roles.

FMT_SMR.2.3 The ***IT Environment*** shall ensure that the conditions [*none*] are satisfied.

Dependencies: FIA_UID.1 Timing of identification

5.2.5 Class FPT – Protection of the TOE Security Functions

FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components

FPT_RVM.1.1 The ***IT Environment*** shall ensure that TSP enforcement functions are invoked and succeed before each function within the ***TSF Scope of Control (TSC)*** is allowed to proceed.

Dependencies: None.

FPT_SEP.1 TSF domain separation

Hierarchical to: No other components

FPT_SEP.1.1 The ***IT Environment*** shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The ***IT Environment*** shall enforce separation between the security domains of subjects in the TSC.

Dependencies: None.

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

FPT_STM.1.1 The ***IT Environment*** shall be able to provide reliable time stamps for TSF use.

Dependencies: None.

5.2.6 Strength of Function Requirement

The TOE performs no authentication. Since the TOE does not include probabilistic or permutational mechanisms, the SOF claim is not applicable.

5.3 Assurance Requirements

The TOE Evaluation Assurance Level is EAL4 augmented by ALC_FLR.1. All requirements are drawn from Part 3 of the Common Criteria. The assurance components are listed in Table 5.4. EAL4 provides added assurance from positive security engineering based on good commercial development practices, which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest assurance level at which it is likely to be economically feasible to retrofit to an existing product line. ALC_FLR.1 is added to provide basic flaw remediation.

Table 5.4 – EAL4 with Augmentation Assurance Requirements

Assurance Component ID	Assurance Component Title
ACM_AUT.1	Partial CM automation
ACM_CAP.4	Generation support and acceptance procedures
ACM_SCP.2	Problem tracking CM coverage
ADO_DEL.2	Detection of modification
ADO_IGS.1	Installation, generation, and start-up procedures
ADV_FSP.2	Fully defined external interfaces
ADV_HLD.2	Security enforcing high-level design
ADV_IMP.1	Subset of the Implementation of the TSF
ADV_LLD.1	Descriptive low-level design
ADV_RCR.1	Informal correspondence demonstration
ADV_SPM.1	Informal TOE security policy model
AGD_ADM.1	Administrator guidance
AGD_USR.1	User guidance
ALC_DVS.1	Identification of security measures
ALC_FLR.1	Basic flaw remediation
ALC_LCD.1	Developer defined life-cycle model
ALC_TAT.1	Well-defined development tools
ATE_COV.2	Analysis of coverage
ATE_DPT.1	Testing: high-level design
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing – sample
AVA_MSU.2	Validation of analysis
AVA_SOF.1	Strength of TOE security function evaluation
AVA_VLA.2	Independent vulnerability analysis

6 TOE Summary Specification

PKIF is a C++ software library designed to simplify the task of adding PKI support to applications. It performs PKI-related functions, including the following:

- Certification Path Processing
- CMS based Signature Generation
- CMS based Signature Verification using PKI
- PKI Encryption using Key Transfer Algorithms functionality
- PKI Decryption using Key Transfer Algorithms functionality
- Online Certificate Status Protocol Client functionality
- Certificate revocation list processing functionality

The interface to PKIF permits applications to perform a variety of tasks in addition to and in support of the functions listed above. The following sections describe the PKIF functions and the TSF interface of the library.

PKIF uses FIPS-compliant cryptographic modules from the IT environment to perform encryption, decryption, and hashing operations.

6.1 Certification Path Processing, CRL Processing and OCSP Processing

PKIF performs X.509 certification path processing, including certification path development and certification path validation. Certification path validation consists of validating certificates starting with the one certified by a trust anchor and ending with the one issued to the subscriber of interest. PKIF supports X.509 version 3 Certificates and X.509 CRLs, versions 1 and 2. All processing is X.509 and PKIX RFC3280 compliant.

There are three types of public key certificates involved in certificate path validation:

- Trust anchor (TA) certificates: These are certificates containing public keys that do not require any validation. Trust anchors generally take the form of a self-signed certificate. TAs must be delivered to entities that rely on the TA's public key using trusted means. The primary purpose of the trust anchor is to provide a means of conveying a Distinguished Name (DN), public key, algorithm identifier, and the public key parameters (if applicable) for use in validating certification paths.
- Intermediate certificates: These are the certificates issued to CAs. All certificates in a certification path are intermediate certificates, except the trust anchor certificate and end entity certificate.
- End certificates: This is the last certificate in the certification path and is issued to the subscriber of interest. A subscriber certificate is also called an end-entity certificate (i.e., a certificate issued to an entity not functioning as a CA). Sometimes, the last certificate can be a CA certificate, e.g., when the certification path is used to verify signature on a CRL.

PKIF processes the following security-related certificate extensions: ocsp-nocheck, keyUsage, extendedKeyUsage, and basicConstraints. PKIF performs the processing of the following certificate policy-related extensions: certificatePolicies, policyMapping, inhibitAnyPolicy, policyConstraints, and nameConstraints extensions

PKIF can generate Online Certificate Status Protocol (OCSP) requests and validate OCSP responses to determine the revocation status of public key certificates. PKIF verifies OCSP Responder as a trust anchor or as an end entity authorized to sign OCSP responses. PKIF establishes trust in the OCSP responder certificates by performing Certification Path Validation.

PKIF provides Certificate Revocation List (CRL) validation functionality that enables applications to determine the revocation status of a certificate using a CRL. PKIF may be used to process CRLs obtained from a variety of sources including: locations indicated by a CRL Distribution Point (CRLDP) extension in a certificate, local storage facilities or LDAP-accessible directories.

PKIF permits the use of the same public key for CRL signature verification as the one used for verifying the signature on the certificate, but does not mandate it. In other words, a PKIF will develop and validate certification paths to CRL signers where necessary.

Table 6.1 Primary Path Processing and Revocation Status-related Interfaces

Interface	Function
`bool CPKIFPathProcessingMediator2::BuildAndValidatePath (CPKIFCertificatePathPtr &, CPKIFPathValidationResultsPtr &)`	Performs path development and path validation, including revocation status determination
`bool CPKIFPathProcessingMediator2::BuildPath (CPKIFCertificatePath &)`	Performs path development
`bool CPKIFPathProcessingMediator2::ValidatePath (CPKIFCertificatePath &, CPKIFPathValidationResults &, CPKIFFuncStoragePtr &)`	Performs path validation, including revocation status determination

This function implements the following SFRs:

- FDP_CPD.1
- FDP_DAU_CPV_INI.1
- FDP_DAU_CPV_CER.1
- FDP_DAU_CPV_CER.2
- FDP_DAU_CPV_OUT.1
- FDP_DAU_CPV_INI.2
- FDP_DAU_CPV_OUT.2
- FDP_DAU_CPV_INI.3
- FDP_DAU_CPV_CER.3

- FDP_DAU_CPV_OUT.3
- FDP_DAU_CPV_INI.4
- FDP_DAU_CPV_CER.4
- FDP_DAU_CPV_CER.5
- FDP_DAU_OCS.1
- FDP_DAU_CRL.1

6.2 Signature Generation Functionality

PKIF enables application to use a private key for signature generation and to specify information covered by that signature, e.g. using the CMS SignedData format. The CMS structures implemented by PKIF are based on those defined in [RFC3369]. Several CMS related samples are provided in the PKIF User's Guide Section 6.4 under: *Creating signed messages*, *Verifying signed messages*, *Creating encrypted messages* and *Decrypting encrypted messages*.

Table 6.2 Primary Signature Generation-related Interfaces

Interface	Function
`CPKIFBufferPtr CPKIFSignedData::Encode (void)`	Generates a SignedData message, including generation of signatures for the specified signers

This function implements the following SFRs:

- FDP_ETC_SIG.1

6.3 PKI Signature Verification Functionality

PKIF enables application to process signature information, e.g. using the CMS SignedData format, and to verify signatures using a public key. The CMS structures implemented by PKIF are based on those defined in [RFC3369]. Several CMS related samples are provided in the PKIF User's Guide Section 6.4 under: *Creating signed messages*, *Verifying signed messages*, *Creating encrypted messages* and *Decrypting encrypted messages*.

Table 6.3 Primary Signature Verification-related Interfaces

Interface	Function
`void CPKIFSignedData::Decode(CPKIFBufferPtr &)`	This function is used to decode a binary, encoded SignedData message

`bool CPKIFSignedData::Verify`
`(int,`
`enum CMSVerificationStatus &,`
`class boost::shared_ptr<class CPKIFCertificate> &,`
	Verifies a signature contained in a SignedData message including validation of the signer's certificate

`enum CMSPathValidationStatus)`	

`bool CPKIFSignedData::Verify`
`(int,`
`enum CMSVerificationStatus &,`
`enum CMSPathValidationStatus)`
	Verifies a signature contained in a SignedData message including validation of the signer's certificate

This function implements the following SFRs:

- FDP_ITC_SIG.1
- FDP_DAU_SIG.1

6.4 PKI Encryption using Key Transfer Algorithms Functionality

PKIF enables application to perform public key encryption using key transfer algorithms such as RSA. The CMS structures implemented by PKIF are based on those defined in [RFC3369]. Several CMS related samples are provided in the PKIF User's Guide Section 6.4 under: *Creating signed messages*, *Verifying signed messages*, *Creating encrypted messages* and *Decrypting encrypted messages*.

Table 6.4 Primary PKI Encryption-related Interfaces

Interface	Function

`void CPKIFEnvelopedData::AddRecipient`
`(CPKIFCertificatePtr &,`
`CPKIFCertificatePathPtr &,`
`CPKIFPathValidationResultsPtr &,`
`enum CMSPathValidationStatus)`
	Adds a recipient to an EnvelopedData message after verifying the recipient's certificate

`void CPKIFEnvelopedData::AddRecipient`
`(CPKIFCertificatePtr &,`
`enum CMSPathValidationStatus)`
	Adds a recipient to an EnvelopedData message after verifying the recipient's certificate

`CPKIFBufferPtr CPKIFEnvelopedData::Encode`
`(void)`
	Generates the encoded EnvelopedData message including generation of a content encryption key and encryption of the content encryption for each recipient

This function implements the following SFRs:

- FDP_ETC_ENC.1
- FDP_DAU_ENC.1

6.5 PKI Decryption using Key Transfer Algorithms Functionality

PKIF enables applications to perform private key decryption using key transfer algorithms such as RSA. The CMS structures implemented by PKIF are based on those defined in [RFC3369]. Several CMS related samples are provided in the PKIF User's Guide Section 6.4 under: *Creating signed messages*, *Verifying signed messages*, *Creating encrypted messages* and *Decrypting encrypted messages*.

Table 6.5 Primary PKI Decryption using Key Transfer Algorithms-related Interfaces

Interface	Function
void CPKIFEnvelopedData::Decode (CPKIFBufferPtr &)	Decodes an EnvelopedData message
CPKIFBufferPtr CPKIFEnvelopedData::Decrypt (CPKIFCredentialPtr &)	Decrypts an EnvelopedData message

This function implements the following SFRs:

- FDP_ITC_ENC.1

6.6 Supporting Functionality

The interfaces identified in the sections 6.1-6.5 require support from a number of objects to prepare for and review the results from the various operations. The following list describes the entire TSFI for the library, including the interfaces cited above:

Certificate and CRL Storage and Retrieval

1) void CPKIFCacheMediator2::AddColleague(IPKIFColleague* module, bool transferOwnership = true, void (*deleteFunc)(void *) = NULL)
2) CPKIFLDAPRepository::CPKIFLDAPRepository(void)
3) void CPKIFLDAPRepository::Set_Port(int)
4) void CPKIFLDAPRepository::SetHost(char const *)

Cryptography

5) char const * CPKIFCredential::ID(void)
6) char const * CPKIFCredential::Name(void)
7) void CPKIFCryptoMediator2::GetKeyList(CPKIFCredentialList &, std::bitset<9> *)

Cryptographic Message Syntax

8) CPKIFEncapsulatedContentInfo::CPKIFEncapsulatedContentInfo(void)
9) CPKIFBufferPtr CPKIFEncapsulatedContentInfo::GetContent(void)
10) CPKIFOIDPtr CPKIFEncapsulatedContentInfo::GetOID(void)
11) void CPKIFEncapsulatedContentInfo::SetContent(CPKIFBufferPtr &)
12) void CPKIFEncapsulatedContentInfo::SetOID(CPKIFOIDPtr &)
13) CPKIFEncryptedContentInfo::CPKIFEncryptedContentInfo(void)
14) CPKIFAlgorithmIdentifierPtr CPKIFEncryptedContentInfo::GetAlgorithmIdentifier(void)
15) CPKIFBufferPtr CPKIFEncryptedContentInfo::GetContent(void)
16) CPKIFOIDPtr CPKIFEncryptedContentInfo::GetOID(void)
17) void CPKIFEncryptedContentInfo::SetAlgorithmIdentifier(CPKIFAlgorithmIdentifierPtr &)
18) void CPKIFEncryptedContentInfo::SetContent(CPKIFBufferPtr &)
19) void CPKIFEncryptedContentInfo::SetOID(CPKIFOIDPtr &)
20) void CPKIFEnvelopedData::AddRecipient(CPKIFCertificatePtr &, CPKIFCertificatePathPtr &, CPKIFPathValidationResultsPtr &,enum CMSPathValidationStatus)
21) void CPKIFEnvelopedData::AddRecipient(CPKIFCertificatePtr &, enum CMSPathValidationStatus)
22) CPKIFEnvelopedData::CPKIFEnvelopedData(void)

23) void CPKIFEnvelopedData::Decode(CPKIFBufferPtr &)

24) CPKIFBufferPtr CPKIFEnvelopedData::Decrypt(CPKIFCredentialPtr &)

25) CPKIFBufferPtr CPKIFEnvelopedData::Encode(void)

26) void CPKIFEnvelopedData::SetDataToEncrypt(CPKIFEncryptedContentInfoPtr &)

27) void CPKIFEnvelopedData::SetPathSettings(CPKIFPathSettingsPtr &)

28) void CPKIFSignedData::AddSignerInfo(CPKIFSignerInfoPtr &)

29) CPKIFSignedData::CPKIFSignedData(void)

30) void CPKIFSignedData::Decode(CPKIFBufferPtr &)

31) CPKIFBufferPtr CPKIFSignedData::Encode(void)

32) CPKIFEncapsulatedContentInfoPtr CPKIFSignedData::GetEncapsulatedContent(void)

33) void CPKIFSignedData::SetEncapsulatedContent(CPKIFEncapsulatedContentInfoPtr &)

34) void CPKIFSignedData::SetPathSettings(CPKIFPathSettingsPtr &)

35) bool CPKIFSignedData::Verify(int, enum CMSVerificationStatus &,class boost::shared_ptr<class CPKIFCertificate> &,enum CMSPathValidationStatus)

36) bool CPKIFSignedData::Verify(int,enum CMSVerificationStatus &,enum CMSPathValidationStatus)

37) CPKIFSignerInfo::CPKIFSignerInfo(void)

38) void CPKIFSignerInfo::SetCredential(CPKIFCredentialPtr &)

39) void keyUsageChecker_Encryption(const CPKIFCertificateNodeEntryPtr& certNode, CPKIFPathValidationResults& results, CertificateType type)

40) void keyUsageChecker_Signature(const CPKIFCertificateNodeEntryPtr& certNode, CPKIFPathValidationResults& results, CertificateType type)

Online Certificate Status Protocol

41) CPKIFOCSPChecker::CPKIFOCSPChecker(void)

42) void CPKIFOCSPChecker::Set_Port(int)

43) void CPKIFOCSPChecker::SetHost(char const *)

Path Processing

44) CPKIFCertificatePath::CPKIFCertificatePath(void)

45) void CPKIFCertificatePath::SetPathSettings(CPKIFPathSettingsPtr const &)

46) void CPKIFCertificatePath::SetTarget(CPKIFCertificatePtr const &)

47) CPKIFFuncStorage::CPKIFFuncStorage(void (*) (const CPKIFCertificateNodeEntryPtr&, CPKIFPathValidationResults&, CertificateType))

48) void CPKIFFuncStorage::addFunc(void (*) (const CPKIFCertificateNodeEntryPtr&, CPKIFPathValidationResults&, CertificateType))

49) bool CPKIFPathProcessingMediator2::BuildAndValidatePath(CPKIFCertificatePathPtr &, CPKIFPathValidationResultsPtr &)

50) bool CPKIFPathProcessingMediator2::BuildPath(CPKIFCertificatePath &)

51) bool CPKIFPathProcessingMediator2::ValidatePath(CPKIFCertificatePath &, CPKIFPathValidationResults &, CPKIFFuncStoragePtr &)

52) CPKIFPathSettings::CPKIFPathSettings(void)

53) void CPKIFPathSettings::SetCheckRevocationStatus(bool)

54) void CPKIFPathSettings::SetInitialExplicitPolicyIndicator(bool)

55) void CPKIFPathSettings::SetInitialInhibitAnyPolicyIndicator(bool)

56) void CPKIFPathSettings::SetInitialPolicyMappingInhibitIndicator(bool)

57) void CPKIFPathSettings::SetInitialPolicySet(CPKIFPolicyInformationListPtr &)

58) void CPKIFPathSettings::SetRequireFreshRevocationData(bool)

59) void CPKIFPathSettings::SetRequireSufficientlyRecent(bool)

60) void CPKIFPathSettings::SetSufficientlyRecent(int)

61) void CPKIFPathSettings::SetValidationTime(CPKIFTimePtr &)
62) CPKIFPathValidationResults::CPKIFPathValidationResults(void)
63) int CPKIFPathValidationResults::DiagnosticCode(void)
64) voidCPKIFPathValidationResults::GetAuthorityConstrainedSet(CPKIFPolicyInformationListPtr &)
65) std::vector<CPKIFPolicyInformationListPtr> const CPKIFPathValidationResults::GetAuthorityConstrainedSetTable(void)
66) bool CPKIFPathValidationResults::GetExplicitPolicyIndicator(void)
67) void CPKIFPathValidationResults::GetUserConstrainedSet(CPKIFPolicyInformationListPtr &)
68) CPKIFAlgorithmIdentifierPtr CPKIFPathValidationResults::GetWorkingParams(void)
69) bool CPKIFPathValidationResults::PathSuccessfullyValidated(void)

Utility

70) char const * CPKIFException::GetDescription(void)
71) int CPKIFException::GetErrorCode(void)
72) void FreeDefaultMediator(IPKIFMediator *)
73) IPKIFMediator * MakeDefaultMediator(bool, CPKIFOCSPChecker *)
74) template<class T> T* IPKIFMediator::GetMediator() const

X.509 ASN.1 Encoding/Decoding

75) CPKIFOIDPtr CPKIFAlgorithmIdentifier::oid(void)
76) bool CPKIFAlgorithmIdentifier::hasParameters() const
77) CPKIFBufferPtr CPKIFAlgorithmIdentifier::parameters(void)
78) CPKIFBuffer::CPKIFBuffer(unsigned char const *,unsigned int)
79) unsigned char const * CPKIFBuffer::GetBuffer(void)
80) unsigned int CPKIFBuffer::GetLength(void)
81) CPKIFCertificate::CPKIFCertificate(void)
82) void CPKIFCertificate::Decode(unsigned char const *,int)
83) CPKIFBufferPtr CPKIFCertificate::Encoded(void)
84) template <class T> shared_ptr<T> GetExtension()
85) CPKIFNamePtr CPKIFCertificate::Subject(void)
86) CPKIFSubjectPublicKeyInfoPtr CPKIFCertificate::SubjectPublicKeyInfo(void)
87) void CPKIFExtendedKeyUsage::KeyPurposeIDs(std::vector<CPKIFOIDPtr> &)
88) CPKIFNamePtr CPKIFGeneralName::directoryName(void)
89) char const * CPKIFGeneralName::dnsName(void)
90) enum CPKIFGeneralName::GENNAMETYPE CPKIFGeneralName::GetType(void)
91) CPKIFBufferPtr CPKIFGeneralName::ipAddress(void)
92) CPKIFBufferPtr CPKIFGeneralName::otherName(void)
93) char const * CPKIFGeneralName::rfc822Name(void)
94) char const * CPKIFGeneralName::uri(void)
95) CPKIFBufferPtr CPKIFGeneralName::x400Address(void)
96) bool CPKIFKeyUsage::CRLSign(void)
97) bool CPKIFKeyUsage::DataEncipherment(void)
98) bool CPKIFKeyUsage::DecipherOnly(void)
99) bool CPKIFKeyUsage::DigitalSignature(void)
100) bool CPKIFKeyUsage::EncipherOnly(void)
101) bool CPKIFKeyUsage::KeyAgreement(void)
102) bool CPKIFKeyUsage::KeyCertSign(void)

```
103) bool CPKIFKeyUsage::KeyEncipherment(void)
104) bool CPKIFKeyUsage::NonRepudiation(void)
105) CPKIFOID::CPKIFOID(std::string *)
106) char const* CPKIFOID::ToString(void)
107) CPKIFPolicyInformation::CPKIFPolicyInformation(CPKIFOIDPtr const &)
108) CPKIFOIDPtr CPKIFPolicyInformation::PolicyOID(void)
109) CPKIFPolicyQualifierListPtr CPKIFPolicyInformation::Qualifiers(void)
110) void CPKIFSubjectAltName::GeneralNames(CPKIFGeneralNames &)
111) CPKIFAlgorithmIdentifierPtr CPKIFSubjectPublicKeyInfo::alg(void)
112) struct ASN1DynBitStr* CPKIFSubjectPublicKeyInfo::rawKey(void)
113) CPKIFTime::CPKIFTime(char const *)
```

6.7 Assurance Measures

PKIF satisfies the assurance requirements for Evaluation Assurance Level EAL4 augmented with ALC_FLR.1. The following items are provided as evaluation evidence to satisfy the EAL4 augmented assurance requirements:

Table 6.6 Assurance Measures and How Satisfied

Assurance Component ID	Assurance Component Title	How Satisfied
ACM_AUT.1	Partial CM automation	[CMPLAN]
ACM_CAP.4	Generation support and acceptance procedures	[CMPLAN]
ACM_SCP.2	Problem tracking CM coverage	[CMPLAN]
ADO_DEL.2	Detection of modification	[DELIVERY]
ADO_IGS.1	Installation, generation, and start-up procedures	[DELIVERY]
ADV_FSP.2	Fully defined external interfaces	[HELP], [INT]
ADV_HLD.2	Security enforcing high-level design	[HELP], [INT], [PRIVATE]
ADV_IMP.1	Subset of the Implementation of the TSF	Source Code, [INT]
ADV_LLD.1	Descriptive low-level design	[HELP], [INT], [PRIVATE]
ADV_RCR.1	Informal correspondence demonstration	[RCR-S], [RCR-D]
ADV_SPM.1	Informal TOE security policy model	[ISPM]
AGD_ADM.1	Administrator guidance	[HELP]
AGD_USR.1	User guidance	[HELP]
ALC_DVS.1	Identification of security measures	[DEVSEC]
ALC_FLR.1	Basic flaw remediation	[CMPLAN]
ALC_LCD.1	Developer defined life-cycle model	[LCMOD]
ALC_TAT.1	Well-defined development tools	[DEVSEC]

Assurance Component ID	Assurance Component Title	How Satisfied
ATE_COV.2	Analysis of coverage	[TSTCOV]
ATE_DPT.1	Testing: high-level design	[TSTCOV]
ATE_FUN.1	Functional testing	[TEST], [TSTLST]
ATE_IND.2	Independent testing – sample	To be provided by the evaluation lab
AVA_MSU.2	Validation of analysis	[VULAN]
AVA_SOF.1	Strength of TOE security function evaluation	Not Applicable
AVA_VLA.2	Independent vulnerability analysis	[VULAN], [TEST]

The following is a description of the TOE assurance documents listed in the table above.

[CMPLAN] Public Key Infrastructure Framework (PKIF) Configuration Management Plan, Version 1.9.3, October 21, 2004.

This document contains the configuration management plan, list of configuration items, description of configuration management system and processes, and process for reporting, tracking, and expediting flaws found in the TOE.

[DELIVERY] Public Key Infrastructure Framework (PKIF) Delivery, Installation, Generation and Start-up Procedures, Version 1.2, August 2, 2004.

This document describes the secure delivery options. The document also contains a description with screen shots of TOE installation procedures.

[DEVSEC] Public Key Infrastructure Framework (PKIF) Development Security, Development Tools, Version 1.3, March 9, 2005

This document describes the development facility, personnel and environment security. The document also describes the tools used to develop the TOE.

[HELP] PKIF Usage Guide, Version 1.1.12.1, March 2005

The TOE help files contain configuration and usage instructions for the user. They also contain linked documentation of the TOE to facilitate easy navigation. Documentation includes the TOE header files providing TSFI details.

[INT] Public Key Infrastructure Framework (PKIF) Internals, Version 1.1.1, March 9, 2005

This document provides the structure of the TOE source code tree. The document explains the key object oriented concepts used. The document also provides a overview of how the following key TOE functionality is achieved: ASN.1 encoding and decoding; certification path processing; cryptographic processing, and CMS processing.

[ISPM] Public Key Infrastructure Framework (PKIF) Security Policy Model, Version 1.2, February 25, 2005.

This document contains the informal security policy model for the TOE. The following TOE enforced policies are described informally: certification path processing, signature generation, signature verification, encryption, decryption, and audit generation.

[LCMOD] Public Key Infrastructure Framework (PKIF) Life-Cycle Model, Version 1.1, July 12, 2004.

This document describes the life-cycle model used in the development and maintenance of the TOE.

[PRIVATE] Public Key Infrastructure Framework (PKIF) Private Member Function Description, Version 1.7, November 11, 2004

This document contains a description of the TOE software that does not provide TSFI. The software is described in terms of internal interface used to invoke and description of actions taken by the software when invoked.

[RCR-D] RCR Spreadsheet, Version 1.1.12, March 8, 2005

This spreadsheet provides a mapping of TSS functions to TSFI and TSFI to High Level Design and Low Level Design.

[RCR-S] Public Key Infrastructure Framework (PKIF) Correspondence Demonstration, Version 1.4, March 6, 2005.

This document provides an overview of the representation correspondence. The document includes an overview of TSS → Functional Specification Mapping; Functional Specification → High Level Design Mapping; High Level Design → Low Level Design Mapping; and Low Level Design → Implementation Representation Mapping.

[TEST] Public Key Infrastructure Framework (PKIF) Test Set-up and Execution, Version 1.3, March 7, 2005

This document describes the test setup and how to run the tests. The document also describes the test suites used; and TOE security functions (as described in the TOE Summary Specifications) and TOE SFRs tested by each test suite.

[TSTCOV] TSFI-To-TestCase-Mapping.xls

This spreadsheet provides a mapping from test case to TSFI and to TOE subsystems.

[TSTLST] Test_Case_List.xls

This spreadsheet provides a mapping of TOE test number to test cases.

[VULAN]	Public Key Infrastructure Framework (PKIF) Vulnerability Analysis, Version 1.2, March 8, 2005

This document describes the vulnerability analysis o the TOE. The document contains descriptions of potential vulnerabilities, their disposition and results of penetration testing conducted by the TOE Developer. The document also contains the analysis of other deliverables such as the Administrator and User Guidance document.

7 PP Conformance

This ST is conformant with the *U.S. Government PKE PP with:*

- *Certification Path Validation (CPV) – Basic Package,*
- *CPV – Basic Policy Package,*
- *CPV – Policy Mapping Package,*
- *CPV – Name Constraints Package,*
- *PKI Signature Generation Package,*
- *PKI Signature Verification Package,*
- *PKI Encryption using Key Transfer Algorithms Package,*
- *PKI Decryption using Key Transfer Algorithms Package,*
- *Online Certificate Status Protocol (OCSP) Client Package, and*
- *Certificate Revocation List (CRL) Validation Package*

at EAL4 with augmentation. (Version: 2.61; Date: July 31, 2004; Prepared for: US Marine Corps,)

The following sections provide the evidence of the conformance with the PP:

7.1 Conformance with PP Requirements

The completed operations are marked in section 5. At the beginning of section 5, the formatting of the operations is described. All operations on the SFRs within section 5 follow this formatting.

7.2 Conformance with PP Assumptions

This ST is conformant with the PP security assumptions for the IT environment. The following table provides the evidence of this conformance:

Table 7.1 – Conformance with PP Base Assumptions for IT Environment

Base Assumptions for the IT Environment			
#	**PP Assumption Name**	**Description**	**ST Assumption Name**
1	AE.Authorized_Users	Authorized users are trusted to perform their assigned functions.	AE.Authorized_Users
2	AE.Configuration	The TOE will be properly installed and configured.	AE.Configuration

Base Assumptions for the IT Environment			
#	**PP Assumption Name**	**Description**	**ST Assumption Name**
3	AE.Crypto_Module	The TOE environment is assumed to include one or more cryptographic module(s) that are all validated at FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, Hash based Message Authentication Code (HMAC) and/or other required cryptographic functions. In summary, all cryptographic modules in the TOE shall be validated at FIPS 140 series Level 1.	AE.Crypto_Module
4	AE.Low	The attack potential on the TOE is assumed to be low.	AE.Low
5	AE.Physical_Protection	Physical protection is assumed to be provided by the environment. The TOE hardware and software is assumed to be protected from unauthorized physical access.	AE.Physical_Protection
6	AE.PKI_Info	The certificate and certificate revocation information is available to the TOE for the time of interest (TOI).	AE.PKI_Info
7	AE.Time	Accurate system time with required precision in GMT format is assumed to be provided by the environment.	AE.Time

7.3 Conformance with PP Threats

7.3.1 Conformance with PP Threats to TOE Security

This ST is conformant with the PP security threats for the TOE. The following table provides the evidence of this conformance.

Table 7.2 – Conformance with PP Threats to TOE Security

<table>
<tr><th colspan="4">Threats for the 1. CPV – Basic Package</th></tr>
<tr><th>#</th><th>PP Threat Name</th><th>Threat Description</th><th>ST Threat Name</th></tr>
<tr><td>1</td><td>T.Certificate_Modi</td><td>An untrusted user may modify a certificate resulting in using a wrong public key.</td><td>T.Certificate_Modi</td></tr>
<tr><td>2</td><td>T.DOS_CPV_Basic</td><td>The revocation information or access to revocation information could be made unavailable, resulting in loss of system availability.</td><td>T.DOS_CPV_Basic</td></tr>
<tr><td>3</td><td>T.Expired_Certificate</td><td>An expired (and possibly revoked) certificate as of TOI could be used for signature verification.</td><td>T.Expired_Certificate</td></tr>
<tr><td>4</td><td>T.Masquarade</td><td>An untrusted entity (Certification Authority (CA)) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users.</td><td>T.Masquarade</td></tr>
<tr><td>5</td><td>T.No_Crypto</td><td>The user public key and related information may not be available to carry out the cryptographic function.</td><td>T.No_Crypto</td></tr>
<tr><td>6</td><td>T.Path_Not_Found</td><td>A valid certification path is not found due to lack of system functionality.</td><td>T.Path_Not_Found</td></tr>
<tr><td>7</td><td>T.Revoked_Certificate</td><td>A revoked certificate could be used as valid, resulting in security compromise.</td><td>T.Revoked_Certificate</td></tr>
<tr><td>8</td><td>T.User_CA</td><td>A user could act as a CA, issuing unauthorized certificates.</td><td>T.User_CA</td></tr>
<tr><th colspan="4">Threats for the 2. CPV – Basic Policy Package</th></tr>
<tr><th>#</th><th>PP Threat Name</th><th>Threat Description</th><th>ST Threat Name</th></tr>
<tr><td>9</td><td>T.Unknown_Policies</td><td>The user may not know the policies under which a certificate was issued.</td><td>T.Unknown_Policies</td></tr>
<tr><th colspan="4">Threats for the 3. CPV – Policy Mapping Package</th></tr>
<tr><th>#</th><th>PP Threat Name</th><th>Threat Description</th><th>ST Threat Name</th></tr>
<tr><td>10</td><td>T.Mapping</td><td>The user may accept unacceptable certificates or reject acceptable certificates due to improper certificate policy mapping.</td><td>T.Mapping</td></tr>
<tr><td>11</td><td>T.Wrong_Policy_Dec</td><td>The user may accept certificates that were not generated with the diligence and security acceptable to the user. The user may reject certificates that were generated with the diligence and security acceptable to the user.</td><td>T.Wrong_Policy_Dec</td></tr>
<tr><th colspan="4">Threats for the 4. CPV – Name Constraints Package</th></tr>
<tr><th>#</th><th>PP Threat Name</th><th>Threat Description</th><th>ST Threat Name</th></tr>
<tr><td>12</td><td>T.Name_Collision</td><td>The user may accept certificates from CA where the CA's understanding and the user's understanding of the names differ, i.e., user and CA associate different identity with the same name.</td><td>T.Name_Collision</td></tr>
</table>

Threats for the 5. PKI Signature Generation Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
13	T.Clueless_PKI_Sig	The user may try only inappropriate certificates for signature in absence of hint.[1]	T.Clueless_PKI_Sig
Threats for the 6. PKI Signature Verification Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
14	T.Assumed_Identity_PKI_Ver	A user may assume the identity of another user in order to verify a PKI signature.	T.Assumed_Identity_PKI_Ver
15	T.Clueless_PKI_Ver	The user may try only inappropriate certificates for verification in absence of hint.[2]	T.Clueless_PKI_Ver
Threats for the 7. PKI Encryption using Key Transfer Algorithms Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
16	T.Assumed_Identity_WO_En	A user may assume the identity of another user in order to perform encryption using Key Transfer algorithms.	T.Assumed_Identity_WO_En
17	T.Clueless_WO_En	The user may try only inappropriate certificates for encryption using Key Transfer algorithms in absence of hint.	T.Clueless_WO_En
Threats for the 8. PKI Decryption using Key Transfer Algorithms Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
18	T.Garble_WO_De	The user may not apply the correct key transfer algorithm or private key, resulting in garbled data.	T.Garble_WO_De
Threats for the 9. OCSP Client Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
19	T.DOS_OCSP	The OCSP response or access to the OCSP response could be made unavailable, resulting in loss of system availability.	T.DOS_OCSP
20	T.Replay_OCSP_Info	The user may accept an old OCSP response resulting in accepting a currently revoked certificate.	T.Replay_OCSP_Info
21	T.Wrong_OCSP_Info	The user may accept a revoked certificate or reject a valid certificate due to a wrong OCSP response.	T.Wrong_OCSP_Info
Threats for the 10. Certificate Revocation List (CRL) Validation Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
22	T.DOS_CRL	The CRL or access to CRL could be made unavailable, resulting in loss of system availability.	T.DOS_CRL
23	T.Replay_Revoc_Info_CRL	The user may accept a CRL issued before TOI resulting in accepting a revoked certificate.	T.Replay_Revoc_Info_CRL
24	T.Wrong_Revoc_Info_CRL	The user may accept a revoked certificate or reject a valid certificate due to a wrong CRL.	T.Wrong_Revoc_Info_CRL

[1] There are minor differences in the wording of the threat. These threats are the same with ST words being clearer.

[2] There are minor differences in the wording of the threat. These threats are the same with ST words being clearer.

7.3.2 Conformance with PP Threats to IT Environment Security

This ST is conformant with the PP security threats for the IT environment. The following table provides the evidence of this conformance.

Table 7.3 - Conformance with PP Threats to IT Environment Security

Base Threats to Security for all PPs in this PP Family			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
1E	T.Attack	An undetected compromise of the TOE assets may occur as a result of an attacker (whether an insider or outsider) attempting to perform actions that the individual is not authorized to perform.	TE.Attack
2E	T.Bypass	An unauthorized individual or user may tamper with security attributes or other data in order to bypass TOE security functions and gain unauthorized access to TOE assets.	TE.Bypass
3E	T.Imperson	An unauthorized individual may impersonate an authorized user of the TOE and thereby gain access to TOE data, keys, and operations.	TE.Imperson
4E	T.Modify	An attacker may modify TSF or user data, e.g., stored security attributes or keys, in order to gain access to the TOE and its assets.	TE.Modify
5E	T.Object_Init	An attacker may gain unauthorized access to an object upon its creation, if the security attributes are not assigned to the object or any one can assign the security attributes upon object creation.	TE.Object_Init
6E	T.Private_Key	An attacker may assume the identity of a user by generating or using the private key of the user.	TE.Private_Key
7E	T.Role	A user may assume more privileged role than permitted and use the enhanced privilege to take unauthorized actions.	TE.Role
8E	T.Secure Attributes	A user may be able to change the security attributes of an object and gain unauthorized access to the object.	TE.Secure Attributes
9E	T.Shoulder_Surf	An authorized user may look over the shoulder of the authorized user while authentication is in progress and read the authentication information.	TE.Shoulder_Surf
10E	T.Tries	An unauthorized individual may guess the authentication information using trial and error.	TE.Tries

7.4 Conformance with PP Objectives

7.4.1 Conformance with PP Objectives for IT Environment

This ST is conformant with PP objectives for IT environment. The following table provides the evidence of this conformance.

Table 7.4 – Conformance with PP Security Objectives for the IT Environment

Security Objectives for the TOE for all PPs in this PP Family			
#	**PP Objective Name**	**Objective Description**	**ST Objective Name**
1E	O.DAC	The TSF shall control and restrict user access to the TOE assets in accordance with a specified access control policy.	OE.DAC
2E	O.I&A	The TSF shall uniquely identify all users, and shall authenticate the claimed identify before granting a user access to the TOE facilities.	OE.I&A
3E	O.Init_Secure_Attr	The TSF shall provide valid default security attributes when an object is initialized.	OE.Init_Secure_Attr
4E	O.Invoke	The TSF shall be invoked for all actions.	OE.Invoke
5E	O.Limit_Actions_Auth	The TSF shall restrict the actions a user may perform before the TSF verifies the identity of the user.	OE.Limit_Actions_Auth
6E	O.Limit_Tries	The TSF shall restrict the number of consecutive unsuccessful authentication attempts.	OE.Limit_Tries
7E	O.No_Echo	The TSF shall not echo the authentication information.	OE.No_Echo
8E	O.Protect_I&A_Data	The TSF shall permit only authorized users to change the I&A data.	OE.Protect_I&A_Data
9E	O.Secure_Attributes	The TSF shall permit only the authorized users to change the security attributes.	OE.Secure_Attributes
10E	O.Security_Roles	The TSF shall maintain security-relevant roles and association of users with those roles.	OE.Security_Roles
11E	O.Self_Protect	The TSF shall maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure.	OE.Self_Protect
12E	O.Trust_Anchor	The TSF shall permit only authorized users to manage the trust anchors.	OE.Trust_Anchor
13E	O.TSF_Data	The TSF shall permit only authorized users to modify the TSF data.	OE.TSF_Data

Security Objectives for the Environment			
#	PP Objective Name	Objective Description	ST Objective Name
14E	O.Authorized_Users	Authorized users are trusted to perform their authorized tasks.	OE.Authorized_Users
15E	O.Configuration	The TOE shall be installed and configured properly for starting up the TOE in a secure state.	OE.Configuration
16E	O.Crypto	The IT environment shall include one or more cryptographic modules that are validated at FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, HMAC and/or other required cryptographic functions. In summary, all cryptographic modules within the TOE shall be FIPS 140 series level 1 validated.	OE.Crypto
17E	O.Low	The Identification and Authentication functions in the TOE shall be designed and implemented for a minimum attack potential of low as validated by the vulnerability assessment and Strength of Function analyses.	OE.Low
18E	O.Physical_Security	The environment shall provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.	OE.Physical_Security
19E	O.PKI_Info	The IT environment shall provide the TOE certificate and certificate revocation information for the time of interest (TOI).	OE.PKI_Info
20E	O.Time	The environment shall provide access to accurate current time with required precision, translated to GMT.	OE.Time

7.4.2 Conformance with PP Objectives for TOE

This ST is conformant with PP objectives for TOE. The following table provides the evidence of this conformance.

Table 7.5 – Conformance with Security Objectives for the TOE

<table>
<tr><th colspan="4">Security Objectives for 1. CPV – Basic Package</th></tr>
<tr><th>#</th><th>PP Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>1</td><td>O.Availability</td><td>The TSF shall continue to provide security services even if revocation information is not available.</td><td>O.Availability</td></tr>
<tr><td>2</td><td>O.Correct_Temporal</td><td>The TSF shall provide accurate temporal validation results.</td><td>O.Correct_Temporal</td></tr>
<tr><td>3</td><td>O.Current_Certificate</td><td>The TSF shall only accept certificates that are not expired as of TOI.</td><td>O.Current_Certificate</td></tr>
<tr><td>4</td><td>O.Get_KeyInfo</td><td>The TSF shall provide the user public key and related information in order to carry out cryptographic functions.</td><td>O.Get_KeyInfo</td></tr>
<tr><td>5</td><td>O.Path_Find</td><td>The TSF shall be able to find a certification path from a trust anchor to the subscriber.</td><td>O.Path_Find</td></tr>
<tr><td>6</td><td>O.Trusted_Keys</td><td>The TSF shall use trusted public keys in certification path validation.</td><td>O.Trusted_Keys</td></tr>
<tr><td>7</td><td>O.User</td><td>The TSF shall only accept certificates issued by a CA.</td><td>O.User</td></tr>
<tr><td>8</td><td>O.Verified_Certificate</td><td>The TSF shall only accept certificates with verifiable signatures.</td><td>O.Verified_Certificate</td></tr>
<tr><td>9</td><td>O.Valid_Certificate</td><td>The TSF shall use certificates that are valid, i.e., not revoked.</td><td>O.Valid_Certificate</td></tr>
<tr><th colspan="4">Security Objectives for 2. CPV – Basic Policy Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>10</td><td>O.Provide_Policy_Info</td><td>The TSF shall provide certificate policies for which the certification path is valid.</td><td>O.Provide_Policy_Info</td></tr>
<tr><th colspan="4">Security Objectives for 3. CPV – Policy Mapping Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>11</td><td>O.Map_Policies</td><td>The TSF shall map certificate policies in accordance with user and CA constraints.</td><td>O.Map_Policies</td></tr>
<tr><td>12</td><td>O.Policy_Enforce</td><td>The TSF shall validate a certification path in accordance with certificate policies acceptable to the user.</td><td>O.Policy_Enforce</td></tr>
<tr><th colspan="4">Security Objectives for 4. CPV – Name Constraints Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>13</td><td>O.Authorised_Names</td><td>The TSF shall validate a certificate only if the CA is authorized to issue a certificate to the subject.</td><td>O.Authorised_Names</td></tr>
</table>

<table>
<tr><th colspan="4">Security Objectives for 5. PKI Signature Generation Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>14</td><td>O.Give_Sig_Hints</td><td>The TSF shall provide hints for selecting correct certificates for signature verification.</td><td>O.Give_Sig_Hints</td></tr>
<tr><th colspan="4">Security Objectives for 6. PKI Signature Verification Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>15</td><td>O.Use_Sig_Hints</td><td>The TSF shall use hints for selecting correct certificates for signature verification.</td><td>O.Use_Sig_Hints</td></tr>
<tr><td>16</td><td>O.Linkage_Sig_Ver</td><td>The TSF shall use the correct user public key for signature verification.</td><td>O.Linkage_Sig_Ver</td></tr>
<tr><th colspan="4">Security Objectives for 7. PKI Encryption using Key Transfer Algorithms Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>17</td><td>O.Hints_Enc_WO</td><td>The TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer Algorithms.</td><td>O.Hints_Enc_WO</td></tr>
<tr><td>18</td><td>O.Linkage_Enc_WO</td><td>The TSF shall use the correct user public key for key transfer.</td><td>O.Linkage_Enc_WO</td></tr>
<tr><th colspan="4">Security Objectives for 8. PKI Decryption using Key Transfer Algorithms Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>19</td><td>O.Correct_KT</td><td>The TSF shall use appropriate private key and key transfer algorithm.</td><td>O.Correct_KT</td></tr>
<tr><th colspan="4">Security Objectives for 9. OCSP Client Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>20</td><td>O.Accurate_OCSP_Info</td><td>The TSF shall accept only accurate OCSP responses.</td><td>O.Accurate_OCSP_Info</td></tr>
<tr><td>21</td><td>O.Auth_OCSP_Info</td><td>The TSF shall accept the revocation information from an authorized source for OCSP transactions.</td><td>O.Auth_OCSP_Info</td></tr>
<tr><td>22</td><td>O.Current_OCSP_Info</td><td>The TSF accept only OCSP responses current as of TOI.</td><td>O.Current_OCSP_Info</td></tr>
<tr><td>23</td><td>O.User_Override_Time_OCSP</td><td>The TSF shall permit the user to override the time checks on the OCSP response.</td><td>O.User_Override_Time_OCSP</td></tr>
<tr><th colspan="4">Security Objectives for 10. Certificate Revocation List (CRL) Validation Package</th></tr>
<tr><th>#</th><th>Objective Name</th><th>Objective Description</th><th>ST Objective Name</th></tr>
<tr><td>24</td><td>O.Accurate_Rev_Info</td><td>The TSF shall accept only accurate revocation information.</td><td>O.Accurate_Rev_Info</td></tr>
<tr><td>25</td><td>O.Auth_Rev_Info</td><td>The TSF shall accept the revocation information from an authorized source for CRL.</td><td>O.Auth_Rev_Info</td></tr>
<tr><td>26</td><td>O.Current_Rev_Info</td><td>The TSF shall accept only CRL that are current as of TOI</td><td>O.Current_Rev_Info</td></tr>
<tr><td>27</td><td>O.User_Override_Time_CRL</td><td>The TSF shall permit the user to override the time checks on the CRL.</td><td>O.User_Override_Time_CRL</td></tr>
</table>

8 Rationale

8.1 Security Objectives Rationale

8.1.1 Base and Environmental Security Objectives Rationale for TOE

Table 8.1 maps base assumptions and threats to objectives, demonstrating that all assumptions and threats are mapped to at least one objective. Table 8.2 maps base objectives to threats and assumptions, demonstrating that all objectives are mapped to at least one threat or assumption.

Table 8.1 – Mapping the Base Assumptions and Threats to Objectives

Assumption/Threat	Objectives
AE.Authorized_Users	OE.Authorized_Users
AE.Configuration	OE.Configuration
AE.Crypto_Module	OE.Crypto
AE.Low	OE.Low
AE.PKI_Info	OE.PKI_Info
AE.Physical_Protection	OE.Physical_Security
AE.Time	OE.Time
TE.Attack	OE.DAC
TE.Bypass	OE.Invoke
TE.Imperson	OE.I&A, OE.Limit_Actions_Auth
TE.Modify	OE.Self_Protect, OE.DAC, OE.Protect_I&A_Data, OE.Trust_Anchor, OE.TSF_Data
TE.Object_Init	OE.Init_Secure_Attr
TE.Private_Key	OE.DAC
TE.Role	OE.Security_Roles
TE.Secure Attributes	OE.Secure_Attributes
TE.Shoulder_Surf	OE.No_Echo
TE.Tries	OE.Limit_Tries

AE.Authorized_Users states that authorized users are trusted to perform their assigned functions. This assumption is mapped to:

- **OE.Authorized_Users**, which states that authorized users are trusted to perform their authorized tasks.

AE.Configuration states that the TOE will be properly installed and configured. This assumption is mapped to:

- **OE.Configuration**, which states that the TOE shall be installed and configured properly for starting up the TOE in a secure state.

AE.Crypto_Module states that the TOE environment is assumed to include one or more cryptographic module(s) that are all validated at FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation, encryption, decryption, secure hash, random number generation, HMAC and/or other required cryptographic functions. In summary, all cryptographic modules in the TOE shall be validated at FIPS 140 series Level 1. This assumption is mapped to:

- **OE.Crypto**, which states that the environment shall include one or more cryptographic modules that are validated at FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, HMAC and/or other required cryptographic functions. In summary, all cryptographic modules within the TOE shall be FIPS 140 series level 1 validated.

AE.Low states that the attack potential on the TOE is assumed to be low. AE.Low is mapped to:

- **OE.Low,** which states that the Identification and Authentication functions in the TOE will be designed for a minimum attack potential of low as validated by the vulnerability assessment and Strength of Function analyses.

AE.PKI_Info states that the certificate and certificate revocation information for the time of interest (TOI) is available to the TOE. AE.PKI_Info is mapped to:

> **OE.PKI_Info**, which states that the IT environment shall provide the TOE certificate and certificate revocation information for the time of interest (TOI).

AE.Physical_Protection states that physical protection is assumed to be provided by the environment. The TOE hardware and software is assumed to be protected from unauthorized physical access. This assumption is mapped to:

- **OE.Physical_Security**, which states that the environment shall provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis.

AE.Time states that accurate system time with required precision in GMT format is assumed to be provided by the environment. This assumption is mapped to:

- **OE.Time**, which states that the environment shall provide access to accurate current time with required precision, translated to GMT.

TE.Attack states that an undetected compromise of the TOE assets may occur as a result of an attacker (whether an insider or outsider) attempting to perform actions that the individual is not authorized to perform. This threat is mapped to:

- **OE.DAC**, which states that the TSF shall control and restrict user access to the TOE assets in accordance with a specified access control policy.

TE.Bypass states that an unauthorized individual or user may tamper with security attributes or other data in order to bypass TOE security functions and gain unauthorized access to TOE assets. This threat is mapped to:

- **OE.Invoke**, which states that the TSF shall be invoked for all actions.

TE.Imperson states that an unauthorized individual may impersonate an authorized user of the TOE and thereby gain access to TOE data, keys, and operations. This threat is mapped to:

- **OE.I&A**, which states that the TSF shall uniquely identify all users, and shall authenticate the claimed identify before granting a user access to the TOE facilities.
- **OE.Limit_Actions_Auth**, which states that the TSF shall restrict the actions a user may perform before the TSF verifies the identity of the user.

TE.Modify states that an attacker may modify TSF or user data, e.g., stored security attributes or keys, in order to gain access to the TOE and its assets. This threat is mapped to:

- **OE.Self_Protect**, which states that the TSF will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure.
- **OE.DAC**, which states that the TSF shall control and restrict user access to the TOE assets in accordance with a specified access control policy.
- **OE.Protect_I&A_Data**, which states that the TSF shall permit only authorized users to change the I&A data.
- **OE.Trust_Anchor**, which states that the TSF shall permit only authorized users to manage the trust anchors.
- **OE.TSF_Data**, which states that the TSF shall permit only authorized users to modify the TSF data.

TE.Object_Init states that an attacker may gain unauthorized access to an object upon its creation, if the security attributes are not assigned to the object or any one can assign the security attributes upon object creation. This threat is mapped to:

- **OE.Init_Secure_Attr**, which states that the TSF shall provide valid default security attributes when an object is initialized.

TE.Private_Key states that an attacker may assume the identity of a user by generating or using the private key of the user. This threat is mapped to:

- **OE.DAC**, which states that the TSF shall control and restrict user access to the TOE assets in accordance with a specified access control policy.

TE.Role states that a user may assume more privileged role than permitted and use the enhanced privilege to take unauthorized actions. This threat is mapped to:

- **OE.Security_Roles**, which state that the TSF shall maintain security-relevant roles and association of users with those roles.

TE.Secure Attributes states that a user may be able to change the security attributes of an object and gain unauthorized access to the object. This threat is mapped to:

- **OE.Secure_Attributes**, which states that the TSF shall permit only the authorized users to change the security attributes.

TE.Shoulder_Surf states that an authorized user may look over the shoulder of the authorized user while authentication is in progress and read the authentication information. This threat is mapped to:

- **OE.No_Echo**, which states that the TSF shall not echo the authentication information.

TE.Tries states that an unauthorized individual may guess the authentication information using trial and error. This threat is mapped to:

- **OE.Limit_Tries**, which states that the TSF shall restrict the number of consecutive unsuccessful authentication attempts.

In Table 8.2, the Base TOE and Environmental Objectives are mapped back to threats and assumptions, thereby demonstrating that every objective is mapped to a threat or assumption. Explanation of the mapping is defined above and is not repeated following Table 8.2.

Note, once again, these threats and objectives are included in every PP in this PP family.

Table 8.2 – Mapping of Base TOE and Environmental Objectives to Threats and Assumptions

Objective	Threats
OE.Authorized_Users	AE.Authorized_Users
OE.Configuration	AE.Configuration
OE.Crypto	AE.Crypto_Module
OE.Low	AE.Low
OE.PKI_Info	AE.PKI_Info
OE.Physical_Security	AE.Physical_Protection
OE.Time	AE.Time
OE.DAC	TE.Attack, TE.Modify, TE.Private_Key
OE.I&A	TE.Imperson
OE.Init_Secure_Attr	TE.Object_Init
OE.Invoke	TE.Bypass
OE.Limit_Actions_Auth	TE.Imperson
OE.Limit_Tries	TE.Tries
OE.No_Echo	TE.Shoulder_Surf
OE.Protect_I&A_Data	TE.Modify
OE.Secure_Attributes	TE.Secure Attributes
OE.Security_Roles	TE.Role
OE.Self_Protect	TE.Modify
OE.Trust_Anchor	TE.Modify
OE.TSF_Data	TE.Modify

8.1.2 Security Objectives Rationale for the TOE

Table 8.3 below demonstrates the mapping of threats to objectives for the applicable family of PP packages. Explanatory text is provided below the table to support the mapping. Table 8.4 maps objectives to threats, demonstrating that all objectives are mapped to at least one threat.

Table 8.3 – Mapping of TOE Security Threats to Objectives

1. CPV – Basic Package		
#	**Threat**	**Objectives**
1	T.Certificate_Modi	O.Verified_Certificate
2	T.DOS_CPV_Basic	O.Availability
3	T.Expired_Certificate	O.Correct_Temporal O.Current_Certificate
4	T.Masquarade	O.Trusted_Keys
5	T.No_Crypto	O.Get_KeyInfo
6	T.Path_Not_Found	O.Path_Find
7	T.Revoked_Certificate	O.Valid_Certificate
8	T.User_CA	O.User
2. CPV – Basic Policy Package		
#	**Threat**	**Objectives**
9	T.Unknown_Policies	O.Provide_Policy_Info
3. CPV - Policy Mapping Package		
#	**Threat**	**Objectives**
10	T.Mapping	O.Map_Policies
11	T.Wrong_Policy_Dec	O.Policy_Enforce
4. CPV – Name Constraints Package		
#	**Threat**	**Objectives**
12	T.Name_Collision	O.Authorised_Names
5. PKI Signature Generation Package		
#	**Threat**	**Objectives**
13	T.Clueless_PKI_Sig	O.Give_Sig_Hints
6. PKI Signature Verification Package		
#	**Threat**	**Objectives**
14	T.Assumed_Identity_PKI_Ver	O.Linkage_Sig_Ver
15	T.Clueless_PKI_Ver	O.Use_Sig_Hints
7. PKI Encryption using Key Transfer Algorithms Package		
#	**Threat**	**Objectives**
16	T.Assumed_Identity_WO_En	O.Linkage_Enc_WO
17	T.Clueless_WO_En	O.Hints_Enc_WO
8. PKI Decryption using Key Transfer Algorithms Package		
#	**Threat**	**Objectives**
18	T.Garble_WO_De	O.Correct_KT

9. OCSP Client Package		
#	**Threat**	**Objectives**
19	T.DOS_OCSP	O.User_Override_Time_OCSP
20	T.Replay_OCSP_Info	O.Current_OCSP_Info
21	T.Wrong_OCSP_Info	O.Accurate_OCSP_Info, O.Auth_OCSP_Info
10. CRL Validation Package		
	Threat	**Objectives**
22	T.DOS_CRL	O.User_Override_Time_CRL
23	T.Replay_Revoc_Info_CRL	O.Current_Rev_Info
24	T.Wrong_Revoc_Info_CRL	O.Accurate_Rev_Info, O.Auth_Rev_Info

8.1.2.1 CPV – Basic Package Security Objectives Rationale

T.Certificate_Modi states that an untrusted user may modify a certificate resulting in using a wrong public key. This threat is mapped to:

- **O.Verified_Certificate**, which states that the TSF shall only accept certificates with verifiable signatures.

T.DOS_CPV_Basic states that the revocation information or access to revocation information could be made unavailable, resulting in loss of system availability. This threat is mapped to:

- **O.Availability**, which states that the TSF shall continue to provide security services even if revocation information is not available.

T.Expired_Certificate states that an expired (and possibly revoked) certificate as of TOI could be used for signature verification. This threat is mapped to:

- **O.Correct_Temporal**, which states that the TSF shall provide accurate temporal validation results.
- **O.Current_Certificate**, which states that the TSF shall only accept certificates that are not expired as of TOI.

T.Masquarade states that an untrusted entity (Certification Authority (CA)) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users. This threat is mapped to:

- **O.Trusted_Keys**, which states that the TSF shall use trusted public keys in certification path validation.

T.No_Crypto states that the user public key and related information may not be available to carry out the cryptographic function. This threat is mapped to:

- **O.Get_KeyInfo**, which states that the TSF shall provide the user public key and related information in order to carry out cryptographic functions.

T.Path_Not_Found states that a valid certification path is not found due to lack of system functionality. This threat is mapped to:

- **O.Path_Find**, which states that the TSF shall be able to find a certification path from a trust anchor to the subscriber.

T.Revoked_Certificate states that a revoked certificate could be used as valid, resulting in security compromise. This threat is mapped to:

- **O.Valid_Certificate**, which states that the TSF shall use certificates that are valid, i.e., not revoked.

T.User_CA states that a user could act as a CA, issuing unauthorized certificates. This threat is mapped to:

- **O.User**, which states that the TSF shall only accept certificates issued by a CA.

8.1.2.2 CPV – Basic Policy Package Security Objectives Rationale

T.Unknown_Policies states that the user may not know the policies under which a certificate was issued. This threat is mapped to:

- **O.Provide_Policy_Info**, which states that the TSF shall provide certificate policies for which the certification path is valid.

8.1.2.3 CPV –Policy Mapping Package Security Objectives Rationale

T.Mapping states that the user may accept unacceptable certificates or reject acceptable certificates due to improper certificate policy mapping. This threat is addressed by:

- **O.Map_Policies**, which states that the TSF shall map certificate policies in accordance with user and CA constraints.

T.Wrong_Policy_Dec states that the user may accept certificates that were not generated with the diligence and security acceptable to the user. The user may reject certificates that were generated with the diligence and security acceptable to the user. This threat is addressed by:

- **O.Policy_Enforce**, which states that he TSF shall validate a certification path in accordance with certificate policies acceptable to the user.

8.1.2.4 CPV – Name Constraints Package Security Objectives Rationale

T.Name_Collision states that the user may accept certificates from CA where the CA's understanding and the user's understanding of the names differ, i.e., user and CA associate different identity with the same name. This threat is addressed by:

- **O.Authorised_Names**, which states that the TSF shall validate a certificate only if the CA is authorized to issue a certificate to the subject.

Table 6.10 maps objectives for the CPV – Name Constraints Package to threats, demonstrating that every objective is mapped to a threat. The mapping is described in the text above and is not repeated following Table 6.10.

8.1.2.5 PKI Signature Generation Package Security Objectives Rationale

T.Clueless_PKI_Sig states that the user may try only inappropriate certificates for PKI signature verification because the signature does not include a hint. This threat is addressed by:

- **O.Give_Sig_Hints**, which states that the TSF shall give hints for selecting correct certificates or keys for PKI signature.

8.1.2.6 PKI Signature Verification Package Security Objectives Rationale

T.Assumed_Identity_PKI_Ver states that a user may assume the identity of another user for PKI signature verification. This threat is addressed by:

- **O.Linkage_Sig_Ver**, which states that the TSF shall use the correct user public key for signature verification.

T.Clueless_PKI_Ver states that the user may try only inappropriate certificates for PKI signature verification because hints in the signature are ignored. This threat is addressed by:

- **O.Use_Sig_Hints**, which states that the TSF shall provide hints for selecting correct certificates or keys for signature verification.

8.1.2.7 PKI Encryption using Key Transfer Algorithms Package Security Objectives Rationale

T.Assumed_Identity_WO_En states that a user may assume the identity of another user in order to perform encryption using Key Transfer algorithms. This threat is addressed by:

- **O.Linkage_Enc_WO**, which states that the TSF shall use the correct user public key for key transfer.

T.Clueless_WO_En states that the user may try only inappropriate certificates in absence of hint for encryption using Key Transfer algorithms. This threat is addressed by:

- **O.Hints_Enc_WO**, which states that the TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer algorithms.

8.1.2.8 PKI Decryption using Key Transfer Algorithms Package Security Objectives Rationale

T.Garble_WO_De states that the user may not apply the correct key transfer algorithm or private key, resulting in garbled data. This threat is addressed by:

- **O.Correct_KT**, which states that the TSF shall use appropriate private key and key transfer algorithm.

8.1.2.9 OCSP Client Package Security Objectives Rationale

T.DOS_OCSP states that the OCSP response or access to the OCSP response could be made unavailable, resulting in loss of system availability. This threat is mapped to:

- **O.User_Override_Time_OCSP**, which states that the TSF shall permit the user to override the time checks on the OCSP response.

T.Replay_OCSP_Info states that the user may accept revocation information from well before TOI resulting in accepting currently revoked certificate for OCSP transactions. This threat is mapped to:

- **O.Current_OCSP_Info**, which states that the TSF accept only OCSP responses current as of TOI.

T.Wrong_OCSP_Info states that the user may accept a revoked certificate or reject a valid certificate due to wrong revocation information. This threat is mapped to:

- **O.Accurate_OCSP_Info**, which states that the TSF shall accept only accurate OCSP responses.
- **O.Auth_OCSP_Info**, which states that the TSF shall accept the OCSP response from an authorized source.

8.1.2.10 CRL Validation Package Security Objectives Rationale

T.DOS_CRL states that the CRL or access to the CRL could be made unavailable, resulting in loss of system availability. This threat is mapped to:

- **O.User_Override_Time_CRL**, which states that the TSF shall permit the user to override the time checks on the CRL.

T.Replay_Revoc_Info_CRL states that the user may accept a CRL issued before TOI resulting in accepting currently revoked certificate. This threat is mapped to:

- **O.Current_Rev_Info**, which states that the TSF shall accept only CRL that are current as TOI.

T.Wrong_Revoc_Info_CRL states that the user may accept a revoked certificate or reject a valid certificate due to wrong revocation information. This threat is mapped to:

- **O.Accurate_Rev_Info**, which states that the TSF shall accept only accurate revocation information.
- **O.Auth_Rev_Info**, which states that the TSF shall accept the revocation information from an authorized source for CRL.

In Table 8.4below, the TOE security objectives are mapped back to threats, thereby demonstrating that every objective is mapped to a threat. The mapping is defined in te text above and is not repeated following Table 8.4.

Note, once again, these threats and objectives are included in every PP in this PP family.

Table 8.4 – Mapping of TOE Security Objectives to Threats

1. CPV – Basic Package		
#	**Objective**	**Threats**
1	O.Availability	T.DOS_CPV_Basic
2	O.Correct_Temporal	T.Expired_Certificate
3	O.Current_Certificate	T.Expired_Certificate
4	O.Get_KeyInfo	T.No_Crypto
5	O.Path_Find	T.Path_Not_Found

6	O.Trusted_Keys	T.Masquarade
7	O.User	T.User_CA
8	O.Verified_Certificate	T.Certificate_Modi
9	O.Valid_Certificate	T.Revoked_Certificate
2. CPV – Basic Policy Package		
#	**Objective**	**Threats**
10	O.Provide_Policy_Info	T.Unknown_Policies
3. CPV - Policy Mapping Package		
#	**Objective**	**Threats**
11	O.Map_Policies	T.Mapping
12	O.Policy_Enforce	T.Wrong_Policy_Dec
4. CPV – Name Constraints Package		
#	**Objective**	**Threats**
13	O.Authorised_Names	T.Name_Collision
5. PKI Signature Generation Package		
#	**Objective**	**Threats**
14	O.Give_Sig_Hints	T.Clueless_PKI_Sig
6. PKI Signature Verification Package		
#	**Objective**	**Threats**
15	O.Use_Sig_Hints	T.Clueless_PKI_Ver
16	O.Linkage_Sig_Ver	T.Assumed_Identity_PKI_Ver
7. PKI Encryption using Key Transfer Algorithms Package		
#	**Objective**	**Threats**
17	O.Hints_Enc_WO	T.Clueless_WO_En
18	O.Linkage_Enc_WO	T.Assumed_Identity_WO_En
8. PKI Decryption using Key Transfer Algorithms Package		
#	**Objective**	**Threats**
19	O.Correct_KT	T.Garble_WO_De
9. OCSP Client Package		
#	**Objective**	**Threats**
20	O.Accurate_OCSP_Info	T.Wrong_OCSP_Info
21	O.Auth_OCSP_Info	T.Wrong_OCSP_Info
22	O.Current_OCSP_Info	T.Replay_OCSP_Info
23	O.User_Override_Time_OCSP	T.DOS_OCSP
10. CRL Validation Package		
#	**Objective**	**Threats**
24	O.Accurate_Rev_Info	T.Wrong_Revoc_Info_CRL
25	O.Auth_Rev_Info	T.Wrong_Revoc_Info_CRL

26	O.Current_Rev_Info	T.Replay_Revoc_Info_CRL
27	O.User_Override_Time_CRL	T.DOS_CRL

8.2 Security Requirements Rationale

In this section, the objectives are mapped to the functional requirements and rationale is provided for the selected EAL and its components and augmentation.

8.2.1 Functional Security Requirements Rationale

The mapping of all security objectives to functional requirements (components) or to assumptions is provided in Table 8.5 below. Rationale for the base TOE security functional requirements mapping and for each package are described in separate subsections following Table 8.5.

Explicitly stated security functional requirements are IT processing oriented security requirements. These requirements are similar in nature to the security functional requirements in the Common Criteria Part 2. Thus, security assurance requirements from the Common Criteria Part 3 can be used to test the explicitly stated requirements also; no additional assurance requirements beyond those taken from the Common Criteria Part 3 are required.

Table 8.5 – Security Objective to Functional Component Mapping

#	Objective	Functional Components
Mapping for Objectives for the TOE		
1E	OE.DAC	FDP_ACC.1, FDP_ACF.1
2E	OE.I&A	FIA_ATD.1, FIA_UAU.1, FIA_UID.1
3E	OE.Init_Secure_Attr	FMT_MSA.3
4E	OE.Invoke	FPT_RVM.1
5E	OE.Limit_Actions_Auth	FIA_UAU.1, FIA_UID.1
6E	OE.Limit_Tries	FIA_AFL.1
7E	OE.No_Echo	FIA_UAU.7
8E	OE.Protect_I&A_Data	FMT_MTD.1, FMT_SMF.1
9E	OE.Secure_Attributes	FMT_MSA.1, FMT_SMF.1
10E	OE.Security_Roles	FMT_SMR.2
11E	OE.Self_Protect	FPT_SEP.1
12E	OE.Trust_Anchor	FMT_MTD.1, FMT_SMF.1
13E	OE.TSF_Data	FMT_MTD.1, FMT_SMF.1

Table 8.5 (continued)

#	Objective	Functional Components
Mapping for Objectives for the Environment		
14E	OE.Authorized_Users	Defined in the Administrator and User Guides under AGD_ADM.1 and AGD_USR.1, respectively
15E	OE.Configuration	Defined in startup and installation guides under ADO_IGS.1
16E	OE.Crypto	FCS_CRM_FPS.1
17E	OE.Low	Defined in the SOF analysis and vulnerability assessment.
18E	OE.Physical_Security	Defined as part of the physical security policy in AGD_ADM.1 and AGD_USR.1
19E	OE.PKI_Info	FDP_ITC_PKI_INF.1
20E	OE.Time	FPT_STM.1
Mapping for 1. CPV – Basic Package		
1	O.Availability	FDP_DAU_CPV_CER.1
2	O.Correct_Temporal	FDP_DAU_CPV_INI.1
3	O.Current_Certificate	FDP_DAU_CPV_CER.1
3	O.Get_KeyInfo	FDP_DAU_CPV_OUT.1
5	O.Path_Find	FDP_CPD.1
6	O.Trusted_Keys	FDP_DAU_CPV_INI.1
7	O.User	FDP_DAU_CPV_CER.2
8	O.Verified_Certificate	FDP_DAU_CPV_CER.1
9	O.Valid_Certificate	FDP_DAU_CPV_CER.1
Mapping for 2. CPV – Basic Policy Package		
10	O.Provide_Policy_Info	FDP_DAU_CPV_INI.2, FDP_DAU_CPV_OUT.2
Mapping for 3. CPV – Policy Mapping Package		
11	O.Map_Policies	FDP_DAU_CPV_INI.3, FDP_DAU_CPV_CER.3, FDP_DAU_CPV_OUT.3
12	O.Policy_Enforce	FDP_DAU_CPV_INI.3, FDP_DAU_CPV_CER.3, FDP_DAU_CPV_OUT.3
Mapping for 4. CPV – Name Constraints Package		
13	O.Authorised_Names	FDP_DAU_CPV_INI.4, FDP_DAU_CPV_CER.4, FDP_DAU_CPV_CER.5
Mapping for 5. PKI Signature Generation Package		
14	O.Give_Sig_Hints	FDP_ETC_SIG.1
Mapping for 6. PKI Signature Verification Package		
15	O.Use_Sig_Hints	FDP_ITC_SIG.1,
16	O.Linkage_Sig_Ver	FDP_DAU_SIG.1

Table 8.5 (concluded)

#	Objective	Functional Components
Mapping for 7. PKI Encryption using Key Transfer Algorithms Package		
17	O.Hints_Enc_WO	FDP_ETC_ENC.1
18	O.Linkage_Enc_WO	FDP_ETC_ENC.1, FDP_DAU_ENC.1
Mapping for 8. PKI Decryption using Key Transfer Algorithms Package		
19	O.Correct_KT	FDP_ITC_ENC.1
Mapping for 9. OCSP Client Package		
20	O.Accurate_OCSP_Info	FDP_DAU_OCS.1
21	O.Auth_OCSP_Info	FDP_DAU_OCS.1
22	O.Current_OCSP_Info	FDP_DAU_OCS.1
23	O.User_Override_Time_OCSP	FDP_DAU_OCS.1
Mapping for 10. Certificate Revocation List (CRL) Validation Package		
24	O.Accurate_Rev_Info	FDP_DAU_CRL.1
25	O.Auth_Rev_Info	FDP_DAU_CRL.1
26	O.Current_Rev_Info	FDP_DAU_CRL.1
27	O.User_Override_Time_CRL	FDP_DAU_CRL.1

8.2.1.1 Security Objectives for the TOE Rationale

OE.DAC states that the TSF shall control and restrict user access to the TOE assets in accordance with a specified access control policy. This security objective is met by:

- **FDP_ACC.1**, Subset access control – PKI Credential Management, which requires that the TSF shall enforce the PKI credential management SFP on subjects, objects and operations assigned by the ST author. The terms object and subject refer to generic elements in the TOE. For a policy to be implemented, these entities will be identified by the ST author. For most systems there is only one type of subject, usually called a process or task, which needs to be specified in the ST. The ST author must specify the list of subjects, objects, and operations among subjects and objects covered by the SFP. This requirement calls for the specification of an access control policy
- **FDP_ACF.1**, Security attribute based access control – PKI Credential Management, which requires that the TSF shall enforce the PKI credential management SFP access control policy to objects. This requirement calls for the definition and enforcement of the policy specified in FDP_ACC.1.

OE.I&A states that the TSF shall uniquely identify all users, and shall authenticate the claimed identify before granting a user access to the TOE facilities. This security objective is met by:

- **FIA_ATD.1**, User attribute definition, which requires that the TSF shall maintain the roles for individual users. This requirement ensures that all users are identified with a role or roles that provide certain permissions and access.
- **FIA_UAU.1**, Timing of authentication, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is authenticated and that TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are authenticated.
- **FIA_UID.1**, Timing of identification, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is identified and that TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. This requirement ensures that all users are identified.

OE.Init_Secure_Attr states that the TSF shall provide valid default security attributes when an object is initialized. This security objective is met by:

- **FMT_MSA.3**, Static attribute initialisation, which requires that the TSF shall enforce the PKI credential management SFP to provide specific default values for security attributes that are used to enforce the SFP. The TSF shall allow the roles specified by the ST author to specify alternative initial values to override the default values when an object or information is created. This requirement ensures that valid default security attributes are specified when an object is created.

OE.Invoke states that the TSF shall be invoked for all actions. This security objective is met by:

- **FPT_RVM.1**, Non-bypassability of the TSP, which requires that the TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSF Scope of Control (TSC) is allowed to proceed. This requirement ensures that the TSF is invoked for all actions.

OE.Limit_Actions_Auth states that the TSF shall restrict the actions a user may perform before the TSF verifies the identity of the user. This security objective is met by:

- **FIA_UAU.1**, Timing of authentication, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is authenticated and that TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. This requirement restricts the actions that a user may perform before the user is authenticated.
- **FIA_UID.1**, Timing of identification, which requires that the TSF allow the a list of TSF mediated actions, specified by the ST author, to be performed on behalf of the user before the user is identified and that TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. This requirement restricts the actions that a user may perform before that user is identified.

OE.Limit_Tries states that the TSF shall restrict the number of consecutive unsuccessful authentication attempts. This security objective is met by:

- **FIA_AFL.1**, Authentication failure handling, which requires that the TSF shall detect when a number selected by the ST author of unsuccessful authentication attempts occur related to authentication events specified by the ST author. When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall perform actions specified by the ST author. This requirement restricts the number of consecutive unsuccessful authentication attempts.

OE.No_Echo states that the TSF shall not echo the authentication information. This security objective is met by:

- **FIA_UAU.7**, Protected authentication feedback, which requires that the TSF shall provide only the list of feedback specified by the ST author to the user while the authentication is in progress. This requirement ensures that the TSF shall not echo the authentication information.

OE.Protect_I&A_Data states that the TSF shall permit only authorized users to change the I&A data. This security objective is met by:

- **FMT_MTD.1**, Management of TSF data, which requires that the TSF shall restrict the ability to perform operations specified by the ST author on TSF data specified by the ST author to roles specified by the ST author. This requirement ensures that authorized users and their actions are defined for specified TSF data, including identification and authentication data.
- **FMT_SMF.1**, Specification of management functions, which requires the TSF to be able to perform security management functions.

OE.Secure_Attributes states that the TSF shall permit only the authorized users to change the security attributes. This security objective is met by:

- **FMT_MSA.1**, Management of security attributes, which requires that the TSF shall enforce the PKI credential management SFP to restrict the ability to perform operations specified by the ST author on the security attributes specified by the ST author to roles specified by the ST author. This requirement ensures that only authorized users, i.e., those with the appropriate role, are permitted to change specified security attributes.

- **FMT_SMF.1**, Specification of management functions, which requires the TSF to be able to perform security management functions.

OE.Security_Roles states that the TSF shall maintain security-relevant roles and association of users with those roles. This security objective is met by:

- **FMT_SMR.2**, Restrictions on security roles, which ensures that roles are identified and that all users are associated with a role.

OE.Self_Protect states that the TSF will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure. This security objective is met by:

- **FPT_SEP.1**, TSF domain separation, which requires that the TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects and that the TSF shall enforce separation between the security domains of subjects in the TSC.

OE.Trust_Anchor states that the TSF shall permit only authorized users to manage the trust anchors. This security objective is met by:

- **FMT_MTD.1**, Management of TSF data, which requires that the TSF shall restrict the ability to perform operations specified by the ST author on TSF data specified by the ST author to roles specified by the ST author. This requirement ensures that authorized users and their actions are defined for specified TSF data, including trust anchors.
- **FMT_SMF.1**, Specification of management functions, which requires the TSF to be able to perform security management functions.

OE.TSF_Data states that the TSF shall permit only authorized users to modify the TSF data. This security objective is met by:

- **FMT_MTD.1**, Management of TSF data, which requires that the TSF shall restrict the ability to perform operations specified by the ST author on TSF data specified by the ST author to roles specified by the ST author. This requirement ensures that authorized users and their actions are defined for specified TSF data.
- **FMT_SMF.1**, Specification of management functions, which requires the TSF to be able to perform security management functions.

8.2.1.2 Security Objectives for the Environment Rationale

Security Objectives for the Environment are met through a set of assumptions, as defined in Section 3.1 of this PP, and related objectives and requirements. In all cases, assumptions are made about functionality that will be provided by the environment to meet the environment objectives. Specific rationale for each environmental objective is as follows.

OE.Authorized_Users states that authorized users are trusted to perform their authorized tasks. This environmental security objective covers AE.Authorized_Users, an assumption that states that the Authorized users are trusted to perform their assigned functions. This environmental security objective and assumption are also supported by:

- The Administrator and User Guides as defined under assurance requirements **AGD_ADM.1** and **AGD_USR.1**, respectively.

OE.Configuration states that the TOE shall be installed and configured properly for starting up the TOE in a secure state. This objective covers AE.Configuration, an assumption that states that the TOE will be properly installed and configured. This environmental security objective and assumption are also supported by:

- The startup and installation guides required by the **ADO_IGS.1** assurance requirement, which states that accurate installation and configuration documentation must be provided that allows the TOE to be properly (i.e., in a secure state) installed and configured.

OE.Crypto states that the environment shall include one or more cryptographic modules that are validated at FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, HMAC and/or other required cryptographic functions. In summary, all cryptographic modules within the TOE shall be FIPS 140 series Level 1 validated. This objective is met by AE.Crypto_Module, an assumption that states that the TOE environment is assumed to include one or more cryptographic module(s) that are all validated at FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation, HMAC and/or other required cryptographic functions. In summary, all cryptographic modules in the TOE shall be validated at FIPS 140 series Level 1. This environmental security objective is met by:

- **FCS_CRM_FPS.1**, FIPS compliant cryptographic module, which requires that the IT Environment shall provide all cryptographic modules necessary for the TSF and that each cryptographic module shall be FIPS 140 series Level 1 validated.

OE.Low states that the identification and authentication functions in the TOE shall be designed and implemented for a minimum attack potential of low as validated by the vulnerability assessment and strength of function analyses. This environmental security objective covers the SOF analysis, which analyzes the strength of function of identification and authentication functions.

OE.Physical_Security states that the environment shall provide an acceptable level of physical security so that the TOE cannot be tampered with or be subject to side channel attacks such as the various forms of power analysis and timing analysis. This environmental security objective covers AE.Physical_Protection, an assumption that states that the physical protection is assumed to be provided by the environment. The TOE hardware and software is assumed to be protected from unauthorized physical access. This environmental security objective and assumption are also supported by:

- The Administrator and User Guides as defined under assurance requirements **AGD_ADM.1** and **AGD_USR.1**, respectively. The Administrator and User Guides define the security policy for the installation and operation of the TOE.

OE.PKI_Info states that the IT environment shall provide the TOE certificate and certificate revocation information for the TOI. This environmental security objective is met by:

- **FDP_ITC_PKI_INF.1**, Import of PKI information from outside the TSF, which requires that the IT environment shall make certificates, CRLs, and OCSP responses available to the TOE upon request.

OE.Time states that the environment shall provide access to accurate current time with required precision, translated to GMT. This objective covers AE.Time, an assumption that states that accurate system time with required precision in GMT format is assumed to be provided by the environment. This environmental security objective is met by:

- **FPT_STM.1**, Reliable time stamps, which requires that the IT Environment shall be able to provide reliable time stamps for TSF use.

8.2.1.3 Certification Path Validation – Basic Package Rationale

O.Availability states that the TSF shall continue to provide security services even if revocation information is not available. This objective is met by:

- **FDP_DAU_CPV_CER.1**, Certificate processing – basic, which requires that the TSF bypass the revocation check if the revocation information is not available.

O.Correct_Temporal states that the TSF shall provide accurate temporal validation results. This objective is met by:

- **FDP_DAU_CPV_INI.1**, Certification path initialisation – basic, which requires that the TSF obtain the time of interest called "TOI" from a reliable source.

O.Current_Certificate states that the TSF shall only accept certificates that are not expired as of "TOI". This objective is met by:

- **FDP_DAU_CPV_CER.1**, Certificate Processing – basic , which requires that the TSF accept a certificate only if the specified checks succeed, including that the certificate is not expired.

O.Get_KeyInfo states that the TSF shall provide the user public key and related information in order to carry out cryptographic functions. This objective is met by:

- **FDP_DAU_CPV_OUT.1**, Certification path output – basic, which requires that the TSF output the subject public key from the certification path and other information specified by the ST author.

O.Path_Find states that the TSF shall be able to find a certification path from a trust anchor to the subscriber. This objective is met by:

- **FDP_CPD.1**, Certification path development, which requires that the TSF shall develop a certification path from a trust anchor to the subscriber.

O.Trusted_Keys states that the TSF shall use trusted public keys in certification path validation. This objective is met by:

- **FDP_DAU_CPV_INI.1**, Certification path initialisation -- basic, which requires that the TSF use trusted public keys in the certification path validation.

O.User states that the TSF shall only accept certificates issued by a CA. This objective is met by:

- **FDP_DAU_CPV_CER.2**, Intermediate certificate processing – basic, which requires that the TSF accept an intermediate certificate only the certificate is issued by a CA.

O.Verified_Certificate states that the TSF shall only accept certificates with verifiable signatures. This objective is met by:

- **FDP_DAU_CPV_CER.1**, Certificate processing – basic, which requires that the TSF accept certificates only with verifiable signatures from which a path to a trust anchor can be found and validated.

O.Valid_Certificate states that the TSF shall use certificates that are valid, i.e., not revoked. This objective is met by:

- **FDP_DAU_CPV_CER.1**, Certificate processing – basic, which requires that that the TSF shall use only those certificates that are valid, i.e., revocation status determination demonstrates that the certificate is not revoked.

8.2.1.4 Certification Path Validation – Basic Policy Package Rationale

O.Provide_Policy_Info states that the TSF shall provide certificate policies for which the certification path is valid. This objective is met by:

- **FDP_DAU_CPV_INI.2**, Certification path initialisation – basic policy, which requires that The TSF shall use the initial-certificate-policies provided by user roles specified by the ST author.
- **FDP_DAU_CPV_OUT.2**, Certification path output – basic policy, which requires that the TSF shall output the certificate policies using the following rule: intersection of certificatePolicies extensions in all the certificates in certification path and initial-certificate-policies.

8.2.1.5 Certification Path Validation – Policy Mapping Package Rationale

O.Map_Policies states that the TSF shall map certificate policies in accordance with user and CA constraints. This objective is met by:

- **FDP_DAU_CPV_INI.3**, Certification path initialisation – policy mapping, which requires that the TSF use the explicit-policy-indicator, policy-mapping-inhibit-indicator, inhibit-any-policy-indicator provided by a role specified by the ST author.
- **FDP_DAU_CPV_CER.3**, Intermediate certificate processing – policy mapping, which requires that the TSF use the intermediate certificate to update specified state variables.
- **FDP_DAU_CPV_OUT.3**, Certification path output – policy mapping, which requires that the TSF shall map policies in the calculation of the policies intersection according to defined user and CA constraints.

O.Policy_Enforce states that the TSF shall validate a certification path in accordance with certificate policies acceptable to the user. This objective is met by:

- **FDP_DAU_CPV_INI.3**, Certification path initialisation – policy mapping, which requires that the TSF use the explicit-policy-indicator, policy-mapping-inhibit-indicator, inhibit-any-policy-indicator provided by a role specified by the ST author.
- **FDP_DAU_CPV_CER.3**, Intermediate certificate processing – policy mapping, which requires that the TSF use the intermediate certificate to update specified state variables.
- **FDP_DAU_CPV_OUT.3**, Certification path output – policy mapping, which requires that the TSF shall map policies in the calculation of the policies intersection according to defined user and CA constraints and that specified policies be enforced.

8.2.1.6 Certification Path Validation – Name Constraints Package Rationale

O.Authorised_Names states that the TSF shall validate a certificate only if the CA is authorized to issue a certificate to the subject. This objective is met by:

- **FDP_DAU_CPV_INI.4**, Certification path initialisation – names, which requires that the TSF initialize the following: permitted-subtrees = ∞, excluded-subtrees = ∅
- **FDP_DAU_CPV_CER.4**, Intermediate certificate processing – name constraints, which requires that the TSF accept a certificate only if the conditions specified by the requirement is satisfied.

- **FDP_DAU_CPV_CER.5**, Intermediate Certificate processing – name constraints, states that the TSF shall use the intermediate certificate to update the following states: permitted-subtrees and excluded-subtrees

8.2.1.7 PKI Signature Generation Package Rationale

O.Give_Sig_Hints states that the TSF shall provide hints for selecting correct certificates or keys for PKI signature. This objective is met by:

- **FDP_ETC_SIG.1**, Export of PKI Signature, which requires that the TSF to use the private key to perform digital signature and that the TSF include additional information specified by the ST author with the digital signature.

8.2.1.8 PKI Signature Verification Package Rationale

O.Use_Sig_Hints states that the TSF shall use hints for selecting correct certificates for signature verification. This objective is met by:

- **FDP_ITC_SIG.1**, Import of PKI Signature, which requires that the TSF use the following information from the signed data: signature algorithm, signer public key certificate, signer DN, signer subject alternative name, signer subject key identifier, or other data during signature verification.

O.Linkage_Sig_Ver states that the TSF shall use the correct user public key for signature verification. This objective is met by:

FDP_DAU_SIG.1, Signature Blob Verification, which requires that the TSF invoke a cryptographic module with the following information from Certification Path Validation to verify digital signature on signed data: subject public key algorithm, subject public key, subject public key parameters and that the TSF perform other verification checks as specified by the ST author.

8.2.1.9 PKI Encryption using Key Transfer Algorithms Package Rationale

O.Hints_Enc_WO states that the TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer algorithms. This objective is met by:

FDP_ETC_ENC.1, Export of PKI Encryption – Key Transfer Algorithms, which requires that the TSF include the information with the encrypted data, such as the key identifier, as selected or assigned by the ST author and that the TSF invoke a cryptographic module with the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.

O.Linkage_Enc_WO states that the TSF shall use the correct user public key for key transfer.

FDP_ETC_ENC.1, Export of PKI Encryption – Key Transfer Algorithms, which requires that the TSF invoke a cryptographic module with the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, subject public key parameters.

- **FDP_DAU_ENC.1**, PKI Encryption Verification – Key Transfer, which requires that the TSF apply verification checks for key transfer as selected by the ST author.

8.2.1.10 PKI Decryption using Key Transfer Algorithms Package Rationale

O.Correct_KT states that the TSF shall use appropriate private key and key transfer algorithm:

> **FDP_ITC_ENC.1**, Import of PKI Encryption – Key Transfer Algorithms, which requires that the TSF invoke a cryptographic module with the information from the encrypted data as selected by the ST author to provide a means to identify an appropriate private key and key transfer algorithm.

8.2.1.11 Online Certificate Status Protocol Package Rationale

O.Accurate_OCSP_Info states that the TSF shall accept only accurate OCSP responses. This objective is met by:

- **FDP_DAU_OCS.1**, Basic OCSP Client, which requires that only accurate revocation information be accepted from the OCSP responder.

O.Auth_OCSP_Info states that the TSF shall accept the OCSP responses from an authorized source. This objective is met by:

- **FDP_DAU_OCS.1**, Basic OCSP Client, which requires that the OCSP responder be verified as an authorized source.

O.Current_OCSP_Info states that the TSF may accept only OCSP responses current as of TOI. This objective is met by:

- **FDP_DAU_OCS.1**, Basic OCSP Client, which requires that only reasonably current as of TOI revocation information may be accepted through a series of policy and parameter checks.

O.User_Override_Time_OCSP states that the TSF shall permit the user to override the time checks on the OCSP response. This objective is met by:

- **FDP_DAU_OCS.1**, Basic OCSP Client, which requires that a role or roles specified by the ST author be able to override the time checks on the OCSP response.

8.2.1.12 Certificate Revocation List (CRL) Validation Package Rationale

O.Accurate_Rev_Info states that the TSF shall accept only accurate revocation information. This objective is met by:

- **FDP_DAU_CRL.1**, Basic CRL checking, which requires that the TSF accept accurate revocation information. Accuracy is determined through a series of verification and policy requirements within this Part 2 extended requirement.

O.Auth_Rev_Info states that the the TSF shall accept the revocation information from an authorized source for CRL. This objective is met by:

- **FDP_DAU_CRL.1**, Basic CRL checking, which requires that the TSF accept revocation information from an authorized source as selected or assigned by the ST author.

O.Current_Rev_Info states that the TSF shall accept only CRL that are current as of TOI. This objective is met by:

- **FDP_DAU_CRL.1**, Basic CRL checking, which requires that the TSF accept only reasonably current as of TOI revocation information through a series of policy requirements defined in FDP_DAU_CRL.1.

O.User_Override_Time_CRL states that the TSF shall permit the user to override the time checks on the CRL. This objective is met by:

- **FDP_DAU_CRL.1**, Basic CRL checking, which requires that the TSF accept the CRL as current if a role assigned by the ST author overrides time checks.

8.2.2 Assurance Requirement Rationale

EAL4 provides assurance by an analysis of security functions, using a functional and complete interface specification, guidance documentation, the high-level and low-level design of the TOE, and a subset of the implementation, to understand the security behaviour. Assurance is additionally gained through an informal model of the TOE security policy. EAL4 represents a meaningful increase in assurance from EAL3 by requiring more design description, a subset of the implementation, and improved mechanisms and/or procedures that provide confidence that the TOE will not be tampered with during development or delivery. EAL4 is augmented with ALC_FLR.1 to track and correct the reported and found security flaws in the product.

8.2.3 Strength of Function Rationale

Since the TOE does not include probabilistic or permutational mechanisms, the SOF claim is not applicable.

Table B.4 from CEM Annex B

Range of Values	Resistant to attack with attack potential of:	SOF rating
<10	No rating	No rating
10 – 17	Low	Basic
18 – 24	Moderate	Medium
>25	High	High

8.2.4 Security Functional Requirements Dependencies Rationale

Table 8.6 – Functional Requirements Dependencies

#	**Requirement**	**Dependencies**
Base Functional Requirements		
2E	FDP_ACC.1	FDP_ACF.1
3E	FDP_ACF.1	FDP_ACC.1, FMT_MSA.3
5E	FIA_AFL.1	FIA_UAU.1
6E	FIA_ATD.1	None
7E	FIA_UAU.1	FIA_UID.1
8E	FIA_UAU.7	FIA_UAU.1
9E	FIA_UID.1	None
10E	FMT_MSA.1	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1
11E	FMT_MSA.3	FMT_MSA.1, FMT_SMR.1
12E	FMT_MTD.1	FMT_SMF.1, FMT_SMR.1
13E	FMT_SMF.1	None

14E	FMT_SMR.2	FIA_UID.1
15E	FPT_RVM.1	None
16E	FPT_SEP.1	None
IT Environment Functional Requirements		
1E	FCS_CRM_FPS.1	None
4E	FDP_ITC_PKI_INF.1	None
17E	FPT_STM.1	None
1. CPV – Basic Package		
1	FDP_CPD.1	None
2	FDP_DAU_CPV_INI.1	FCS_COP.1 (See Note 4) FPT_STM.1 (See Note 1)
3	FDP_DAU_CPV_CER.1	FCS_COP.1 (See Note 4) FPT_STM.1 (See Note 1)
4	FDP_DAU_CPV_CER.2	FDP_DAU_CPV_CER.1
5	FDP_DAU_CPV_OUT.1	None
2. CPV – Basic Policy Package		
6	FDP_DAU_CPV_INI.2	FDP_DAU_CPV_INI.1 (See Note 5)
7	FDP_DAU_CPV_OUT.2	FDP_DAU_CPV_OUT.1 (See Note 5)
3. CPV – Policy Mapping Package		
8	FDP_DAU_CPV_INI.3	FDP_DAU_CPV_INI.2 (See Note 6)
9	FDP_DAU_CPV_CER.3	FDP_DAU_CPV_CER.2 (See Note 7)
10	FDP_DAU_CPV_OUT.3	FDP_DAU_CPV_OUT.2 (See Note 6)
4. CPV – Name Constraints Package		
11	FDP_DAU_CPV_INI.4	FDP_DAU_CPV_INI.1 (See Note 5)
12	FDP_DAU_CPV_CER.4	FDP_DAU_CPV_CER.1 (See Note 5)
13	FDP_DAU_CPV_CER.5	FDP_DAU_CPV_CER.2 (See Note 5)
5. PKI Signature Generation Package		
14	FDP_ETC_SIG.1	FCS_COP.1 (See Note 4)
6. PKI Signature Verification Package		
15	FDP_ITC_SIG.1	None
16	FDP_DAU_SIG.1	FCS_COP.1 (See Note 4) FDP_DAU_CPV_OUT.1 (See Note 5)
7. PKI Encryption using Key Transfer Algorithms Package		
17	FDP_ETC_ENC.1	FCS_COP.1 (See Note 4) FDP_DAU_CPV_OUT.1 (See Note 5)
18	FDP_DAU_ENC.1	FDP_DAU_CPV_OUT.1 (See Note 5)
8. PKI Decryption using Key Transfer Algorithms Package		
19	FDP_ITC_ENC.1	FCS_CRM_FPS.1
9. OCSP Client Package		
20	FDP_DAU_OCS.1	FCS_COP.1 (See Note 4)

		FPT_STM.1 (See Note 1)
10. Certificate Revocation List (CRL) Validation Package		
21	FDP_DAU_CRL.1	FCS_CRM_FPS.1 (See Note 4) FPT_STM.1 (See Note 1)

Note 1: FPT_STM.1 dependency is satisfied by the FPT_STM.1 security requirement for the IT environment.

Note 2: FIA_UID.1 dependency is satisfied by the base TOE security functional requirements.

Note 3: FMT_MTD.1 dependency is satisfied by the base TOE security functional requirements.

Note 4: The FCS_COP.1 dependency is not added to the package since the cryptographic module that is part of the environmental assumption will provide cryptographic operations, including FCS_COP.1.

Note 5: The dependency is satisfied by including the CPV – Basic Package

Note 6: The dependency is satisfied by including the CPV – Basic Policy Package

Note 7: The dependency is satisfied by including the CPV – Basic Package and the CPV – Basic Policy Package

8.3 TOE Summary Specification Rationale

Table 8.7 – Mapping from SFR to IT Security Function

	Security Functional Requirement	IT Security Function
	1. CPV – Basic Package	
1	FDP_CPD.1	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
2	FDP_DAU_CPV_INI.1	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
3	FDP_DAU_CPV_CER.1	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
4	FDP_DAU_CPV_CER.2	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
5	FDP_DAU_CPV_OUT.1	Certification Path Processing, CRL

		Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
	2. CPV – Basic Policy Package	
6	FDP_DAU_CPV_INI.2	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
7	FDP_DAU_CPV_OUT.2	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the ST)
	3. CPV – Policy Mapping Package	
8	FDP_DAU_CPV_INI.3	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
9	FDP_DAU_CPV_CER.3	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
10	FDP_DAU_CPV_OUT.3	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
	4. CPV – Name Constraints Package	
11	FDP_DAU_CPV_INI.4	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
12	FDP_DAU_CPV_CER.4	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
13	FDP_DAU_CPV_CER.5	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
	5. PKI Signature Generation Package	
14	FDP_ETC_SIG.1	Signature Generation Functionality (See Section 6.2 in the TOE Summary Specification)
	6. PKI Signature Verification Package	
15	FDP_ITC_SIG.1	PKI Signature Verification Functionality (See Section 6.3 in the TOE Summary Specification)
16	FDP_DAU_SIG.1	PKI Signature Verification Functionality (See Section 6.3 in the TOE Summary Specification)

	7. PKI Encryption using Key Transfer Algorithms Package	
17	FDP_ETC_ENC.1	PKI Encryption using Key Transfer Algorithms Functionality (See Section 6.4 in the TOE Summary Specification)
18	FDP_DAU_ENC.1	PKI Encryption using Key Transfer Algorithms Functionality (See Section 6.4 in the TOE Summary Specification)
	8. PKI Decryption using Key Transfer Algorithms Package	
19	FDP_ITC_ENC.1	PKI Dencryption using Key Transfer Algorithms Functionality (See Section 6.5 in the TOE Summary Specification)
	9. OCSP Client Package	
20	FDP_DAU_OCS.1	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the ST)
	10. Certificate Revocation List (CRL) Validation Package	
21	FDP_DAU_CRL.1	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the ST)

References

Please see the Applicable documents subsection in Section 1 of this document

Glossary of Terms

Asymmetric Keys
A pair of keys generated together that have different values such that information encrypted with one key may be decrypted with the other key or the information digitally signed using one key can be verified using the other key. One of the keys called the private key cannot be derived from the other key called the public key without extensive computational complexity.

Certificate Revocation List (CRL)
A list of the certificates that relying parties should no longer use or trust because the certificates have been revoked. Normally, the CA that issued the certificates also issues the CRL. The CA may assign responsibility for issuing CRLs to another entity. The CRL is a data structure that the issuer digitally signed.

Digital Envelope
A collection that consists of data encrypted with a symmetric session key and the session key encrypted for each recipient using the recipient’s public key.

Digitally Signed Data
A collection of data (the signed data) and a value (the digital signature) computed from that data. The signature is the result of applying an asymmetric cryptographic algorithm to the data (or an intermediate value derived from the data). The collection may also include information to assist in authenticating the entity that signed the data.

Effective Date
The date when a digital signature was created. The date includes the calendar date and the time of day. The relying party has to have confidence in the accuracy of the effective date. The date may be either the actual date or a presumed date. The relying party may presume that the effective date is the date of receipt of the document. The relying party knows the signature had to occur prior to receipt.

Expired Certificate
A certificate with the **not after** component of its validity field having a value earlier than the current date. Certificates may or may not appear in CRLs issued after their expiration.

Hash Algorithm
An algorithm that maps variable length inputs into a fixed length output value known as the digest or hash. The algorithm is a many-to-one function; multiple inputs may result in the same output. However, discovering an input value that results in a desired or given output is computationally infeasible.

Key Pair
A set of two keys used in asymmetric cryptography. A key generation algorithm creates the keys.

Non-repudiation
The inability to deny performing an action. Non-repudiation is evidence of the identity of the signer of a message and message integrity, sufficient to prevent a party from successfully denying the origin, submission, or delivery of a message and the integrity of its contents.

Public Key Owner
The entity for whom the key pair was generated and who is responsible for the secrecy and protection of the private key. The owner is the same entity as the subscriber listed in a public key certificate containing the owner’s public key.

Path Processing
The means employed by a relying party to ensure that the certificates in a path leading from a relying party trust point to subscriber’s public key certificate, are all valid. The validation activity includes chaining the subscriber and issuer names, using the subject public key from the parent certificate to verify the signature on a certificate, applying constraints imposed by the various extensions in the certificate, verifying that none of the certificates have expired or been revoked, and other X.509 certification path validation rules.

Private Key
A number, known only to the particular entity, its owner (i.e., the owner keeps the key secret). Owners use private keys to compute signatures on data they send and to decrypt information sent to them.

Public Key Certificate
A digitally signed statement from one entity, the Certification Authority, binding the public key (and some other information) and the identity of the owner of the corresponding private key. The owner may be an individual, a system or device, an organization, or function.

Public Key Infrastructure
The resources (people, systems, processes, and procedures) that provide services to register and identify new certificate owners, retrieve certificates, and determine the current validity of certificates.

Public Key Owner
The entity for whom the key pair was generated and who is responsible for the secrecy and protection of the private key. The owner is the same entity as the subscriber associated with a certificate containing the owner's public key.

Public Key Technology
Techniques and methods to generate related but different (asymmetric) keys for encryption and decryption and to use the keys to provide security services for authentication, confidentiality, integrity, and non-repudiation. The owner retains and keeps secret one of the asymmetric keys, the private key, and openly distributes the other asymmetric key, the public key. Also See **Asymmetric Key**.

Public Key–Enabled Application
A software application that uses PK technology to: authenticate its users (people, systems, and devices), ensure information is not changed or modified either during transmission or storage, hold users responsible and accountable for their actions and representations (i.e., preventing subsequent denial of responsibility), or encrypt information between parties where prior arrangement is neither known nor practical. PK–enabled applications rely on a PKI to create certificates that correctly associate a public key with the name of the owner of the associated private key, to retrieve certificates, and to determine the current validity of certificates (e.g., obtain a Certificate Revocation List [CRL]).

Public Key
A number associated with a particular entity and intended to be known to everyone. A public key is used to verifies a signature from the entity and/or to encrypt information that only the entity can decrypt.

Relying Party
An entity or an organization that depends on a certificate (i.e., uses the public key in the certificate for digital signature and/or encryption) and its association of the subscriber's identity (i.e., subject name) and public key.

Revoked Certificate
A certificate that relying parties should not trust or use. The CA that issued the certificate (or some similar authority) may revoke the certificate when conditions warrant. Conditions that may warrant revocation include suspected or actual compromise of the key or departure of the subscriber from the organization. CRLs issued by the CA always include all revoked, unexpired certificates (see **Expired Certificate**). Optionally, the CA may include revoked, expired certificates.

Root Certificate
The certificate at the top of the certification authority hierarchy. The certificate is self-signed; that is, the certificate issuer and the subject are the same entity, the Root CA. The certificate is generally a trust point. Since self-signed certificates do not have any trust in them, the root certificate or any other self-signed certificate must be distributed using secure means.

Digital Signature (or Signature)
A value determined from first computing a hash of the data to be signed and then applying a cryptographic function (the signature algorithm) to a hash value using the private key of the signer.

Symmetric Key
A key that is used to both encrypt and decrypt information. Parties involved in using the key must

keep the key secret; anyone with knowledge of the key could either originate or view encrypted information.

Subscriber

The entity (e.g., an individual) that has possession of the private key corresponding to the public key in a certificate. The certificate's subject field names the subscriber.

Trust anchor

A certificate that a relying party directly trusts. The certificate may belong to either a CA or an end-entity. The certificate is trusted because the relying party obtained the certificate by reliable means outside of the PKI and believes that the certificate accurately binds the name of the subscribing entity and the entity's public key. If the trust point is a CA certificate, the relying party trusts any certificates the CA issues. This trust is transitive to the extent the X.509 certificate extensions permit; if the CA issues a certificate to another CA, the relying party also trusts the second CA if the X.509 path validation logic succeeds.

Trusted Third Party (TTP)

An entity that other entities believe reliable, trustworthy and beyond reproach for purposes of performing some service. The TTP generally has no bias and is neutral for purposes of performing the service.

Trusted Timestamp

A digitally signed collection or other means that provides proof that a document existed at a particular time. The collection includes the date and time and either the document or the hash of the document. Often a TTP provides a timestamp service.

Signature Verification

The process of verifying a signature that includes the following steps: 1. Certification path validation in order to establish trust in the signer public key, 2. Calculating the hash for the message to be verified, and 3. Using applicable cryptographic algorithm with the signer public key (from step 1), calculated hash (from step 2), and signature to determine if the signature is valid.

List of Acronyms

CA	Certification Authority
CAC	Common Access Card
CAPI	Microsoft Crypto API
CC	Common Criteria
CEM	Common Evaluation Methodology
CMS	Cryptographic Message Syntax protocol
CPV	Certification Path Validation
CRL	Certificate Revocation List
CRLDP	CRL Distribution Point
CSP	Cryptographic Service Provider
DES	Data Encryption Standard
DH	Diffie Hellman
DISA	Defense Information Systems Agency
DN	Distinguished Name
DoD	Department of Defense
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ECDH	Elliptic Curve Diffie Hellman
EFS	Encrypted File System
EKU	Extended Key Usage
FIPS	Federal Information Processing Standard
GMT	Greenwich Mean Time
HMAC	Hash based Message Authentication Code
IDP	Issuing Distribution Point
IEC	International Electrotechnical Committee
IETF	Internet Engineering Task Force
ISO	International Organisation for Standards
IT	Information Technology
JITC	Joint Interoperability Test Center
LDAP	Lightweight Directory Access Protocol

NSA	National Security Agency
OCSP	On-line Certificate Status Protocol
OS	Operating System
PKCS	Public Key Cryptography Standard
PKE	Public Key Enabled
PKEPP	Public Key Enabled (**PKE**) Protection Profile (**PP**)
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure Working Group -- **IETF**
PP	Protection Profile
RFC	Request for Comment
RSA	Rivest, Shamir, and Adelman
SCL	Smart Card Logon
SCVP	Simple Certificate Validation Protocol
SFP	Security Function Policy
SOF	Strength of Function
SSL	Secure Socket Layer
ST	Security Target
TAP	Trusted Archive Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSC	**TSF** Scope of Control
TSF	**TOE** Security Function
TSP	Time Stamp Protocol (Internet X.509 Public Key Infrastructure)
USMC	United States Marine Corps