Public Key Infrastructure Framework Version 2.1 (PKIFv2)

PKIFv2 Security Target

Version 1.8
Date: January 03, 2008

Prepared for: US Marine Corps

Authors
Jean Petty, CygnaCom Solutions, Inc
Peter Kukura, CygnaCom Solutions, Inc
Santosh Chokhani, Orion Security Solutions, Inc
Carl Wallace, CygnaCom Solutions, Inc
Armen Galustyan, CygnaCom Solutions, Inc

Foreword

This Security Target (ST) defines the PKI Framework Version 2.1(PKIFv2), 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, *U.S. Government Family of Protection Profiles Public Key-Enabled Applications for Basic Robustness Environments, Version 2.77:*

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.

Revision History

Version	Date	ST Author	Description
0.1	August, 18 2006	Armen Galustyan	Initial conversion from PKIFv1 ST to PKIFv2 ST
0.2	September 22 2006	Armen Galustyan	2 Minor corrections. Parameter change to one of the TSFI functions, version correction in section 1 and a removal of a red highlight.
0.3	September 25, 2006	Armen Galustyan	Modified the ST to align with the new version of PKEPP 2.75
0.3	September 29, 2006	Santosh Chokhani	Modified the ST to align with Basic Robustness PP.
0.4	October 11, 2006	Armen Galustyan	Addressed some initial lab comments.
0.5	November 02, 2006	Armen Galustyan	Moved document list from section 6.7 to section 1.3. Specified Operating Systems.
0.6	January 3, 2007	Armen Galustyan	Made various changes in response to comments from CCTL from 2006-12-20
0.7	January 10, 2007	Armen Galustyan	Additional changes bases on feedback from CCTL.
0.8	January 15, 2007	Armen Galustyan	More changes based on the review by CCTL from 1/15/07
0.9	January 22, 2007	Armen Galustyan	Added extended SARs to address requirements levied by the PP.
1.0	April 3, 2007	Armen Galustyan	Addressed validator comments raised during the VOR.
1.1	June 19, 2007	Armen Galustyan	Added an assumption AE.GOOD_USER based on the lab feedback.
1.2	September 26, 2007	Armen Galustyan	Addressed validator comments raised during the test VOR.
1.3	October 5, 2007	Armen Galustyan	Minor changes based on lab comments.
1.4	October 10, 2007	Armen Galustyan	Fixed inconsistencies in TSFI list in section 6 base on lab comments.
1.5	October 11, 2007	Armen Galustyan	Added more precise platform information.
1.6	October 18, 2007	Armen Galustyan	More minor changes to address latest issues raised by the lab.
1.7	November 16, 2007	Armen Galustyan	Addressed issues raised in the final VOR.
1.8	January 03, 2008	Armen Galustyan	Added a mention regarding Java and

			.Net wrappers.

Table of Contents

List of Tables

Page

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 2.1

ST Title: Public Key Infrastructure Framework Version 2.1 (PKIFv2) Security Target

ST Version Number: Version 1.8

ST Date: January 03, 2008

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

Assurance Level: EAL4, augmented with ALC_FLR.2, 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. PKIFv2 is a toolkit used by application developers to incorporate secure PKI functionality into an application. PKIFv2 provides application developers a set of extensible classes, packaged as a Windows dynamic link library (DLL) or a dynamically loaded shared library for Linux and Unix, 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 Standardization/International 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
- [CMPLAN] Public Key Infrastructure Framework Version 2.1(PKIFv2) Configuration Management Plan, Version 1.0, October 23, 2007.

  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 Version 2.1 (PKIFv2) Delivery, Installation, Generation and Start-up Procedures, Version 0.7, October 23, 2007.

  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 Version 2.1 (PKIFv2) Development Security, Development Tools, Version 0.6, June 22, 2007

  This document describes the development facility, personnel and environment security. The document also describes the tools used to develop the TOE.
- [HELP] Usage Guide, Version 2.1, September 2007

  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 Version 2.1 (PKIFv2) Internals, Version 0.1, September 14, 2006

  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 Version 2.1 (PKIFv2) Security Policy Model, Version 01, July 18, 2006.

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 Version 2.1 (PKIFv2) Life-Cycle Model, Version 0.1, July 18, 2006.

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

- [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 0.7, June 19, 2007.

  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 1.2, October 23, 2007

  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 0.5, October 19, 2007

  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.

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 2.3 (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 2.3, 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 Family of Protection Profiles Public Key-Enabled Applications for Basic Robustness Environments, Version 2.77 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.

2 TOE Description

2.1 Overview

PKIFv2 is a C++ software library designed to simplify the task of adding PKI support to applications. PKIFv2 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)

Figure 1 illustrates PKIFv2 architecture. PKIFv2 consists of two DLLs (Dynamic Link Library) PKIF.dll and PKIFCMS.dll or dynamically loaded shared libraries libPKIF.so and libPKIFCMS.so (for Linux and Unix platforms). PKIF.dll/libPKIF.so is responsible for certification path processing, OCSP client functionality, CRL functionality, symmetric key encryption and decryption, and ASN.1 encoding/ decoding. PKIFCMS.dll/libPKIFCMS.so is responsible for signature generation functionality, signature verification functionality, PKI encryption using key transfer algorithms functionality and PKI decryption using key transfer algorithms functionality. Green arrows indicate the security functions contained in each dll/library in the TOE. Black errors indicate possible API calls. The application that uses PKIFv2 is part of the IT environment; it will make API calls to PKIF.dll/libPKIF.so and/or PKIFCMS.dll/libPKIFCMS to obtain TOE security functionality. PKIF.dll/libPKIFCMS.so will make API calls to the IT environment consisting of OS, NSS to obtain necessary functionality like certificate and CRL storage, cryptography, LDAP and HTTP support. CAPI CSP and CAPI certificate and CRL store are part of the Windows operating system. Detailed descriptions of TOE security functions can be found in section 2.2.

Note, for cryptographic functions, cryptographic key lengths supported by PKIFv2 are not a function of the PKIFv2 DLL, but rather, are determined by the capabilities of the relevant CSP.

PKIFv2 also provides Java and .NET interfaces to access the PKIFv2 functionality. These interfaces, which do not enforce any security, are simply wrapper interfaces that allow convenient access to the full PKIF library.

Note, PKIFv2 is not source code but a compiled DLL or dynamically loaded shared library.

Note, the application using the TOE must satisfy the appropriate assumptions and organizational security policies of the IT environment imposed by the Security Target (e.g., application developers are trusted and follow guidance properly) when using the TOE in the evaluated configuration. The application using the TOE must also possess the appropriate level of robustness when operating the TOE as intended.

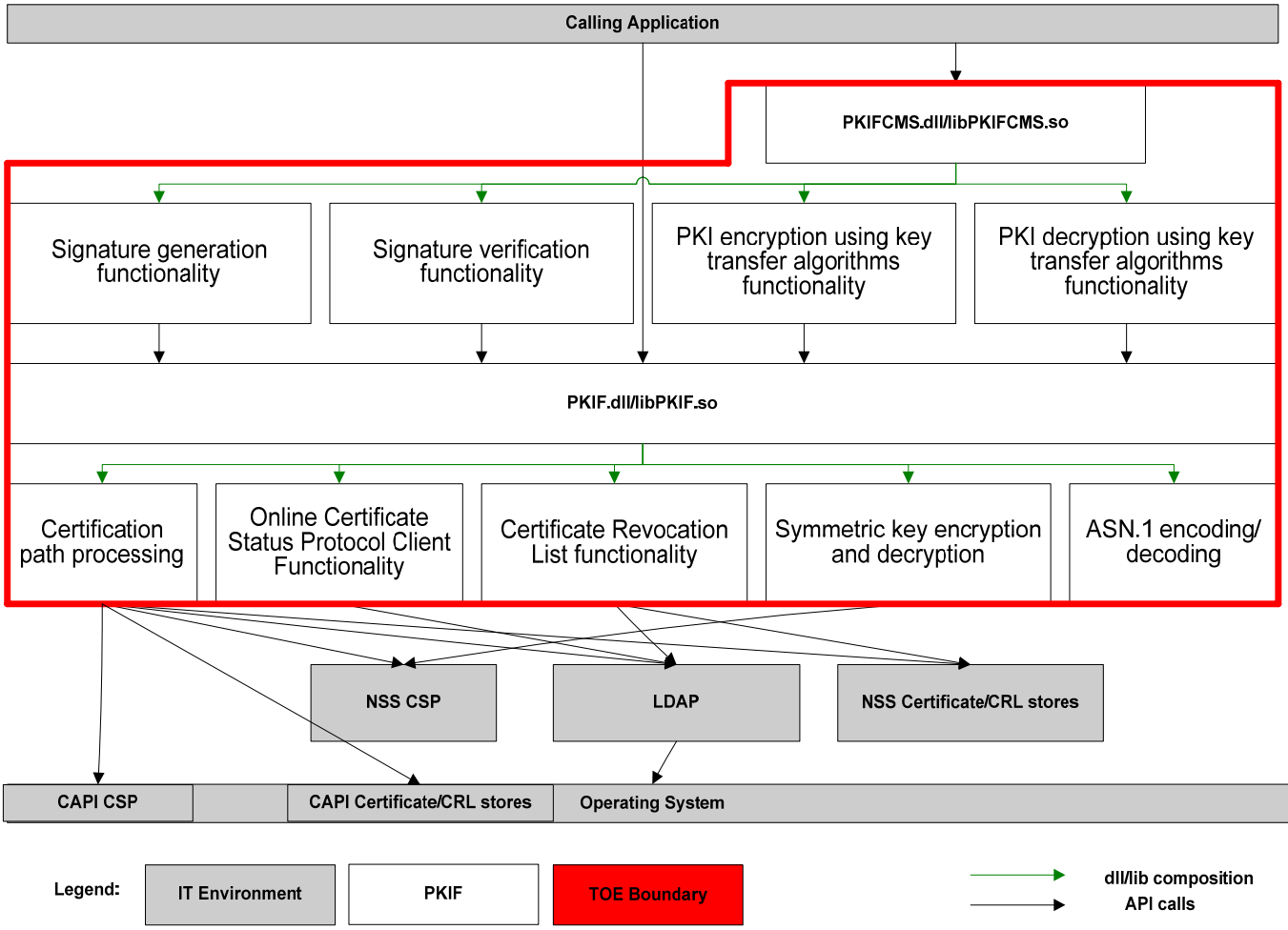

Figure 1: PKIFv2 Architecture

PKIFv2 includes following guidance documentation:

- PKIFv2 Usage Guide [HELP]
- Public Key Infrastructure Framework Version 2.1 (PKIFv2) Delivery, Installation, Generation and Start-up Procedures [DELIVERY]

Note, Only the downloadable version of the guidance is supported by the evaluation configuration of the TOE.

PKI Framework Version 2.1 can be obtained by the general public with full functionality, as evaluated, from http://pkif.sourceforge.net/. It can be used in wide range of applications for example to provide certificate path building and validation functionality to an IIS Plug-In, to provide OCSP support to CAPI OCSP PlugIn.

2.2 TOE Description

2.2.1 Certification Path Processing

PKIFv2 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. PKIFv2 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. This is an end-entity certificate (i.e., a certificate issued to an entity not functioning as a CA).

PKIFv2 processes the following security-related certificate extensions: ocsp-nocheck, keyUsage, extendedKeyUsage, and basicConstraints. PKIFv2 performs the processing of the following certificate policy-related extensions: certificatePolicies, policyMapping, inhibitAnyPolicy, policyConstraints, and nameConstraints extensions

By default, PKIFv2 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

PKIFv2 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

PKIFv2 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

PKIFv2 enables applications to perform public key encryption using key transfer algorithms such as RSA.

2.2.5 PKI Decryption using Key Transfer Algorithms Functionality

PKIFv2 enables applications to perform private key decryption using key transfer algorithms such as RSA.

2.2.6 Online Certificate Status Protocol Client Functionality

PKIFv2 can generate Online Certificate Status Protocol (OCSP) requests and validate OCSP responses to determine the revocation status of public key certificates. PKIFv2 verifies OCSP Responder as a trust anchor, as a CA, or as an end entity authorized to sign OCSP responses. PKIFv2 establishes trust in the OCSP responder certificates by performing Certification Path Validation.

2.2.7 Certificate Revocation List functionality

PKIFv2 provides Certificate Revocation List (CRL) validation functionality that enables applications to determine the revocation status of a certificate using a CRL. PKIFv2 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.

PKIFv2 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 PKIFv2 will develop and validate certification paths to CRL signers where necessary.

2.2.8 Symmetric key encryption and decryption

PKIFv2 provides functionality to perform symmetric key encryption and decryption using algorithms including DES, Triple DES, and AES. PKIFv2 itself does not perform any cryptographic operations, PKIFv2 relies on Microsoft CAPI when running on Windows operating system and on NSS when running on Linux operating system for that functionality.

2.2.9 ASN.1 encoding/decoding

PKIFv2 performs decoding of objects in support of processing related to X.509, RFC3280, OCSP and CMS. PKIFv2 performs encoding of objects in support of processing related to OCSP and CMS.

2.3 Roles, User Data, and TSF Data

PKIFv2 is a toolkit used by application developers to incorporate secure PKI functionality into an application; PKIFv2 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. State variables used by the TOE are not persistent and therefore are not considered TSF data. Thus, there are no TSF data in PKIF.

2.4 TOE Environment Description

PKIFv2 is intended for use with Microsoft Visual C++ .NET 2005 and GCC 4,0.

PKIFv2 is designed to operate with any CAPI- or PKCS11-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.

PKIFv2 is intended for use on Windows, Linux, and UNIX. PKIFv2 CCEVS evaluation will apply to the library hosted on 32-bit Microsoft Windows Server 2003; SP 1 running on Intel x86 architectures and 32-bit Red Hat Enterprise Linux 4 (RHEL) WS Update 1 running on Intel x86 architectures. Both operating system have obtained CCEVS certificate for the TOE assurance level EAL 4 augmented by ALC_FLR.3. Validation ID for Microsoft Windows Server 2003; SP 1 is 4025. Validation ID for Red Hat Enterprise Linux 4 (RHEL) WS Update 1 is 10072.

Windows, Linux and UNIX operating systems generally include LDAP client functionality, which PKIFv2 will use. Windows includes a CAPI-compatible FIPS 140 Level 1 validated cryptographic module. PKIFv2 on Linux and Unix will use Netscape Security Services, which provides FIPS 140 Level 1 validated cryptographic module.

PKIFv2 relies on the underlying operating system and application that uses the toolkit to provide protection for TOE security functions against attempts to bypass, corrupt or compromise. The operating system ensures that users cannot interfere with other users by implementing identification and authentication of users and restricting access to TSF (and user) data to authorized users.

Though PKIF is a shared library, its architecture ensures that no information is shared across multiple instances of the library at runtime. Each mapping of the library is assigned its own set of variables with no sharing across applications. The underlying operating system is required to provide protection against reuse of data that has been temporarily stored in shared memory, and to provide domain separation for the execution of the TSF.

As a software library, a developer is ultimately free to bypass functionality by not invoking the functionality or by ignoring the results of the function invocation. However, when used in

accordance with the usage documentation, PKIF security functions cannot be bypassed. As an additional means to determine any improper configuration of the underlying operating system's supporting security functionality or the TOE itself, the IT environment is required to provide auditing capabilities.

PKIFv2 does not require any additional privileges from the operating system.

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.

PKIFv2 will build and validate certification paths to any trust anchor. For example, in order to use PKIFv2 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 PKIFv2 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.

PKIFv2 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.

Identification and Authentication

Windows, Linux and Unix operating systems provide I&A. I&A is useful for access control of resources managed by the operating system, including files, folders, certificate stores, private keys, and audit logs (audit logs are maintained in a specific folder in the file system hierarchy). Windows, Linux, and Unix 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 Family of Protection Profiles Public Key-Enabled Applications for Basic Robustness Environments, Version 2.77 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: 2.77; Date: February 1, 2007; 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 PKIFv2 is EAL4 with augmentation; EAL4 was selected because PKIFv2 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.2 is added to provide basic flaw remediation.

3 TOE Security Environment

3.1 Relationship between Basic Robustness Level and the formation of applicable assumptions, threats and the policies of the TSE

Basic robustness TOEs falls in the upper left area of the robustness figures discussed in Appendix D. A Basic Robustness TOE is considered sufficient for low threat environments or where compromise of protected information will not have a significant impact on mission objectives. This implies that the motivation of the threat agents will be low in environments that are suitable for TOEs of this robustness. In general, basic robustness results in “good commercial practices” that counter threats based in casual and accidental disclosure or compromise of data protected by the TOE.

Threat agent motivation can be considered in a variety of ways. One possibility is that the value of the data process or protected by the TOE will generally be seen as of little value to the adversary (i.e., compromise will have little or no impact on mission objectives). Another possibility, (where higher value data is processed or protected by the TOE) is that procuring organizations will provide other controls or safeguards (i.e., controls that the TOE itself does not enforce) in the fielded system in order to increase the threat agent motivation level for compromise beyond a level of what is considered reasonable or expected to be applied.

3.2 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.Configuration	The TOE will be properly installed and configured.
2	AE.Low	The attack potential on the TOE is assumed to be low.
3	AE.NO_EVIL	Administrators are non-hostile, appropriately trained and follow all administrator guidance.
4	AE.PHYSICAL	It is assumed that the environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.
5	AE.GOOD_USER	TOE users are non-hostile and will follow all user guidance.

3.3 Threat Agent Characterization

In addition to helping define the robustness appropriate for a given environment, the threat agent is a key component of the formal threat statements in the ST. Threat agents are typically characterized by a number of factors such as *expertise*, *available resources*, and *motivation*. Because each robustness level is associated with a variety of environments, there are corresponding varieties of specific threat agents (that is, the threat agents will have different combinations of motivation, expertise, and available resources) that are valid for a given level of robustness. The following discussion explores the impact of each of the

threat agent factors on the ability of the TOE to protect itself (that is, the robustness required of the TOE).

The *motivation* of the threat agent seems to be the primary factor of the three characteristics of threat agents outlined above. Given the same expertise and set of resources, an attacker with low motivation may not be as likely to attempt to compromise the TOE. For example, an entity with no authorization to low value data none-the-less has low motivation to compromise the data; thus a basic robustness TOE should offer sufficient protection. Likewise, the fully authorized user with access to highly valued data similarly has low motivation to attempt to compromise the data, thus again a basic robustness TOE should be sufficient.

Unlike the motivation factor, however, the same can't be said for *expertise*. A threat agent with low motivation and low expertise is just as unlikely to attempt to compromise a TOE as an attacker with low motivation and high expertise; this is because the attacker with high expertise does not have the motivation to compromise the TOE even though they may have the expertise to do so. The same argument can be made for *resources* as well

Therefore, when assessing the robustness needed for a TOE, the motivation of threat agents should be considered a “high water mark”. That is, the robustness of the TOE should increase as the motivation of the threat agents increases.

Having said that, the relationship between expertise and resources is somewhat more complicated. In general, if resources include factors other than just raw processing power (money, for example), then expertise should be considered to be at the same “level” (low, medium, high, for example) as the resources because money can be used to purchase expertise. Expertise in some ways is different, because expertise in and of itself does not automatically procure resources. However, it may be plausible that someone with high expertise can procure the requisite amount of resources by virtue of that expertise (for example, hacking into a bank to obtain money in order to obtain other resources).

It may not make sense to distinguish between these two factors; in general, it appears that the only effect these may have is to lower the robustness requirements. For instance, suppose an organization determines that, because of the value of the resources processed by the TOE and the trustworthiness of the entities that can access the TOE, the motivation of those entities would be “medium”. This normally indicates that a medium robustness TOE would be required because the likelihood that those entities would attempt to compromise the TOE to get at those resources is in the “medium” range. However, now suppose the organization determines that the entities (threat agents) that are the least trustworthy have no resources and are unsophisticated. In this case, even though those threat agents have medium motivation, the likelihood that they would be able to mount a successful attack on the TOE would be low, and so a basic robustness TOE may be sufficient to counter that threat.

It should be clear from this discussion that there is no “cookbook” or mathematical answer to the question of how to specify exactly the level of motivation, the amount of resources, and the degree of expertise for a threat agent so that the robustness level of TOEs facing those threat agents can be rigorously determined. However, an organization can look at combinations of these factors and obtain a good understanding of the likelihood of a successful attack being attempted against the TOE. Each organization wishing to procure a TOE must look at the threat factors applicable to their environment; discuss the issues raised in the previous paragraph; consult with appropriate accreditation authorities for input; and document their decision regarding likely threat agents in their environment.

The important general points we can make are:

- The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE
- A threat agent's expertise and/or resources that is "lower" than the threat agent's motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however).
- The availability of attacks associated with high expertise and/or high availability of resources (for example, via the Internet or "hacker chat rooms") introduces a problem when trying to define the expertise of, or resources available to, a threat agent.

3.4 Threats to Security for the TOE

The following subsections define base threats and security threats for each of the packages defined. 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, 2) authorized users who want to exceed their authority, 3) unauthorized users with logical access to the TOE, 4) humans and systems with no access to the TOE, but with the ability to modify, insert, delete or eavesdrop user data that is protected using PKI based cryptography or 5) failure of the TOE.

The following are the base threats to the TOE, included in Table 3.2, below

Table 3.2 – Base Threats to Security

Threat Name	Threat Description
TE.AUDIT_COMPROMISE	A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action
TE.CHANGE_TIME	An unauthorized user may change the TSF notion of time resulting in accepting old revocation information or expired certificates.
TE.CRYPTO_COMPROMISE	A user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
TE.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
TE.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered

Threat Name	Threat Description
	thereby causing potential security vulnerabilities.
TE.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
TE.TSF_COMPROMISE	A user or process may cause, through an unsophisticated attack, TSF data, security attributes, or executable code to be inappropriately accessed (viewed, modified, or deleted).
TE.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
TE.UNAUTHORIZED_ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy.
TE.UNIDENTIFIED_ACTIONS	The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

3.5 Threats to Security for Packages

The following subsections define security threats for each of the packages used in the ST. 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.

3.5.1 Certification Path Validation – Basic Package

In addition to the base threats, the following threats are defined for the Certification Path Validation – Basic package.

Table 3.3 – Threats for the CPV – Basic Package

Threat Name	Threat Description
T.Certificate_Modi	An untrusted user may modify a certificate resulting in using a wrong public key.
T.DOS_CPV_Basic	The revocation information or access to revocation information could be made unavailable, resulting in loss of system availability.
T.Expired_Certificate	An expired (and possibly revoked) certificate as of TOI could be used for signature verification.

T.Untrusted_CA	An untrusted entity (Certification Authority (CA)) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users.
T.No_Crypto	The user public key and related information may not be available to carry out the cryptographic function.
T.Path_Not_Found	A valid certification path is not found due to lack of system functionality.
T.Revoked_Certificate	A revoked certificate could be used as valid, resulting in security compromise.
T.User_CA	A user could act as a CA, issuing unauthorized certificates.

3.5.2 Certification Path Validation – Basic Policy Package

The following threats are defined for the Certification Path Validation – Basic Policy package.

Table 3.4 – Threats for the CPV – Basic Policy Package

Threat Name	Threat Description
T.Unknown_Policies	The user may not know the policies under which a certificate was issued.

3.5.3 Certification Path Validation – Policy Mapping Package

The following threats are defined for the Certification Path Validation – Policy Mapping package..

Table 3.5 – Threats for the CPV – Policy Mapping Package

Threat Name	Threat Description
T.Mapping	The user may accept unacceptable certificates or reject acceptable certificates due to improper certificate policy mapping.
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.

3.5.4 Certification Path Validation – Name Constraints Package

The following threats are defined for the Certification Path Validation – Name Constraints Package.

Table 3.6 – Threats for the CPV – Name Constraints Package

Threat Name	Threat Description
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.

3.5.5 PKI Signature Generation Package

The following threats are defined for the PKI Signature Generation package.

Table 3.7 – Threats for the PKI Signature Generation Package

Threat Name	Threat Description
T.Clueless_PKI_Sig	The user may try only inappropriate certificates for signature verification because the signature does not include a hint.

3.5.6 PKI Signature Verification Package

The following threats are defined for the PKI Signature Verification Package.

Table 3.8 – Threats for the PKI Signature Verification Package

Threat Name	Threat Description
T.Assumed_Identity_PKI_Ver	A user may assume the identity of another user in order to verify a PKI signature.
T.Clueless_PKI_Ver	The user may try only inappropriate certificates for signature verification because hints in the signature are ignored.

3.5.7 PKI Encryption using Key Transfer Algorithms Package

The following threats are defined for the PKI Encryption using Key Transfer Algorithms Package.

Table 3.9 – Threats for the PKI Encryption using Key Transfer Algorithms Package

Threat Name	Threat Description
T.Assumed_Identity_WO_En	A user may assume the identity of another user in order to perform encryption using Key Transfer algorithms.

T.Clueless_WO_En	The user may try only inappropriate certificates for encryption using Key Transfer algorithms in absence of hint.

3.5.8 PKI Decryption using Key Transfer Algorithms Package

The following threats are defined for the PKI Decryption using Key Transfer Algorithms package.

Table 3.10 – Threats for the PKI Decryption using Key Transfer Algorithms Package

Threat Name	Threat Description
T.Garble_WO_De	The user may not apply the correct key transfer algorithm or private key, resulting in garbled data.

3.5.9 Online Certificate Status Protocol Client Package

The following threats are defined for Online Certificate Status Protocol Client package.

Table 3.11 – Threats for the OCSP Client Package

Threat Name	Threat Description
T.DOS_OCSP	The OCSP response or access to the OCSP response could be made unavailable, resulting in loss of system availability.
T.Replay_OCSP_Info	The user may accept an OCSP response from well before TOI resulting in accepting a revoked certificate.
T.Wrong_OCSP_Info	The user may accept a revoked certificate or reject a valid certificate due to a wrong OCSP response.

3.5.10 Certificate Revocation List (CRL) Validation Package

The following threats are defined for the Certificate Revocation List (CRL) Validation package.

Table 3.12 – Threats for the Certificate Revocation List (CRL) Validation Package

Threat Name	Threat Description
T.DOS_CRL	The CRL or access to CRL could be made unavailable, resulting in loss of system availability.
T.Replay_Revoc_Info_CRL	The user may accept a CRL issued well before TOI resulting in accepting a revoked certificate.

Threat Name	Threat Description
T.Wrong_Revoc_Info_CRL	The user may accept a revoked certificate or reject a valid certificate due to a wrong CRL.

3.6 Organizational Security Policies

The following table defines organizational security policies.

Table 3.13 – Organizational Security Policies

Policy Name	Policy Description
P.ACCESS_BANNER	The IT Environment shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.CRYPTOGRAPHY	Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).

4 Security Objectives

4.1 Security Objectives for the IT Environment

Security objectives for the IT Environment are defined in the table below.

There are five security objectives for the non-IT environment of the TOE: OE.Configuration, OE.NO_EVIL, OE.PHYSICAL, OE.Low, and OE.GOOD_USER. The remaining objectives are for the IT environment.

Table 4.1 – Security Objectives for the IT Environment

Objective Name	Objective Description
OE.AUDIT_GENERATION	The IT Environment will provide the capability to detect and create records of security-relevant events associated with users.
OE.AUDIT_PROTECTION	The IT Environment will provide the capability to protect audit information.
OE.AUDIT_REVIEW	The IT Environment will provide the capability to selectively view audit information,
OE.Configuration	The TOE will be installed and configured properly for starting up the TOE in a secure state.
OE.CORRECT_TSF_OPERATION	The IT environment will provide functionality to support the correct operation of the TSF. The IT Environment will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
OE.CRYPTOGRAPHY	The IT Environment will provide NIST FIPS 140-2 validated cryptographic services for the TOE
OE.DISPLAY_BANNER	The IT Environment will display an advisory warning regarding use of the TOE.
OE.Low	The Identification and Authentication functions in the IT Environment will be designed and implemented for a minimum attack potential of low as validated by the vulnerability assessment and Strength of Function analyses.
OE.MANAGE	The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
OE.MEDIATE	The IT Environment will protect user data in accordance with its security policy.
OE.NO_EVIL	Sites using the TOE will ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.PHYSICAL	The non-IT environment will 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.

Objective Name	Objective Description
OE.RESIDUAL_INFORMATION	The IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.
OE.SELF_PROTECTION	The IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure.
OE.TIME_STAMPS	The IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.
OE.TIME_TOE	The IT Environment will provide reliable time for the TOE use.
OE.TOE_ACCESS	The IT Environment will provide mechanisms that control a user's logical access to the TOE.
OE.TOE_PROTECTION	The IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification.
OE.GOOD_USER	Sites using the TOE will ensure that TOE users are non-hostile and follow all user guidance.

4.2 Security Objectives for the TOE

The following subsections define security objectives for each of the packages defined

4.2.1 Certification Path Validation – Basic Package

The following security objectives are defined for the Certification Path Validation – Basic Package.

Table 4.2 – Security Objectives for CPV – Basic Package

Objective Name	Objective Description
O.Availability	The TSF shall continue to provide security services even if revocation information is not available and user overrides revocation checking.
O.Correct_Temporal	The TSF shall provide accurate temporal validation results.
O.Current_Certificate	The TSF shall only accept certificates that are not expired as of TOI.
O.Get_KeyInfo	The TSF shall provide the user public key and related information in order to carry out cryptographic functions.

O.Path_Find	The TSF shall be able to find a certification path from a trust anchor to the subscriber.
O.Trusted_Keys	The TSF shall use trusted public keys in certification path validation.
O.User	The TSF shall only accept certificates issued by a CA.
O.Verified_Certificate	The TSF shall only accept certificates with verifiable signatures.
O.Valid_Certificate	The TSF shall use certificates that are valid, i.e., not revoked.

Objectives O.Availability and O.Valid_Certificate mitigate threats T.DOS_CPV_Basic and T.Revoked_Certificate, respectively. But these objectives cannot completely counter the threats simultaneously. The SFR FDP_DAU_CPV_(EXP).1.3 has been operated on so that user can determine whether to override the lack of availability of revocation information.

4.2.2 Certification Path Validation – Basic Policy Package

The following security objective is defined for the Certification Path Validation – Basic Policy package.

Table 4.3 – Security Objectives for CPV – Basic Policy Package

Objective Name	Objective Description
O.Provide_Policy_Info	The TSF shall provide certificate policies for which the certification path is valid.

4.2.3 Certification Path Validation – Policy Mapping Package

The following security objectives are defined for the Certification Path Validation – Policy Mapping package.

Table 4.4 – Security Objectives for CPV – Policy Mapping Package

Objective Name	Objective Description
O.Map_Policies	The TSF shall map certificate policies in accordance with user and CA constraints.
O.Policy_Enforce	The TSF shall validate a certification path in accordance with certificate policies acceptable to the user.

4.2.4 Certification Path Validation – Name Constraints Package

The following security objective is defined for the Certification Path Validation – Name Constraints package.

Table 4.5 – Security Objectives for CPV – Name Constraints Package

Objective Name	Objective Description
O.Authorised_Names	The TSF shall validate a certificate only if the CA is authorized to issue a certificate to the subject.

4.2.5 PKI Signature Generation Package

The following security objective is defined for the PKI Signature Generation package.

Table 4.6 – Security Objectives for PKI Signature Generation Package

Objective Name	Objective Description
O.Give_Sig_Hints	The TSF shall provide hints for selecting correct certificates for signature verification.

4.2.6 PKI Signature Verification Package

The following security objectives are defined for the PKI Signature Verification package.

Table 4.7 – Security Objectives for PKI Signature Verification Package

Objective Name	Objective Description
O.Use_Sig_Hints	The TSF shall use hints for selecting correct certificates for signature verification.
O.Linkage_Sig_Ver	The TSF shall use the correct user public key for signature verification.

4.2.7 PKI Encryption using Key Transfer Algorithms Package

The following security objectives are defined for the PKI Encryption using Key Transfer Algorithms package.

Table 4.8 – Security Objectives for PKI Encryption using Key Transfer Algorithms Package

Objective Name	Objective Description
O.Hints_Enc_WO	The TSF shall provide hints for selecting correct certificates or keys for PKI Encryption using Key Transfer Algorithms.
O.Linkage_Enc_WO	The TSF shall use the correct user public key for key transfer.

4.2.8 PKI Decryption using Key Transfer Algorithms Package

The following security objectives are defined for the PKI Decryption using Key Transfer Algorithms package.

Table 4.9 – Security Objectives for PKI Decryption using Key Transfer Algorithms Package

Objective Name	Objective Description
O.Correct_KT	The TSF shall use appropriate private key and key transfer algorithm.

4.2.9 Online Certificate Status Protocol Client Package

The following security objectives are defined for the Online Certificate Status Protocol Client package.

Table 4.10 – Security Objectives for Online Certificate Status Protocol Client Package

Objective Name	Objective Description
O.Accurate_OCSP_Info	The TSF shall accept only accurate OCSP responses.
O.Auth_OCSP_Info	The TSF shall accept the revocation information from an authorized source for OCSP transactions.
O.Current_OCSP_Info	The TSF accept only OCSP responses current as of TOI.
O.User_Override_Time_OCSP	The TSF shall permit the user to override the time checks on the OCSP response.

Objectives O.Current_OCSP_Info and O.User_Override_Time_OCSP mitigate threats T.Replay_OCSP_Info and T.DOS_OCSP, respectively. But these objectives cannot completely counter the threats simultaneously.

To fully mitigate the threat T.Replay_OCSP, the ST author has used request nonce as listed in the security functional requirements element FDP_DAU_OCS_(EXP).1.12.

To mitigate the threat T.DOS_OCSP, the ST author has used the operations on FDP_DAU_OCS_(EXP).1.9 security functional requirements element to ignore time checks when overridden by user. In addition, the user can accept a response past nextUpdate if it is not to beyond thisUpdate + x where x is defined by the user.

4.2.10 Certificate Revocation List (CRL) Validation Package

The following security objectives are defined for the Certificate Revocation List Validation Package.

Table 4.11 – Security Objectives for Certificate Revocation List (CRL) Validation Package

Objective Name	Objective Description

O.Accurate_Rev_Info	The TSF shall accept only accurate revocation information.
O.Auth_Rev_Info	The TSF shall accept the revocation information from an authorized source for CRL.
O.Current_Rev_Info	The TSF shall accept only CRL that are current as of TOI.
O.User_Override_Time_CRL	The TSF shall permit the user to override the time checks on the CRL.

Objectives O.Current_Rev_Info and O.User_Override_Time_CRL mitigate threats TT.Replay_Revoc_Info_CRL and T.DOS_CRL, respectively. But these objectives cannot completely counter the threats simultaneously. To mitigate the threat T.DOS_CRL, the ST author used the operations on FDP_DAU_CRL_(EXP).1.6 security functional requirements element to to ignore time checks when overridden by user. In addition, the user can accept a response past nextUpdate if it is not to beyond thisUpdate + x where x is defined by the user..

5 IT Security Requirements

This section defines the TOE security functional requirements and assurance requirements. Requirements are drawn from PP that is validated under the Common Criteria Evaluation and Validation Scheme, *U.S. Government Family of Protection Profiles Public Key-Enabled Applications for Basic Robustness Environments, Version 2.77* and CC Parts 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
FAU_GEN.1-NIAP-0407:1 & 2	Part 2 Extended
FAU_GEN.2-NIAP-0410: 1 & 2	Part 2 Extended
FAU_SAR.1	Part 2
FAU_SAR.2	Part 2
FAU_SAR.3	Part 2
FAU_SEL.1-NIAP-0407	Part 2 Extended
FAU_STG.1-NIAP-0429	Part 2 Extended
FAU_STG.NIAP-0429-1	Part 2 Extended
FCS_CRM_FPS_(EXP).1	Part 2 Extended
FDP_ACC.1	Part 2
FDP_ACF.1-NIAP-0407	Part 2 Extended
FDP_RIP.2	Part 2
FIA_AFL.1	Part 2
FIA_ATD.1	Part 2
FIA_UAU.2	Part 2
FIA_UAU.7	Part 2
FIA_UID.2	Part 2
FIA_USB.1	Part 2
FMT_MOF.1	Part 2

FMT_MSA.1	Part 2
FMT_MSA.3-NIAP-0429	Part 2 Extended
FMT_MTD.1:1 through 5	Part 2
FMT_SMF.1	Part 2
FMT_SMR.1	Part 2
FPT_RVM.1	Part 2
FPT_SEP.1	Part 2
FPT_SEP_ENV_(EXP).1	Part 2 Extended
FPT_STM.1	Part 2
FPT_TST_SOF_(EXP).1	Part 2 Extended
FTA_SSL.1	Part 2
FTA_SSL.2	Part 2
FTA_TAB.1	Part 2
FDP_CPD_(EXP).1	Part 2 Extended
FDP_DAU_CPV_(EXP).1	Part 2 Extended
FDP_DAU_CPV_(EXP).2	Part 2 Extended
FDP_DAU_CPV_(EXP).3	Part 2 Extended
FDP_DAU_CPV_(EXP).4	Part 2 Extended
FDP_DAU_CPV_(EXP).5	Part 2 Extended
FDP_DAU_CPI_(EXP).1	Part 2 Extended
FDP_DAU_CPI_(EXP).2	Part 2 Extended
FDP_DAU_CPI_(EXP).3	Part 2 Extended
FDP_DAU_CPI_(EXP).4	Part 2 Extended
FDP_DAU_CPO_(EXP).1	Part 2 Extended
FDP_DAU_CPO_(EXP).2	Part 2 Extended
FDP_DAU_CPO_(EXP).3	Part 2 Extended
FDP_DAU_CRL_(EXP).1	Part 2 Extended
FDP_DAU_ENC_(EXP).1	Part 2 Extended
FDP_DAU_OCS_(EXP).1	Part 2 Extended
FDP_DAU_SIG_(EXP).1	Part 2 Extended
FDP_ETC_ENC_(EXP).1	Part 2 Extended
FDP_ETC_SIG_(EXP).1	Part 2 Extended
FDP_ITC_ENC_(EXP).1	Part 2 Extended
FDP_ITC_SIG_(EXP).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

Package Name	Functional Requirement	Dependency Package
Certification Path Validation – Basic	FDP_CPD_(EXP).1	none
	FDP_DAU_CPI_(EXP).1	
	FDP_DAU_CPV_(EXP).1	
	FDP_DAU_CPV_(EXP).2	
	FDP_DAU_CPO_(EXP).1	
Certification Path Validation – Basic Policy	FDP_DAU_CPI_(EXP).2	Certification Path Validation – Basic
	FDP_DAU_CPO_(EXP).2	
Certification Path Validation – Policy Mapping	FDP_DAU_CPI_(EXP).3	Certification Path Validation – Basic, Certification Path Validation – Basic Policy
	FDP_DAU_CPV_(EXP).3	
	FDP_DAU_CPO_(EXP).3	
Certification Path Validation – Name Constraints	FDP_DAU_CPI_(EXP).4	Certification Path Validation – Basic
	FDP_DAU_CPV_(EXP).4	
	FDP_DAU_CPV_(EXP).5	
PKI Signature Generation	FDP_ETC_SIG_(EXP).1	none
PKI Signature Verification	FDP_ITC_SIG_(EXP).1	Certification Path Validation – Basic
	FDP_DAU_SIG_(EXP).1	
PKI Encryption using Key Transfer Algorithms	FDP_ETC_ENC_(EXP).1	Certification Path Validation – Basic
	FDP_DAU_ENC_(EXP).1	
PKI Decryption using Key Transfer Algorithms	FDP_ITC_ENC_(EXP).1	None
Online Certificate Status Protocol Client	FDP_DAU_OCS_(EXP).1	None
Certificate Revocation List Validation	FDP_DAU_CRL_(EXP).1	None

5.1.1 Certification Path Validation – Basic Package

The functions in this package address the validation of the certification path. All processing defined is X.509 and PKIX compliant.

From certification path processing perspective, certificates can be of up to three types:

- Self-signed trust anchor certificate: The trust anchor can be in the form of a self-signed certificate. . The trust anchor is used to obtain the Distinguished Name (DN), public key, algorithm identifier, and the public key parameters (if applicable). This package permits validation of trust anchor if it is in the form of self-signed certificate,

including validating signature and verifying that the self-signed certificate validity period has not expired.

- Intermediate certificates: These are the certificates issued to the CAs. All certificates in a certification path are intermediate certificates, except the last one.
- End certificate: This is the last certificate in the certification path and is issued to the subscriber of interest. This is typically an end-entity (i.e., not a CA) certificate. However, this package permits that certificate to be a CA certificate also.

This package processes the following security related certificate extensions checks: no-check, keyUsage, extendedKeyUsage, and basicConstraints.

This ST provides the capability to validate path as of a user-defined time called TOI which can be current time or earlier.

5.1.1.1 Class FDP – User Data Protection

FDP_CPD_(EXP).1 Certification path development

Hierarchical to: No other components.

FDP_CPD_(EXP).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_(EXP).1.2	The TSF shall develop the certification path using the following additional matching rule: [*none*].
FDP_CPD_(EXP).1.3	The TSF shall develop the certification path using the following additional matching rule [*none*].
FDP_CPD_(EXP).1.4	The TSF shall bypass any matching rules except [*distinguished name*] if additional certification paths are required.
Dependencies:	None

FDP_DAU_CPI_(EXP).1 Certification path initialisation -- basic

Hierarchical to: No other components.

FDP_DAU_CPI_(EXP).1.1	The TSF shall use the trust anchor provided by [*user*].
FDP_DAU_CPI_(EXP).1.2	The TSF shall obtain the time of interest called “TOI’ from a reliable source [*local environment,* [or application]].
FDP_DAU_CPI_(EXP).1.3	The TSF shall perform the following checks on the trust anchor [*none*]
FDP_DAU_CPI_(EXP).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_(EXP).1 Certificate processing -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_(EXP).1.1The TSF shall reject a certificate if any of 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_(EXP).1.2 The TSF shall bypass the revocation status check if the certificate contains no-check extension.

FDP_DAU_CPV_(EXP).1.3 The TSF shall bypass the revocation check if the revocation information is not available and [*user*] overrides revocation checking.

FDP_DAU_CPV_(EXP).1.4 The TSF shall reject a certificate if the revocation status using [*CRL, OCSP*] demonstrates that the certificate is revoked.

FDP_DAU_CPV_(EXP).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, [FDP_DAU_OCS_(EXP).1 or FDP_DAU_CRL_(EXP).1]

Application Note: *While each certificate is expected to be checked using only one of the revocation mechanisms, each certificate in a certification path can be checked using different revocation mechanism. That is why the selection is one or more.*

FDP_DAU_CPV_(EXP).2 Intermediate certificate processing -- basic

Hierarchical to: No other components.

FDP_DAU_CPV_(EXP).2.1 The TSF shall reject an intermediate certificate if any of the following additional checks fails:

a) basicConstraints field is present with cA = TRUE;

b) pathLenConstraint is not violated; or

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

Dependencies: FDP_DAU_CPV_(EXP).1

FDP_DAU_CPO_(EXP).1 Certification path output -- basic

Hierarchical to: No other components.

FDP_DAU_CPO_(EXP).1.1 The TSF shall output certification path validation failure if any certificate in the certification path is rejected.

FDP_DAU_CPO_(EXP).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_CPO_(EXP).1.3 The TSF shall output the following additional variables from the end certificate [*certificate, subject alternative names, extendedKeyUsage*].

FDP_DAU_CPO_(EXP).1.4 The TSF shall output the subject public key parameters from the certification path parameter state machine.

Dependencies: FDP_DAU_CPV_(EXP).1

5.1.2 Certification Path Validation – Basic Policy Package

The security functional requirements in this package address certificate path processing with the processing of certificatePolicies extension. This package is dependent upon the Certification Path Validation – Basic package.

5.1.2.1 Class FDP – User Data Protection

FDP_DAU_CPI_(EXP).2 Certification path initialisation – basic policy

Hierarchical to: No other components.

FDP_DAU_CPI_(EXP).2.1 The TSF shall use the initial-certificate-policies provided by [*user*].

Dependencies: FDP_DAU_CPI_(EXP).1

FDP_DAU_CPO_(EXP).2 Certification path output – basic policy

Hierarchical to: No other components.

FDP_DAU_CPO_(EXP).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_CPO_(EXP).1

5.1.3 Certification Path Validation – Policy Mapping Package

The security functional requirements in this package address certificate path processing, including the processing of the following certificate policies related extensions: policyMapping, inhibitAnyPolicy, and policyConstraints. This package is dependent upon the Certification Path Validation – Basic package and the Certification Path Validation – Basic Policy package.

5.1.3.1 Class FDP – User Data Protection

FDP_DAU_CPI_(EXP).3 Certification path initialisation – policy mapping

Hierarchical to: No other components.

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

Dependencies: FDP_DAU_CPI_(EXP).2

FDP_DAU_CPV_(EXP).3 Intermediate certificate processing – policy mapping

Hierarchical to: No other components.

FDP_DAU_CPV_(EXP).3.1 The TSF shall use the intermediate certificate to update the following state variables in accordance with X.509 Standard:

a) explicit-policy-indicator

b) policy-mapping-inhibit-indicator

c) inhibit-any-policy-indicator

Dependencies: FDP_DAU_CPV_(EXP).2

FDP_DAU_CPO_(EXP).3 Certification path output – policy mapping

Hierarchical to: No other components.

FDP_DAU_CPO_(EXP).3.1 The TSF shall perform policy processing in accordance with X.509 standard.

FDP_DAU_CPO_(EXP).3.2 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_CPO_(EXP).3.3 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_CPO_(EXP).3.4 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_CPO_(EXP).3.5 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_CPO_(EXP).3.6 The TSF shall output policy mapping history.

FDP_DAU_CPO_(EXP).3.7 The TSF shall output policy qualifiers applicable to output policies.

Dependencies: FDP_DAU_CPO_(EXP).2

5.1.4 Certification Path Validation – Name Constraints Package

The security functional requirements in this package address certificate path processing, including the processing of the nameConstraints extension. This package is dependent upon the Certification Path Validation – Basic package.

5.1.4.1 Class FDP – User Data Protection

FDP_DAU_CPI_(EXP).4 Certification path initialisation – names

Hierarchical to: No other components.

FDP_DAU_CPI_(EXP).4.1 The TSF shall initialize the following: permitted-subtrees = ∞, excluded-subtrees = ∅

Dependencies: FDP_DAU_CPI_(EXP).1

FDP_DAU_CPV_(EXP).4 Certificate processing – name constraints

Hierarchical to: No other components.

FDP_DAU_CPV_(EXP).4.1 The TSF shall reject a certificate if any one of the following is not 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, RFC-822, URL*] in the subjectAlternateName field is in at least one of the permitted-subtrees for that name form; or

d) each hierarchical name form of type [*DN, RFC-822, URL*] in the subjectAlternateName field is in none of the excluded-subtrees for that name form

Dependencies: FDP_DAU_CPV_(EXP).1

FDP_DAU_CPV_(EXP).5 Intermediate Certificate processing – name constraints

Hierarchical to: No other components.

FDP_DAU_CPV_(EXP).5.1 The TSF shall use the intermediate certificate to update the following states:

a) permitted-subtrees

b) excluded-subtrees

Dependencies: FDP_DAU_CPV_(EXP).2

5.1.5 PKI Signature Generation Package

The PKI Signature Generation package invokes a cryptographic module for digital signature generation. The package functionality includes generation of signature information that identifies the signer and is useful in efficient signature verification.

5.1.5.1 Class FDP – User Data Protection

FDP_ETC_SIG_(EXP).1 Export of PKI Signature

Hierarchical to: No other component

FDP_ETC_SIG_(EXP).1.1 The TSF shall invoke the cryptographic module with the user selected private key to generate digital signature.

FDP_ETC_SIG_(EXP).1.2 The TSF shall include the following information with the digital signature [*hashing algorithm, signature algorithm*].

Dependencies: FCS_CRM_FPS_(EXP).1

5.1.6 PKI Signature Verification Package

The PKI Signature Verification package processes and verifies the signature information, and invokes a cryptographic module to verify digital signatures. This package is dependent upon the Certification Path Validation – Basic package. The signature verification package uses the Certification Path Validation package data as input.

5.1.6.1 Class FDP – User Data Protection

FDP_ITC_SIG_(EXP).1 Import of PKI Signature

Hierarchical to no other component

FDP_ITC_SIG_(EXP).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_(EXP).1 Signature Blob Verification

Hierarchical to: No other components.

FDP_DAU_SIG_(EXP).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_(EXP).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_(EXP).1.3 The TSF shall apply the following additional checks [*none*].

Dependencies: FCS_CRM_FPS_(EXP).1, FDP_DAU_CPO_(EXP).1

5.1.7 PKI Encryption using Key Transfer Algorithms Package

This package supports the performance of public key encryption using key transfer algorithms such as RSA. Certification path validation is used to ensure that the correct public key of the decrypting party is used. This package is dependent upon the Certification Path Validation – Basic package.

5.1.7.1 Class FDP – User Data Protection

FDP_ETC_ENC_(EXP).1 Export of PKI Encryption – Key Transfer Algorithms

Hierarchical to: No other component

FDP_ETC_ENC_(EXP).1.1 The TSF shall include the following information with the encrypted data [*key encryption algorithm, data encryption algorithm, decryptor key identifier*].

FDP_ETC_ENC_(EXP).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_(EXP).1, FDP_DAU_CPO_(EXP).1

FDP_DAU_ENC_(EXP).1 PKI Encryption Verification – Key Transfer

Hierarchical to: No other components.

FDP_DAU_ENC_(EXP).1.1 The TSF shall verify that the keyUsage output from Certification Path Validation contains keyEncipherment bit set.

FDP_DAU_ENC_(EXP).1.2 The TSF shall apply the following additional checks [*none*].

Dependencies: FDP_DAU_CPV_OUTCPO_(EXP).1

5.1.8 PKI Decryption using Key Transfer Algorithms Package

This package supports the performance of public key decryption using key transfer algorithms such as RSA. Since only the decrypting party's private key is used, this package does not depend upon certificate path processing.

5.1.8.1 Class FDP – User Data Protection

FDP_ITC_ENC_(EXP).1 Import of PKI Encryption – Key Transfer Algorithms

Hierarchical to: No other components

FDP_ITC_ENC_(EXP).1.1 The TSF shall invoke the cryptographic module with the following information from the encrypted data [*key encryption algorithm, data encryption algorithm, decryptor key identifier*] to perform decryption.

Dependencies: FCS_CRM_FPS_(EXP).1

5.1.9 Online Certificate Status Protocol Client Package

This package allows for making Online Certificate Status Protocol (OSCP) requests and validating OCSP responses. This package permits the use of the OCSP Responder as a trust anchor, as the CA, or an end entity authorized to sign OCSP responses.

5.1.9.1 Class FDP – User Data Protection

FDP_DAU_OCS_(EXP).1 Basic OCSP Client

Hierarchical to: No other component

FDP_DAU_OCS_(EXP).1.1 The TSF shall formulate the OCSP requests in accordance with PKIX RFC 2560.

FDP_DAU_OCS_(EXP).1.2 The OCSP request shall contain the following extensions: [*nonce*].

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

FDP_DAU_OCS_(EXP).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_(EXP).1.7 The TSF shall match the responderID in the OCSP response with the corresponding information in the responder certificate

FDP_DAU_OCS_(EXP).1.8 The TSF shall match the certID in a request with certID in singleResponse.

FDP_DAU_OCS_(EXP).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 > producedAt + x where x=0 is provided by [no one];

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

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

FDP_DAU_OCS_(EXP).1.10 The TSF shall permit [*user*] to override time checks.

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

FDP_DAU_OCS_(EXP).1.12 The TSF shall perform the following additional checks [*request nonce = response nonce*].

Dependencies: FCS_CRM_FPS_(EXP).1, FPT_STM.1

Application Note: For items b and d in FDP_DAU_OCS_(EXP).1.9 the x value is not used.

5.1.10 Certificate Revocation List (CRL) Validation Package

This package is used for validating a CRL. This package 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 compliant implementation can use that or develop a certification path. The ST author has assigned additional rules to process Issuing Distribution Point CRL and Delta CRL.

5.1.10.1 Class FDP – User Data Protection

FDP_DAU_CRL_(EXP).1 Basic CRL Checking

Hierarchical to no other component

FDP_DAU_CRL_(EXP).1.1 The TSF shall obtain the CRL from [*local cache, repository, location pointed to by the CRL DP in public key certificate of interest, user*].

FDP_DAU_CRL_(EXP).1.2 The TSF shall obtain the trusted public key, algorithm, and public key parameters of the CRL issuer.

FDP_DAU_CRL_(EXP).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_(EXP).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_(EXP).1.5 The TSF shall match the issuer field in the CRL with what it assumes to be the CRL issuer.

FDP_DAU_CRL_(EXP).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 if nextUpdate is present and where x=0 is provided by* [*no one*].

FDP_DAU_CRL_(EXP).1.7 The TSF shall permit [*user*] to override time checks.

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

FDP_DAU_CRL_(EXP).1.9 The TSF shall perform the following additional checks [Partitioned CRLs, Delta CRLs, Indirect CRLs, in accordance with RFC 3280].

Dependencies: FCS_CRM_FPS_(EXP).1, FPT_STM.1

Application Note: For item c in FDP_DAU_CRL_(EXP).1.6 the x value is not used.

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, certificate path development –name constraints is 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 is 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.

The following IT Environment requirements not specifically meet by the underlying Operating System ST shall be considered met, if security functional testing commensurate with evaluation assurance level of the ST. It should be noted that to verify that some of these requirements are met, the underlying IT Environment may need additional specific configuration changes after putting the IT Environment in evaluated configuration. In such a case, the IT Environment shall be first configured to meet the evaluated configuration requirements. Then, the testing shall be conducted: FIA_AFL.1, FMT_MOF.1, FMT_MTD.1:1 through 6, FPT_STM.1, FTA_SSL.1, FTA_SSL.2, and FTA_TAB.1

Similarly IT Environment Auditable Events requirements not specifically meet by the underlying Operating System shall be considered met, if security functional testing commensurate with evaluation assurance level of the ST.

Table 5.3 – Security Functional Requirements for the IT Environment

Functional Requirement	Title
FAU_GEN.1-NIAP-0407:1	Audit data generation
FAU_GEN.2-NIAP-0410:1	User identity association
FAU_SAR.1	Audit review
FAU_SAR.2	Restricted audit review
FAU_SAR.3	Selectable audit review
FAU_SEL.1-NIAP-0407	Selective audit
FAU_STG.1-NIAP-0429	Protected audit trail storage
FAU_STG.NIAP-0429-1	Site-configurable Prevention of audit data loss
FCS_CRM_FPS_(EXP).1	FIPS compliant cryptographic module
FDP_ACC.1	Subset access control – PKI Credential Management
FDP_ACF.1-NIAP-0407	Security attribute based access control – PKI Credential Management
FDP_RIP.2	Full residual information protection
FIA_AFL.1	Authentication failure handling

Functional Requirement	Title
FIA_ATD.1	User attribute definition
FIA_UAU.2	User authentication before any action
FIA_UAU.7	Protected authentication feedback
FIA_UID.2	User identification before any action
FIA_USB.1	User-subject binding
FMT_MOF.1	Management of security function behavior
FMT_MSA.1	Management of security attributes
FMT_MSA.3-NIAP-0429	Static attribute initialization
FMT_MTD.1:1	Management of TSF data – I&A Data
FMT_MTD.1:2	Management of TSF data – Authentication Data
FMT_MTD.1:3	Management of TSF data – I&A Attempts
FMT_MTD.1:4	Management of TSF data – Trust Anchors
FMT_MTD.1:5	Management of TSF data – Time
FMT_SMF.1	Specification of management functions
FMT_SMR.1	Security roles
FPT_RVM.1	Non-bypassability of the TSP
FPT_SEP.1	TSF Domain separation
FPT_SEP_ENV_(EXP).1	Support for TSF domain separation
FPT_STM.1	Reliable time stamps
FPT_TST_SOF_(EXP).1	TSF testing for Software only TOEs
FTA_SSL.1	TSF-initiated session locking
FTA_SSL.2	User-initiated locking
FTA_TAB.1	Default TOE access banners

5.2.1 Class FAU – Security Audit

FAU_GEN.1-NIAP-0407:1 Audit data generation

Hierarchical to: No other component

FAU_GEN.1.1-NIAP-0407;1 The ***IT Environment*** shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events listed in Table 5-4; and

c) [*no additional events*].

FAU_GEN.1.2-NIAP-0410;1 The ***IT Environment*** shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three of Table 5-4 below.

Dependencies: FPT_STM.1 Reliable time stamps

Table 5.4 – IT Environment Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1-NIAP-0407:1	None	
FAU_GEN.2-NIAP-0410:1	None	
FAU_SAR.1	Opening the audit trail	The identity of the Audit Administrator performing the function
FAU_SAR.2	Unsuccessful attempts to read information from the audit records	The identity of the administrator performing the function
FAU_SAR.3	None	
FAU_SEL.1-NIAP-0407	All modifications to the audit configuration that occur while the audit collection functions are operating	The identity of the Security Administrator performing the function
FAU_STG.1-NIAP-0429	None	
FAU_STG.NIAP-0429	None	
FCS_CRM_FPS_(EXP).1.	None	
FDP_ACC.1	None	
FDP_ACF.1-NIAP-0407	All requests to perform an operation on an object covered by the SFP	Object identity
FDP_RIP.2	None	
FIA_AFL.1	Reaching of the threshold for the unsuccessful authentication attempts	
FIA_ATD.1	None	
FIA_UAU.2	All use of authentication mechanism	
FIA_UAU.7	None	
FIA_UID.2	All use of identification mechanism	User identity
FIA_USB.1	Success and failure of binding of user security attributes to a subject (e.g. success and failure to create a subject).	
FMT_SMR.1	Modifications to the group of users that are part of a role	
FPT_RVM.1	None	

Requirement	Auditable Events	Additional Audit Record Contents
FPT_SEP.1	None	
FPT_SEP_ENV_(EXP).1	None	
FPT_STM.1	Change to the time	
FTA_TAB.1	None	

FAU_GEN.2-NIAP-0410:1 User identity association

Hierarchical to: No other components.

FAU_GEN.2.1-NIAP-0410;1 For audit events resulting from actions of identified users, the ***IT Environment*** shall be able to associate each auditable event with the identity of the user that caused the event.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1 The ***IT Environment*** shall provide *the administrator* with the capability to read *all audit information* from the audit records.

FAU_SAR.1.2 The ***IT Environment*** shall provide the audit records in a manner suitable for the user to interpret the information.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

FAU_SAR.2.1 The ***IT Environment*** shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

FAU_SAR.3.1 The ***IT Environment*** shall provide the ability to perform *searches and sorting* [*no other operation*] of audit data based on *date, time, user identity and [none*].

Dependencies: FAU_SAR.1 Audit review

FAU_SEL.1-NIAP-0407 Selective audit

Hierarchical to:	No other components.
FAU_SEL.1.1-NIAP-0407	The ***IT Environment*** shall ***allow only the administrator*** to include or exclude auditable events from the set of audited events based on the following attributes: a) user identity; b) event type; c) [*none*]; d) success of auditable security events; e) failure of auditable security events; and f) [*no additional criteria*].
Dependencies:	FAU_GEN.1 Audit data generation FMT_MTD.1 Management of TSF data
Application Note:	*"event type" is defined as a class of audit event, e.g., attempt to login, attempt to access an object, attempt to create a user*.

FAU_STG.1-NIAP-0429 Protected audit trail storage

Hierarchical to:	No other components.
FAU_STG.1.1-NIAP-0429	The ***IT Environment*** shall ***restrict the deletion of*** stored audit records in the audit trail ***to the administrator***.
FAU_STG.1.2-NIAP-0429	The ***IT Environment*** shall be able to *prevent* modifications to the audit records in the audit trail.
Dependencies:	FAU_GEN.1 Audit data generation

FAU_STG.NIAP-0429-1 Site-configurable Prevention of audit data loss

Hierarchical to: FAU_STG.4	
FAU_STG.NIAP-0429-1.1	The ***IT Environment*** shall provide an authorized administrator with the capability to select one or more of the following actions [*prevent auditable events, except those taken by the authorized user with special rights, overwrite the oldest stored audit records*] and [*no additional options*] to be taken if the audit trail is full.
FAU_STG.NIAP-0429-1.2	The ***IT Environment*** shall [*prevent auditable events, except those taken by the authorized user with special rights*] and [*no other action*] if the audit trail is full and no other action has been selected.
Dependencies:	FAU_STG.1 Protected Audit Trail Storage FMT_MTD.1 Management of TSF Data
Application Note:	The IT Environment provides the administrator the option of preventing audit data loss by preventing auditable events from occurring. The

administrator's actions under these circumstances are not required to be audited. The IT Environment also provides the administrator the option of overwriting "old" audit records rather than preventing auditable events, which may protect against a denial-of-service attack.

5.2.2 Class FCS – Cryptographic Support

FCS_CRM_FPS_(EXP).1 FIPS compliant cryptographic module

Hierarchical to: No other components.

FCS_CRM_FPS_(EXP).1.1 The IT environment shall provide all cryptographic modules necessary for the TSF.

FCS_CRM_FPS_(EXP).1.2 Each cryptographic module shall be FIPS 140 series Level 1 validated.

Dependencies: None.

5.2.3 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 Subjects: [User], Objects: cryptographic key, public key certificate, [trust anchors, PKIF.dll/libPKIF.so, PKIFCMS.dll/ libPKIFCMS.so], Operations: [Import, export, and delete public key certificate]

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACF.1-NIAP-0407 Security attribute based access control – PKI Credential Management

Hierarchical to: No other components.

FDP_ACF.1.1-NIAP-0407 The ***IT Environment*** shall enforce the *PKI credential management SFP* to objects based on the following: list of subjects: *all subjects;* list of objects: *cryptographic keys and public key certificate;* list of subjects and object attributes: identity of the subject and the set of roles that the subject is authorized to assume [*object attributes* PKIF.dll/libPKIF.so, PKIFCMS.dll/ libPKIFCMS.so *access rights*].

FDP_ACF.1.2-NIAP-0407 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--NIAP-0407 The ***IT Environment*** shall explicitly authorize access of subjects to objects based on the following additional rules: [*users shell have read and execute privileges on* PKIF.dll/libPKIF.so and PKIFCMS.dll/ libPKIFCMS.so].

FDP_ACF.1.4--NIAP-0407 The ***IT Environment*** shall explicitly deny access of subjects to objects based on the [*users who have read and execute privileges for* PKIF.dll/libPKIF.so and PKIFCMS.dll/ libPKIFCMS.so *shall not be able to delete or modify the file*].

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

FDP_RIP.2 Full residual information protection

Hierarchical to: FDP_RIP.1

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

Dependencies: No dependencies

5.2.4 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 *an administrator configurable positive integer within* [5] unsuccessful authentication attempts occur related to [user *authentication*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the ***IT Environment*** shall prevent *all entities requesting authentication other than the administrator* from performing activities that require authentication until an action is taken by the administrator.

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: user ID, *role.*
Dependencies:	No dependencies

FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1

FIA_UAU.2.1	The ***IT Environment*** shall require each user to be successfully authenticated before allowing any other ***IT Environment*** 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.2 User identification before any action

Hierarchical to: FIA_UID.1

FIA_UID.2.1	The ***IT Environment*** shall require each user identify itself before allowing any other ***IT Environment*** mediated actions on behalf of that user.
Dependencies:	No dependencies

FIA_USB.1 User-subject binding

Hierarchical to: No other components

FIA_USB.1.1	The ***IT Environment*** shall associate the following user security attributes with subjects acting on the behalf of that user: *all user security attributes*.
FIA_USB.1.2	The ***IT Environment*** shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: *none*.
FIA_USB.1.3	The ***IT Environment*** shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: *none*.
Dependencies:	FIA_ATD.1 User attribute definition

5.2.5 Class FMT – Security Management

FMT_MOF.1 Management of security function behavior

Hierarchical to: No other components

FMT_MOF.1.1	The ***IT Environment*** shall restrict the ability to [*disable, enable, modify the behavior* of] the functions *audit*, [none] to *the administrator.*
Dependencies:	FMT_SMF.1 Specification of Management Functions, FMT_SMR.1 Security roles

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*] 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 [FDP_ACC.1 Subset access control or FDP_IFC Subset information flow control]

FMT_MSA.3-NIAP-0429 Static attribute initialization

Hierarchical to: No other components

FMT_MSA.3.1-NIAP-0429	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-NIAP-0429	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.1:1 Management of TSF data – I&A Data

Hierarchical to: No other components

FMT_MTD.1.1;1	The ***IT Environment*** shall restrict the ability to *initialize and modify identification data and authentication data* to *administrator*.
Dependencies:	FMT_SMF.1 Specification of Management Functions, FMT_SMR.1 Security roles

FMT_MTD.1:2 Management of TSF data – Authentication Data

Hierarchical to: No other components

FMT_MTD.1.1;2	The ***IT Environment*** shall restrict the ability to *modify authentication data* to *administrator and the user owning the account*.
Dependencies:	FMT_SMF.1 Specification of Management Functions, FMT_SMR.1 Security roles

FMT_MTD.1:3 Management of TSF data – I&A Attempts

Hierarchical to: No other components

FMT_MTD.1.1;3	The ***IT Environment*** shall restrict the ability to *initialize and modify number of unsuccessful authentication* to *administrator*.
Dependencies:	FMT_SMF.1 Specification of Management Functions, FMT_SMR.1 Security roles

FMT_MTD.1:4 Management of TSF data – Trust Anchors

Hierarchical to: No other components

FMT_MTD.1.1;4	The ***IT Environment*** shall restrict the ability to *add and delete trust anchors**,*** to [*user*].
Dependencies:	FMT_SMF.1 Specification of Management Functions, FMT_SMR.1 Security roles

FMT_MTD.1:5 Management of TSF data – Time

Hierarchical to: No other components

FMT_MTD.1.1;5	The ***IT Environment*** shall restrict the ability to *initialize and modify system time* to *administrator*.
Dependencies:	FMT_SMF.1 Specification of Management Functions, FMT_SMR.1 Security roles

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: *audit management*, *user identity management, trust anchor management, system time management,* [none].
Dependencies:	No dependencies

FMT_SMR.1 Security roles

Hierarchical to: No other component

FMT_SMR.1.1	The ***IT Environment*** shall maintain the roles *user, administrator* [none].
FMT_SMR.1.2	The ***IT Environment*** shall be able to associate users with roles.
Dependencies:	FIA_UID.1 Timing of identification

.

5.2.6 Class FPT – Protection of the TOE Security Functions

FPT_RVM.1 Non-bypassability of the IT Environment security policy

Hierarchical to: No other components

FPT_RVM.1.1	The ***IT Environment*** shall ensure that ***IT Environment security policy*** enforcement functions are invoked and succeed before each function within the ***IT Environment*** **Scope of Control** is allowed to proceed.
Dependencies:	No dependencies

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 ***IT Environment Scope of Control***.
Dependencies:	No dependencies

FPT_SEP_ENV_(EXP).1 Support for TSF domain separation

Hierarchical to: No other components

FPT_SEP_ENV_(EXP).1.1	The IT Environment shall maintain a security domain that protects the TSF from interference and tampering by untrusted subjects.
Dependencies:	None.
Application Note:	*This requirement is provide protection to the TSF. It is generally met by FPT_SEP and DAC on TSF executables, TSF Data, TSF User data including file systems objects and IPC mechanisms such as pipes, shared memory etc..*

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 its own ***and TSF*** use.
Dependencies:	None.

FPT_TST_SOF_(EXP).1 TSF testing for Software only TOEs

Hierarchical to: No other components.

FPT_TST_SOF_(EXP).1.1 The ***IT Environment*** shall provide administrator with the capability to verify the integrity of the following TSF data: [none].

FPT_TST_SOF_(EXP).1.2 The ***IT Environment*** shall provide administrator with the capability to verify the integrity of stored TSF executable code.

Dependencies: No dependencies

5.2.7 Class FTA – TOE Access

FTA_SSL.1 TSF-initiated session locking

Hierarchical to: No other components.

FTA_SSL.1.1 The ***IT Environment*** shall lock an interactive session after [15 minutes] by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 The ***IT Environment*** shall require the following events to occur prior to unlocking the session: *authentication by the user.*

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL.2 User-initiated locking

Hierarchical to: No other components.

FTA_SSL.2.1 The ***IT Environment*** shall allow user-initiated locking of the user's own interactive session, by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 The ***IT Environment*** shall require the following events to occur prior to unlocking the session: *authentication by the user*.

Dependencies: FIA_UAU.1 Timing of authentication

FTA_TAB.1 Default TOE access banners

Hierarchical to: No other components.

FTA_TAB.1.1	Before establishing a user session, the ***IT Environment*** shall display an advisory warning message regarding unauthorized use of the ***System***.
Dependencies:	No dependencies

5.3 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.

5.4 Assurance Requirements

The TOE Evaluation Assurance Level is EAL4 augmented by ALC_FLR.2. EAL4 permits a PKE application developer to gain 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.2 augmentation is done to ensure compliance with the Basic Robustness assurance requirements. The assurance components are listed in Table 5.5 below. These Security Assurance Requirements are drawn from the Common Criteria for Information Technology Security Evaluation, Part 3, dated August 2005, Version 2.3

Table 5.5 – EAL4 with Augmentation Assurance Requirements

Class	Assurance Component	Assurance Component Description
Security Target	ASE_PPC_(EXP).2	Security Target, PP claims Evaluation requirements
Configuration Management	ACM_AUT.1	Partial CM automation
	ACM_CAP.4	Generation support and acceptance procedures
	ACM_SCP.2	Problem tracking CM coverage
Delivery and Operation	ADO_DEL.2	Detection of modification
	ADO_IGS.1	Installation, generation, and start-up procedures
Development	ADV_FSP.2	Fully defined external interfaces
	ADV_HLD_(EXP).6	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

Class	Assurance Component	Assurance Component Description
Guidance Documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life Cycle Support	ALC_DVS.1	Identification of security measures
	ALC_FLR.2	**Flaw Reporting Procedures**
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN_(EXP).3	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability Assessment	AVA_MSU.2	Validation of analysis
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.2	Independent vulnerability analysis

5.4.1 ASE_PPC_(EXP).2 Security Target, PP claims, Evaluation requirements

Dependencies

ASE_OBJ.1 Security Target, Security objectives, Evaluation requirements

ASE_REQ.1 Security Target, IT security requirements, Evaluation requirements

Developer action elements

ASE_PPC_(EXP).2.1D The developer shall provide any PP claims as part of the ST.

ASE_PPC_(EXP).2.2D The developer shall provide the PP claims rationale for each provided PP claim.

ASE_PPC_(EXP).2.3D The developer shall provide the Security Target (ST) for the underlying Operating System(s).

Content and presentation of evidence elements

ASE_PPC_(EXP).2.1C Each PP claim shall identify the PP for which compliance is being claimed, including qualifications needed for that claim.

ASE_PPC_(EXP).2.2C Each PP claim shall identify the IT security requirements statements that satisfy the permitted operations of the PP or otherwise further qualify the PP requirements.

ASE_PPC_(EXP).2.3C Each PP claim shall identify security objectives and IT security requirements statements contained in the ST that are in addition to those contained in the PP.

Evaluator action elements

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

ASE_PPC_(EXP).2.2E The evaluator shall confirm that the PP claims are a correct instantiation of the PP.

ASE_PPC_(EXP).2.3E The evaluator shall verify that the operating systems has obtained a CCEVS certificate for the TOE assurance level (i.e., EAL 3 or EAL 4 augmented by ALC_FLR.2)

ASE_PPC_(EXP).2.4E The evaluator shall examine the operating ST to verify that the operating system assumptions do not contradict the assumptions in this PP.

ASE_PPC_(EXP).2.5E The evaluator shall examine the operating ST and the TOE ST to verify that the operating system assumptions do not contradict the assumptions in the TOE ST.

ASE_PPC_(EXP).2.6E The evaluator shall examine the SFRs in the ST to verify that they provide demonstrable conformance to the SFRs for the IT Environment in this PP, except for the following SFRs:

- FCS_CRM_FPS_(EXP).1
- FPT_SEP_ENV_(EXP).1
- FPT_TST_SOF_(EXP).1

5.4.2 ADV_HLD_(EXP).6 Security enforcing high-level design

Dependencies

ADV_FSP.1 Informal functional specification

ADV_RCR.1 Informal correspondence demonstration

Developer action elements

ADV_HLD_(EXP).6.1D	The developer shall provide the high-level design of the TSF.
ADV_HLD_(EXP).6.2D	The developer shall describe how FCS_CRM_FPS_(EXP).1 requirement is met.
ADV_HLD_(EXP).6.3D	The developer shall describe how FTP_SEP_ENV_(EXP).1 requirement is met.
ADV_HLD_(EXP).6.4D	The developer shall describe how FTP_TST_SOF_(EXP).1 requirement is met.
	Content and presentation of evidence elements
ADV_HLD_(EXP).6.1C	The presentation of the high-level design shall be informal.
ADV_HLD_(EXP).6.2C	The high-level design shall be internally consistent.
ADV_HLD_(EXP).6.3C	The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD_(EXP).6.4C	The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD_(EXP).6.5C	The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD_(EXP).6.6C	The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD_(EXP).6.7C	The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD_(EXP).6.8C	The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD_(EXP).6.9C	The high-level design shall describe the separation of the TOE into TSPenforcing and other subsystems.
ADV_HLD_(EXP).6.10C	The high-level design shall identify the FIPS 140 validated cryptographic module.

ADV_HLD_(EXP).6.11C The high-level design shall describe how FPT_SEP_ENV_(EXP).1 requirement is met. The high-level design shell describe how discretionary access control is enforced on the underlying operating system.

ADV_HLD_(EXP).6.12C The high-level design shall describe the composite mechanism, including details of the operating system interfaces used and how the TOE preserves the underlying operating system security.

Evaluator action elements

ADV_HLD_(EXP).6.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD_(EXP).6.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD_(EXP).6.3E The evaluator CCTL shall check that the design document lists the cryptographic module(s) used by the TOE. .

ADV_HLD_(EXP).6.4E The evaluator shall examine the NIST web site to verify that each cryptographic module listed has been FIPS 140 validated.

ADV_HLD_(EXP).6.5E The evaluator shall use the FIPS 140 security policy for the module and vendor guidance to configure the module in FIPS 140 compliant and validated mode.

ADV_HLD_(EXP).6.6E The evaluator shall check that the design document describes how the FTP_SEP_ENV_(EXP).1 requirement is met.

ADV_HLD_(EXP).6.7E The evaluator shall analyze the design document to analyze that the FTP_SEP_ENV_(EXP).1 requirement is satisfied.

ADV_HLD_(EXP).6.8E The evaluator shall check that the design document describes how the FPT_TST_SOF_(EXP).1 requirement is met.

5.4.3 ATE_FUN_(EXP).3 Functional testing

Objectives

The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

Developer action elements

ATE_FUN_(EXP).3.1D	The developer shall test the TSF and document the results.
ATE_FUN_(EXP).3.2D	The developer shall provide test documentation
ATE_FUN_(EXP).3.3D	The developer shall configure the operating system in accordance with the Guidance Documentation in its evaluated configuration.
ATE_FUN_(EXP).3.4D	The developer shall test FIA_AFL.1 Authentication failure handling IT Environment SFR and document the results.
ATE_FUN_(EXP).3.5D	The developer shall test FIA_AFL.1 Authentication failure audit event and document the results.
ATE_FUN_(EXP).3.6D	The developer shall test FMT_MOF.1 Management of security function behavior IT Environment SFR and document the results
ATE_FUN_(EXP).3.7D	The developer shall test FMT_MTD.1:1 Management of TSF data – I&A Data IT Environment SFR and document the results.
ATE_FUN_(EXP).3.8D	The developer shall test FMT_MTD.1:3 Management of TSF data – I&A Attempts IT Environment SFR and document the results.
ATE_FUN_(EXP).3.9D	The developer shall test FMT_MTD.1:4 Management of TSF data – Trust Anchors IT Environment SFR and document the results.
ATE_FUN_(EXP).3.10D	The developer shall test FMT_MTD.1:5 Management of TSF data – Time IT Environment SFR and document the results.
ATE_FUN_(EXP).3.11D	The developer shall test FTA_SSL.1 TSF-initiated session locking IT Environment SFR and document the results.
ATE_FUN_(EXP).3.12D	The developer shall test FTA_SSL.2 User-initiated session locking IT Environment SFR and document the results.
ATE_FUN_(EXP).3.13D	The developer shall test FTA_TAB.1 Default TOE access banners IT Environment SFR and document the results.
ATE_FUN_(EXP).3.14D	The developer shall test FAU_SEL.1-NIAP-0407 IT environment SRF audit requirement and document the results.

Content and presentation of evidence elements

ATE_FUN_(EXP).3.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN_(EXP).3.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN_(EXP).3.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN_(EXP).3.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN_(EXP).3.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements

ATE_FUN_(EXP).3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6 TOE Summary Specification

PKIFv2 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 PKIFv2 permits applications to perform a variety of tasks in addition to and in support of the functions listed above. The following sections describe the PKIFv2 functions and the TSF interface of the library.

PKIFv2 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

PKIFv2 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. PKIFv2 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. This is an end-entity certificate (i.e., a certificate issued to an entity not functioning as a CA).

PKIFv2 processes the following security-related certificate extensions: ocsp-nocheck, keyUsage, extendedKeyUsage, and basicConstraints. PKIFv2 performs the processing of

the following certificate policy-related extensions: certificatePolicies, policyMapping, inhibitAnyPolicy, policyConstraints, and nameConstraints extensions

PKIFv2 can generate Online Certificate Status Protocol (OCSP) requests and validate OCSP responses to determine the revocation status of public key certificates. PKIFv2 verifies OCSP Responder as a trust anchor or as an end entity authorized to sign OCSP responses. PKIFv2 establishes trust in the OCSP responder certificates by performing Certification Path Validation.

PKIFv2 provides Certificate Revocation List (CRL) validation functionality that enables applications to determine the revocation status of a certificate using a CRL. PKIFv2 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.

PKIFv2 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 PKIFv2 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_(EXP).1
- FDP_DAU_CPI_(EXP).1
- FDP_DAU_CPV_(EXP).1
- FDP_DAU_CPV_(EXP).2
- FDP_DAU_CPO_(EXP).1
- FDP_DAU_CPI_(EXP).2
- FDP_DAU_CPO_(EXP).2
- FDP_DAU_CPI_(EXP).3
- FDP_DAU_CPV_(EXP).3
- FDP_DAU_CPO_(EXP).3

- FDP_DAU_CPI_(EXP).4
- FDP_DAU_CPV_(EXP).4
- FDP_DAU_CPV_(EXP).5
- FDP_DAU_OCS_(EXP).1
- FDP_DAU_CRL_(EXP).1

6.2 Signature Generation Functionality

PKIFv2 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 PKIFv2 are based on those defined in [RFC3369]. Several CMS related samples are provided in the PKIFv2 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_(EXP).1

6.3 PKI Signature Verification Functionality

PKIFv2 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 PKIFv2 are based on those defined in [RFC3369]. Several CMS related samples are provided in the PKIFv2 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, CMSVerificationStatus &, CPKIFCertificatePtr &, CMSPathValidationStatus)`	Verifies a signature contained in a SignedData message including validation of the signer's certificate

`bool CPKIFSignedData::Verify (int, CMSVerificationStatus &, 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_(EXP).1
- FDP_DAU_SIG_(EXP).1

6.4 PKI Encryption using Key Transfer Algorithms Functionality

PKIFv2 enables application to perform public key encryption using key transfer algorithms such as RSA. The CMS structures implemented by PKIFv2 are based on those defined in [RFC3369]. Several CMS related samples are provided in the PKIFv2 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 &, CMSPathValidationStatus)`	Adds a recipient to an EnvelopedData message after verifying the recipient's certificate
`void CPKIFEnvelopedData::AddRecipient (CPKIFCertificatePtr &, 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_(EXP).1
- FDP_DAU_ENC_(EXP).1

6.5 PKI Decryption using Key Transfer Algorithms Functionality

PKIFv2 enables applications to perform private key decryption using key transfer algorithms such as RSA. The CMS structures implemented by PKIFv2 are based on those defined in [RFC3369]. Several CMS related samples are provided in the PKIFv2 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_(EXP).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(IPKIFColleaguePtr&)
2) CPKIFLDAPRepository::CPKIFLDAPRepository(void)
3) void CPKIFLDAPRepository::Set_Port(int)
4) void CPKIFLDAPRepository::SetHost(const char *)

Cryptography

5) const char * CPKIFCredential::ID(void)
6) const char * 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 &, CMSPathValidationStatus)
21) void CPKIFEnvelopedData::AddRecipient(CPKIFCertificatePtr &, 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, CMSVerificationStatus &, CPKIFCertificatePtr &, CMSPathValidationStatus)

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

37) CPKIFSignerInfo::CPKIFSignerInfo(void)

38) void CPKIFSignerInfo::SetCredential(CPKIFCredentialPtr &)

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

40) void PKIFCMS_API 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(const char *)

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 &, CPKIFPathValidationResults &)

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) const vector< CPKIFPolicyInformationListPtr > * 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) const char * CPKIFException::GetDescription(void)

71) int CPKIFException::GetErrorCode(void)

72) CAC_API IPKIFMediatorPtr MakeDefaultMediator(bool, CPKIFOCSPCheckerPtr&)

73) template<class X> X* IPKIFMediator::GetMediator() const

X.509 ASN.1 Encoding/Decoding

74) CPKIFOIDPtr CPKIFAlgorithmIdentifier::oid(void)

75) bool CPKIFAlgorithmIdentifier::hasParameters() const

76) CPKIFBufferPtr CPKIFAlgorithmIdentifier::parameters(void)

77) CPKIFBuffer::CPKIFBuffer(const unsigned char *,unsigned int)

78) const unsigned char * CPKIFBuffer::GetBuffer(void)

79) unsigned int CPKIFBuffer::GetLength(void)

80) CPKIFCertificate::CPKIFCertificate(void)

81) void CPKIFCertificate::Decode(const unsigned char *,int)

82) CPKIFBufferPtr CPKIFCertificate::Encoded(void)

83) template <typename T> shared_ptr<T> GetExtension()

84) CPKIFNamePtr CPKIFCertificate::Subject(void)

85) CPKIFSubjectPublicKeyInfoPtr CPKIFCertificate::SubjectPublicKeyInfo(void)

86) void CPKIFExtendedKeyUsage::KeyPurposeIDs(std::vector<CPKIFOIDPtr> &)

87) CPKIFNamePtr CPKIFGeneralName::directoryName(void)

88) const char * CPKIFGeneralName::dnsName(void)

89) CPKIFGeneralName::GENNAMETYPE CPKIFGeneralName::GetType(void)

90) CPKIFBufferPtr CPKIFGeneralName::ipAddress(void)

91) CPKIFBufferPtr CPKIFGeneralName::otherName(void)

92) const char * CPKIFGeneralName::rfc822Name(void)

93) const char * CPKIFGeneralName::uri(void)

94) CPKIFBufferPtr CPKIFGeneralName::x400Address(void)

95) bool CPKIFKeyUsage::CRLSign(void)

96) bool CPKIFKeyUsage::DataEncipherment(void)

97) bool CPKIFKeyUsage::DecipherOnly(void)

98) bool CPKIFKeyUsage::DigitalSignature(void)

99) bool CPKIFKeyUsage::EncipherOnly(void)

100) bool CPKIFKeyUsage::KeyAgreement(void)

101) bool CPKIFKeyUsage::KeyCertSign(void)

102) bool CPKIFKeyUsage::KeyEncipherment(void)

103) bool CPKIFKeyUsage::NonRepudiation(void)

104) CPKIFOID::CPKIFOID(const std::string &)

105) const char* CPKIFOID::ToString(void)

106) CPKIFPolicyInformation::CPKIFPolicyInformation(const CPKIFOIDPtr &)

107) CPKIFOIDPtr CPKIFPolicyInformation::PolicyOID(void)

108) CPKIFPolicyQualifierListPtr CPKIFPolicyInformation::Qualifiers(void)
109) void CPKIFSubjectAltName::GeneralNames(CPKIFGeneralNameList &)
110) CPKIFAlgorithmIdentifierPtr CPKIFSubjectPublicKeyInfo::alg(void)
111) CPKIFBufferPtr CPKIFSubjectPublicKeyInfo::rawKey(void)
112) CPKIFTime::CPKIFTime(const char *)

6.7 Assurance Measures

PKIFv2 satisfies the assurance requirements for Evaluation Assurance Level EAL4 augmented with ALC_FLR.2. 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_(EXP).6	Security enforcing high-level design	[HELP], [INT]
ADV_IMP.1	Subset of the Implementation of the TSF	Source Code, [INT]
ADV_LLD.1	Descriptive low-level design	[HELP], [INT]
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.2	Basic flaw remediation	[CMPLAN]
ALC_LCD.1	Developer defined life-cycle model	[LCMOD]
ALC_TAT.1	Well-defined development tools	[DEVSEC]
ATE_COV.2	Analysis of coverage	[TSTCOV]
ATE_DPT.1	Testing: high-level design	[TSTCOV]
ATE_FUN_(EXP).3	Functional testing	[TEST], [TSTLST]
ATE_IND.2	Independent testing – sample	To be provided by the evaluation lab
AVA_MSU.2	Validation of analysis	[VULAN]

Assurance Component ID	Assurance Component Title	How Satisfied
AVA_SOF.1	Strength of TOE security function evaluation	Not Applicable
AVA_VLA.2	Independent vulnerability analysis	[VULAN], [TEST]

Description of the TOE assurance documents listed in the table above can be found in the section 1.3.

7 PP Conformance

This ST is conformant with the *U.S. Government Family of Protection Profiles Public Key-Enabled Applications for Basic Robustness Environments, Version 2.77 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.

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	A.Configuration	The TOE will be properly installed and configured.	AE.Configuration
2	A.Low	The attack potential on the TOE is assumed to be low.	AE.Low
3	A.NO_EVIL	Administrators are non-hostile, appropriately trained and follow all administrator guidance	AE.NO_EVIL
4	A.PHYSICAL	It is assumed that the environment provided the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.	AE.PHYSICAL

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

Threats for the 1. CPV – Basic Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
1	T.Certificate_Modi	An untrusted user may modify a certificate resulting in using a wrong public key.	T.Certificate_Modi
2	T.DOS_CPV_Basic	The revocation information or access to revocation information could be made unavailable, resulting in loss of system availability.	T.DOS_CPV_Basic
3	T.Expired_Certificate	An expired (and possibly revoked) certificate as of TOI could be used for signature verification.	T.Expired_Certificate
4	T.Untrusted_CA	An untrusted entity (Certification Authority (CA)) may issue certificates to bogus entities, permitting those entities to assume identity of other legitimate users.	T.Untrusted_CA
5	T.No_Crypto	The user public key and related information may not be available to carry out the cryptographic function.	T.No_Crypto
6	T.Path_Not_Found	A valid certification path is not found due to lack of system functionality.	T.Path_Not_Found
7	T.Revoked_Certificate	A revoked certificate could be used as valid, resulting in security compromise.	T.Revoked_Certificate
8	T.User_CA	A user could act as a CA, issuing unauthorized certificates.	T.User_CA
Threats for the 2. CPV – Basic Policy Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
9	T.Unknown_Policies	The user may not know the policies under which a certificate was issued.	T.Unknown_Policies
Threats for the 3. CPV – Policy Mapping Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
10	T.Mapping	The user may accept unacceptable certificates or reject acceptable certificates due to improper certificate policy mapping.	T.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.	T.Wrong_Policy_Dec
Threats for the 4. CPV – Name Constraints Package			
#	**PP Threat Name**	**Threat Description**	**ST Threat Name**
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.	T.Name_Collision

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.AUDIT_COMPROMISE	A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action	TE.AUDIT_COMPROMISE
2E	T.CHANGE_TIME	An unauthorized user may change the TSF notion of time resulting in accepting old revocation information or expired certificates.	TE.CHANGE_TIME
3E	T.CRYPTO_COMPROMISE	A user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.	TE CRYPTO_COMPROMISE
4E	T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	TE.MASQUERADE
5E	T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.	TE.POOR_TEST
6E	T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.	TE.RESIDUAL_DATA
7E	T.TSF_COMPROMISE	A user or process may cause, through an unsophisticated attack, TSF data, security attributes, or executable code to be inappropriately accessed (viewed, modified, or deleted).	TE.TSF_COMPROMISE
8E	T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.	TE. UNATTENDED_SESSION
9E	T.UNAUTHORIZED_ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy.	TE.UNAUTHORIZED_ACCESS

Base Threats to Security for all PPs in this PP Family			
#	PP Threat Name	Threat Description	ST Threat Name
10E	T.UNIDENTIFIED_ACTIONS	The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.	TE.UNIDENTIFIED_ACTIONS

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	OE.AUDIT_GENERATION	The IT Environment will provide the capability to detect and create records of security-relevant events associated with users.	OE.AUDIT_GENERATION
2E	OE.AUDIT_PROTECTION	The IT Environment will provide the capability to protect audit information.	OE.AUDIT_PROTECTION
3E	OE.AUDIT_REVIEW	The IT Environment will provide the capability to selectively view audit information,	OE.AUDIT_REVIEW
4E	OE.Configuration	The TOE shall be installed and configured properly for starting up the TOE in a secure state.	OE.Configuration
5E	OE.CORRECT_TSF_OPERATION	The IT environment will provide functionality to support the correct operation of the TSF. The IT Environment will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.	OE.CORRECT_TSF_OPERATION
6E	OE.Crypto	The TOE shall use NIST FIPS 140-2 validated cryptographic services provided by the IT Environment.	OE.Crypto
7E	OE.DISPLAY_BANNER	The IT Environment will display an advisory warning regarding use of the TOE.	OE.DISPLAY_BANNER

8E	OE.Low	The Identification and Authentication functions in the IT Environment 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
9E	OE.MANAGE	The IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	OE.MANAGE
10E	OE.MEDIATE	The IT Environment will protect user data in accordance with its security policy.	OE.MEDIATE
11E	OE.NO_EVIL	Sites using the TOE will ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.	OE.NO_EVIL
12E	O. PHYSICAL	The non-IT environment will 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
14E	OE.RESIDUAL_INFORMATION	The IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure.	OE.RESIDUAL_INFORMATION
15E	OE.SELF_PROTECTION	The IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.	OE.SELF_PROTECTION
16E	OE.TIME_STAMPS	The IT Environment will provide reliable time for the TOE use.	OE.TIME_STAMPS
17E	OE.TIME_TOE	The IT Environment will provide reliable time for the TOE use.	OE.TIME_TOE
18E	OE.TOE_ACCESS	The IT Environment will provide mechanisms that control a user's logical access to the TOE.	OE.TOE_ACCESS

19E	OE.TOE_PROTECTION	The IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification.	OE.TOE_PROTECTION

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

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

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

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.Configuration	OE.Configuration
AE.Low	OE.Low
AE.NO_EVIL	OE.NO_EVIL
AE.PHYSICAL	OE.PHYSICAL
AE.GOOD_USER	OE.GOOD_USER
P.ACCESS_BANNER	OE.DISPLAY_BANNER
P.ACCOUNTABILITY	OE.AUDIT_GENERATION; OE.TIME_STAMPS; OE.TOE_ACCESS; OE.TIME_TOE
P.CRYPTOGRAPHY	OE.CRYPTOGRAPHY
TE.AUDIT_COMPROMISE	OE.AUDIT_PROTECTION; OE.RESIDUAL_INFORMATION; OE.SELF_PROTECTION; OE.TOE_PROTECTION
TE.CHANGE_TIME	OE.TIME_TOE
TE.CRYPTO_COMPROMISE	OE.CRYPTOGRAPHY; OE.PHYSICAL
TE.MASQUERADE	OE.TOE_ACCESS
TE.POOR_TEST	OE.CORRECT_TSF_OPERATION
TE.RESIDUAL_DATA	OE.RESIDUAL_INFORMATION
TE.TSF_COMPROMISE	OE.RESIDUAL_INFORMATION; OE.SELF_PROTECTION; OE.TOE_PROTECTION; OE.MANAGE
TE.UNATTENDED_SESSION	OE.TOE_ACCESS
TE.UNAUTHORIZED_ACCESS	OE.MEDIATE
TE.UNIDENTIFIED_ACTIONS	OE.AUDIT_REVIEW; OE.AUDIT_GENERATION; OE.TIME_STAMPS; OE.TIME_TOE

AE.NO_EVIL states that administrators are non-hostile, appropriately trained and follow all administrator guidance. This assumption is mapped to:

:

- **OE.NO_EVIL**, which states that sites using the TOE will ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.

AE.PHYSICAL states that environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE. This assumption is mapped to:

- **OE.PHYSICAL**, which states that the non-IT environment will 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.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.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.GOOD_USER states that TOE users are non-hostile and follow all user guidance. This assumption is mapped to:

- **OE.GOOD_USER,** which states that sites using the TOE will ensure that TOE users are non-hostile and follow all user guidance.

P.ACCESS_BANNER states that the IT Environment shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. This policy is mapped to:

- **OE.DISPLAY_BANNER** which states that the IT Environment will display an advisory warning regarding use of the TOE. **OE.DISPLAY_BANNER** satisfies this policy by ensuring that the TOE displays an administrator configurable banner that provides all interactive users with a warning about the unauthorized use of the TOE

P.ACCOUNTABILITY states that the authorized users of the TOE shall be held accountable for their actions within the TOE. This policy is mapped to:

- **OE.AUDIT_GENERATION** which states that the IT Environment will provide the capability to detect and create records of security-relevant events associated with users. **OE.AUDIT_GENERATION** addresses this policy by providing the administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made (e.g. access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.).
- **OE.TIME_STAMPS** which states that the IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. **OE.TIME_STAMPS** plays a role in supporting this policy by requiring the IT Environment to provide a reliable time stamp (configured locally by the Security Administrator or via an external NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred.

- **OE.TIME_TOE** which states that the IT Environment will provide reliable time for the TOE use. **OE.TIME_STAMPS** plays a role in supporting this policy by permitting the TOE to provide reliable time on audit records generated by the TOE.
- **OE.TOE_ACCESS** which states that the IT Environment will provide mechanisms that control a user's logical access to the TOE. **OE.TOE_ACCESS** supports this policy by requiring the IT Environment to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.

P.CRYPTOGRAPHY states that only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services). This policy is mapped to:

- **OE.CRYPTOGRAPHY** which states The TOE shall use NIST FIPS 140-2 validated cryptographic services provided by the IT Environment. **OE.CRYPTOGRAPHY** satisfies this policy by requiring the IT Environment to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity services as required by the IT Environment and the TOE.

TE.AUDIT_COMPROMISE states that a user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action. This threat is mapped to:

- **OE.AUDIT_PROTECTION** which states that the IT Environment will provide the capability to protect audit information. **OE.AUDIT_PROTECT** contributes to mitigating this threat by controlling access to the audit trail. Only an administrator is allowed to read the audit trail, no one is allowed to modify audit records, the administrator is the only one allowed to delete the audit trail, and the IT Environment has the capability to prevent auditable actions from occurring if the audit trail is full.
- **OE.RESIDUAL_INFORMATION** which states that the IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **OE.RESIDUAL_INFORMATION** prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a resource (e.g., memory). By ensuring the IT Environment prevents residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data.
- **OE.SELF_PROTECTION** which states that the IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure. **OE.SELF_PROTECTION** contributes to countering this threat by ensuring that the IT Environment can protect itself from users. If the IT Environment could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the audit trail which are always invoked is also critical to the migration of this threat.
- **OE.TOE_PROTECTION** which states The IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification. **OE.TOE_PROTECTION** contributes to countering this threat by

ensuring that the IT Environment can protect TOE. If the TOE could not be protected, it could not be trusted to provide accurate audit information.

TE.CHANGE_TIME states that an unauthorized user may change the TSF notion of time resulting in accepting old revocation information or expired certificates. This threat is mapped to:

- **OE.TIME_TOE** which states that the IT Environment will provide reliable time for the TOE use. **OE.TIME_TOE** protects against this threat by ensuring that the IT Environment does not permit users to change the time.

TE.CRYPTO_COMPROMISE states that a user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms. This threat is mapped to:

- **OE.CRYPTOGRAPHY** which states that the TOE shall use NIST FIPS 140-2 validated cryptographic services provided by the IT Environment. **OE.CRYPTOGRAPHY** protects against this threat by ensuring that the cryptography used is sound and has been validated.
- **OE.PHYSICAL** which states that the non-IT environment will 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** contributes to protection against this threat by providing physical protection from side channel attacks protects against the attempts to compromise the cryptographic mechanisms.

TE.MASQUERADE states that a user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources. This threat is mapped to:

- **OE.TOE_ACCESS** which states that the IT Environment will provide mechanisms that control a user's logical access to the TOE. **OE.TOE_ACCESS** mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.

TE.POOR_TEST states that lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities. This threat is mapped to:

- **OE.CORRECT_TSF_OPERATION** which states that the IT environment will provide functionality to support the correct operation of the TSF and provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. **OE.CORRECT_TSF_OPERATION** ensures that the underlying OS satisfies the additional requirements placed by the PP and that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF can be demonstrated, and thus providing end users the confidence that the TOE's security policies continue to be enforced.

TE.RESIDUAL_DATA states that a user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another. This threat is mapped to:

- **OE.RESIDUAL_INFORMATION** which states that the IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **OE.RESIDUAL_INFORMATION** counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.

TE.TSF_COMPROMISE states that a user or process may cause, through an unsophisticated attack, TSF data, security attributes, or executable code to be inappropriately accessed (viewed, modified, or deleted). This threat is mapped to:

- **OE.RESIDUAL_INFORMATION** which states that the IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **OE.RESIDUAL_INFORMATION** is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data
- **OE.SELF_PROTECTION** which states that the IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure. **OE.SELF_PROTECTION** is necessary to mitigate this threat to provide the TOE a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. This feature in turn ensures that other processes can not interfere with the IT Environment and defeat the IT Environment mechanisms.
- **OE.TOE_PROTECTION** which states that the IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification. **OE.TOE_PROTECTION** is necessary to mitigate this threat by ensuring that the IT Environment will protect the TOE. This feature ensures that other processes can not defeat the TOE protection mechanisms.
- **OE.MANAGE** which states that the IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. **OE.MANAGE** is necessary because an access control policy is not specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions

TE.UNATTENDED_SESSION states that a user may gain unauthorized access to an unattended session. This threat is mapped to:

- **OE.TOE_ACCESS** which states that the IT Environment will provide mechanisms that control a user's logical access to the TOE. **OE.TOE_ACCESS** helps to mitigate this threat by including mechanisms that place controls on user's sessions. User and administrator's sessions are locked. Locking the session reduces the opportunity of someone gaining unauthorized access the session when the console is unattended.

TE.UNAUTHORIZED_ACCESS states that a user may gain access to user data for which they are not authorized according to the TOE security policy. This threat is mapped to:

- **OE.MEDIATE** which states that the IT Environment will protect user data in accordance with its security policy. **OE.MEDIATE** ensures that all accesses to user data are subject to mediation, unless said data has been specifically identified as public data**.** The TOE requires successful authentication prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to successfully conduct a man-in-the-middle and/or password guessing attack is greatly reduced. Lastly, the IT Environment will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The IT Environment restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the Administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.

TE.UNIDENTIFIED_ACTIONS states that the administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach. This threat is mapped to:

- **OE.AUDIT_REVIEW** which states that the IT Environment will provide the capability to selectively view audit information. **OE.AUDIT_REVIEW** helps to mitigate this threat by providing the Administrator with a required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events, the IT Environment monitors the occurrences of these events (e.g. set number of authentication failures, set number of information policy flow failures, self-test failures, etc.).
- **OE.AUDIT_GENERATION** which states that the IT Environment will provide the capability to detect and create records of security-relevant events associated with users. **OE.AUDIT_GENERATION** helps to mitigate this threat by recording actions for later review
- **OE.TIME_STAMPS** which states that the IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. **OE.TIME_STAMPS** helps to mitigate this threat by ensuring that audit records have correct timestamps.
- **OE.TIME_TOE** which states that the IT Environment will provide reliable time for the TOE use. **OE.TIME_STAMPS** plays a role in supporting this policy by permitting the TOE to provide reliable time on audit records generated by the TOE.

In Table 8.2, the Base 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.

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

Objective	Threats, Assumption or OSP
OE.AUDIT_GENERATION	P.ACCOUNTABILITY; TE.UNIDENTIFIED_ACTIONS

Objective	Threats, Assumption or OSP
OE.AUDIT_PROTECTION	TE.AUDIT_COMPROMISE
OE.AUDIT_REVIEW	TE.UNIDENTIFIED_ACTIONS
OE.Configuration	AE.Configuration
OE.CORRECT_TSF_OPERATION	TE.POOR_TEST
OE.CRYPTOGRAPHY	P.CRYPTOGRAPHY; TE.CRYPTO_COMPROMISE
OE.DISPLAY_BANNER	P.ACCESS_BANNER
OE.Low	AE.Low
OE.MANAGE	TE.TSF_COMPROMISE
OE.MEDIATE	TE.UNAUTHORIZED_ACCESS
OE.NO_EVIL	AE.NO_EVIL
OE.PHYSICAL	AE.PHYSICAL. TE.CRYPTO_COMPROMISE
OE.RESIDUAL_INFORMATION	TE.AUDIT_COMPROMISE; TE.RESIDUAL_DATA; TE.TSF_COMPROMISE
OE.SELF_PROTECTION	TE.AUDIT_COMPROMISE; TE.TSF_COMPROMISE
OE.TIME_STAMPS	P.ACCOUNTABILITY; TE.UNIDENTIFIED_ACTIONS
OE.TIME_TOE	P.ACCOUNTABILITY; TE.CHANGE_TIME; TE.UNIDENTIFIED_ACTIONS
OE.TOE_ACCESS	P.ACCOUNTABILITY; TE.MASQUERADE; TE.UNATTENDED_SESSION
OE.TOE_PROTECTION	TE.AUDIT_COMPROMISE; TE.TSF_COMPROMISE
Objectives	**Threats, Assumption or OSP**
OE.AUDIT_GENERATION	P.ACCOUNTABILITY; TE.UNIDENTIFIED_ACTIONS
OE.AUDIT_PROTECTION	TE.AUDIT_COMPROMISE
OE.AUDIT_REVIEW	TE.UNIDENTIFIED_ACTIONS
OE.Configuration	AE.Configuration
OE.CORRECT_TSF_OPERATION	TE.POOR_TEST
OE.CRYPTOGRAPHY	P.CRYPTOGRAPHY; TE.CRYPTO_COMPROMISE
OE.DISPLAY_BANNER	P.ACCESS_BANNER
OE.Low	AE.Low
OE.MANAGE	TE.TSF_COMPROMISE
OE.MEDIATE	TE.UNAUTHORIZED_ACCESS

Objective	Threats, Assumption or OSP
OE.NO_EVIL	AE.NO_EVIL
OE.PHYSICAL	AE.PHYSICAL. TE.CRYPTO_COMPROMISE
OE.RESIDUAL_INFORMATION	TE.AUDIT_COMPROMISE; TE.RESIDUAL_DATA; TE.TSF_COMPROMISE
OE.SELF_PROTECTION	TE.AUDIT_COMPROMISE; TE.TSF_COMPROMISE
OE.TIME_STAMPS	P.ACCOUNTABILITY; TE.UNIDENTIFIED_ACTIONS
OE.TIME_TOE	P.ACCOUNTABILITY; TE.CHANGE_TIME; TE.UNIDENTIFIED_ACTIONS
OE.TOE_ACCESS	P.ACCOUNTABILITY; TE.MASQUERADE; TE.UNATTENDED_SESSION
OE.TOE_PROTECTION	TE.AUDIT_COMPROMISE; TE.TSF_COMPROMISE
OE.GOOD_USER	AE.GOOD_USER

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.Untrusted_CA	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.Untrusted_CA 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.

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 or accept responses that are within the user defined time range. Thus, even if revocation information or OCSP Responder are not available, previously generated information can be used or in the worst case completely override the time check..

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 or accept CRLs that are generated within user acceptable time limits. Thus, even if CRLs are not available, previously generated information can be used or in the worst case completely override the time check.

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

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.Untrusted_CA
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 and extended security assurance requirements from Section 5.4 can be used to test the explicitly stated requirements.

Table 8.5 – Security Objective to Functional Component Mapping

#	Objective	Functional Components
Mapping for Objectives for the Environment		
1E	OE.AUDIT_GENERATION	FAU_GEN.1-NIAP-0407:1; FAU_GEN.2-NIAP-0410:1; FIA_USB.1; FAU_SEL.1-NIAP-0407
12E	OE.AUDIT_PROTECTION	FAU_SAR.2; FAU_STG.1-NIAP-0429; FAU_STG.NIAP-0429-1; FMT_MOF.1
3E	OE.AUDIT_REVIEW	FAU_SAR.1; FAU_SAR.3
4E	OE.Configuration	Defined in startup and installation guides under ADO_IGS.1
5E	OE.CORRECT_TSF_OPERATION	FPT_TST_SOF_(EXP).1; ATE_COV.2; ATE_DPT.1; ATE_IND.2; ASE_PPC_(EXP).2; ATE_FUN_(EXP).3; ADV_HLD_(EXP).6
6E	OE.CRYPTOGRAPHY	FCS_CRM_FPS_(EXP).1; ADV_HLD_(EXP).6
7E	OE.DISPLAY_BANNER	FTA_TAB.1
8E	OE.Low	Defined in the SOF analysis and vulnerability assessment.
9E	OE.MANAGE	FMT_MOF.1; FMT_MSA.1; FMT_MSA.3-NIAP-0429; FMT_MTD.1:1; FMT_MTD.1:2; FMT_MTD.1:3; FMT_MTD.1:4; FMT_MTD.1:5; FMT_SMF.1, FMT_SMR.1
10E	OE.MEDIATE	FDP_ACC.1; FDP_ACF.1-NIAP-0407
11E	OE.NO_EVIL	Defined in the Administrator Guide under AGD_ADM.1 and AGD_USR.1
12E	OE.PHYSICAL	Defined as part of the physical security policy in AGD_ADM.1 and AGD_USR.1
13E	OE.RESIDUAL_INFORMATION	FDP_RIP.2

14E	OE.SELF_PROTECTION	FPT_SEP.1; FPT_RVM.1
15E	OE.TIME_STAMPS	FPT_STM.1, FMT_SMF.1, FMT_MTD.1:5
16E	OE.TIME_TOE	FPT_STM.1
17E	OE.TOE_ACCESS	FIA_AFL.1; FIA_ATD.1; FIA_UID.2; FIA_UAU.2; FIA_UAU.7; FTA_SSL.1; FTA_SSL.2
18E	OE.TOE_PROTECTION	FPT_SEP_ENV_(EXP).1; ADV_HLD_(EXP).6
19E	OE.GOOD.USER	Defined in the Users Guide under AGD_USR.1
Mapping for 1. CPV – Basic Package		
1	O.Availability	FDP_DAU_CPV_(EXP).1
2	O.Correct_Temporal	FDP_DAU_CPI_(EXP).1
3	O.Current_Certificate	FDP_DAU_CPV_(EXP).1
3	O.Get_KeyInfo	FDP_DAU_CPO_(EXP).1
5	O.Path_Find	FDP_CPD_(EXP).1
6	O.Trusted_Keys	FDP_DAU_CPI_(EXP).1
7	O.User	FDP_DAU_CPV_(EXP).2
8	O.Verified_Certificate	FDP_DAU_CPV_(EXP).1
9	O.Valid_Certificate	FDP_DAU_CPV_(EXP).1
Mapping for 2. CPV – Basic Policy Package		
10	O.Provide_Policy_Info	FDP_DAU_CPI_(EXP).2, FDP_DAU_CPO_(EXP).2
Mapping for 3. CPV – Policy Mapping Package		
11	O.Map_Policies	FDP_DAU_CPI_(EXP).3, FDP_DAU_CPV_(EXP).3, FDP_DAU_CPO_(EXP).3
12	O.Policy_Enforce	FDP_DAU_CPI_(EXP).3, FDP_DAU_CPV_(EXP).3, FDP_DAU_CPO_(EXP).3
Mapping for 4. CPV – Name Constraints Package		
13	O.Authorised_Names	FDP_DAU_CPI_(EXP).4, FDP_DAU_CPV_(EXP).4, FDP_DAU_CPV_(EXP).5
Mapping for 5. PKI Signature Generation Package		
14	O.Give_Sig_Hints	FDP_ETC_SIG_(EXP).1
Mapping for 6. PKI Signature Verification Package		
15	O.Use_Sig_Hints	FDP_ITC_SIG_(EXP).1,
16	O.Linkage_Sig_Ver	FDP_DAU_SIG_(EXP).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_(EXP).1
18	O.Linkage_Enc_WO	FDP_ETC_ENC_(EXP).1, FDP_DAU_ENC_(EXP).1
Mapping for 8. PKI Decryption using Key Transfer Algorithms Package		
19	O.Correct_KT	FDP_ITC_ENC_(EXP).1
Mapping for 9. OCSP Client Package		
20	O.Accurate_OCSP_Info	FDP_DAU_OCS_(EXP).1
21	O.Auth_OCSP_Info	FDP_DAU_OCS_(EXP).1
22	O.Current_OCSP_Info	FDP_DAU_OCS_(EXP).1
23	O.User_Override_Time_OCSP	FDP_DAU_OCS_(EXP).1
Mapping for 10. Certificate Revocation List (CRL) Validation Package		
24	O.Accurate_Rev_Info	FDP_DAU_CRL_(EXP).1
25	O.Auth_Rev_Info	FDP_DAU_CRL_(EXP).1
26	O.Current_Rev_Info	FDP_DAU_CRL_(EXP).1
27	O.User_Override_Time_CRL	FDP_DAU_CRL_(EXP).1

8.2.1.1 Security Objectives for the IT Environment Rationale

Security Objectives for the Environment are met through a set of assumptions, as defined in Section 3.1 of this ST, 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.AUDIT_GENERATION state that the IT Environment will provide the capability to detect and create records of security-relevant events associated with users. This objective is satisfied by the following requirements:

- **FAU_GEN.1-NIAP-0407:1**defines the set of events that the IT Environment must be capable of recording. This requirement ensures that the Administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds. In this ST, the ST author has not added any security functional requirements over and above those in the PP and hence no new audit events have been identified.
- **FAU_GEN.2-NIAP-0410:1** ensures that the audit records associate a user identity with the auditable event.
- **FIA_USB.1** plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the IT Environment. This only applies to authorized users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail

may not always have the proper identity of the subject that causes an audit record to be generated.

- **FAU_SEL.1-NIAP-0407** allows the Administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism

OE.AUDIT_PROTECTION states that the IT Environment will provide the capability to protect audit information. This objective is satisfied by the following requirements:

- **FAU_SAR.2** restricts the ability to read the audit trail to the Administrator, thus preventing the disclosure of the audit data to any other user. However, the IT Environment is not expected to prevent the disclosure of audit data if it has been archived or saved in another form (e.g., moved or copied to an ordinary file).
- **FAU_STG.1-NIAP-0429; FAU_STG.NIAP-0429-1:** The **FAU_STG** family dictates how the audit trail is protected. **FAU_STG.1-NIAP-0429** restricts the ability to delete audit records to the administrator. **FAU_STG.NIAP-0429-1** defines the actions that must be available to the administrator, as well as the action to be taken if there is no response. This helps to ensure that audit records are kept until the administrator deems they are no longer necessary. This requirement also ensures that no one has the ability to modify audit records (e.g., edit any of the information contained in an audit record). This ensures the integrity of the audit trail is maintained.
- **FMT_MOF.1** restricts the capability to modify the behavior of the audit function to the administrator. This requirement ensures that only administrator can turn audit on or off, this ensuring users actions are audited according to a site defined policy.

OE.AUDIT_REVIEW states that the IT Environment will provide the capability to selectively view audit information. This objective is satisfied by the following requirements:

- **FAU_SAR.1** provides the administrator with the capability to read all the audit data contained in the audit trail. This requirement also mandates the audit information be presented in a manner that is suitable for the administrator to interpret the audit trail, which is subject to interpretation. It is expected that the audit information be presented in such a way that the administrator can examine an audit record and have the appropriate information (that required by FAU_GEN.2) presented together to facilitate the analysis of the audit review
- **FAU_SAR.3** complements FAU_SAR.1 by providing the administrator the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3 requires the administrator be able to establish the audit review criteria based on a user ID and source subject identity, so that the actions of a user can be readily identified and analyzed.

OE.Configuration states that the TOE shall be installed and configured properly for starting up the TOE in a secure state. This objective covers A.Configuration, an assumption that states that the TOE will be properly installed and configured. This objective is 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.CORRECT_TSF_OPERATION states that the IT environment will provide functionality to support the correct operation of the TSF and capability to test the TSF to ensure the correct operation of the TSF at a customer's site.

- **FPT_TST_SOF_(EXP).1** is necessary to ensure the correctness of the TSF configuration files and TSF data and executable. If TSF software is corrupted it is possible that the TSF would no longer be able to enforce the security policies. This also holds true for TSF data, if TSF data is corrupted, the TOE may not correctly enforce its security policies.
- **ADV_HLD_(EXP).6** is nessary to demonstrate what cryptographic mechanism is used to check the integrity of the TOE. The cryptographic mechanism much be NIST FIPS 140-2 validated.
- **ATE** security assurance requirements will provide assurance that the TOE has been thoroughly tested to ensure the correct operation of the TSF. Work units for ATE_COV.2, ATE_DPT.1, ATE_IND.2 will demonstrate that the TOE testing contained enough depth and coverage to test TOE TSF functionality.
- **ASE_PPC_(EXP).2** assurance requirement will provide assurance that the underlying Operating System ST satisfies additional requirements levied by the PP.
- ATE_FUN_(EXP).3 assurance requirement will provide assurance that the TOE has been thoroughly tested to ensure the correct operation of the TSF. It will also provide assurance that additional requirements levied by the PP have been satisfied.

  Addition testing requirements levied by the PP:

  - FIA_AFL.1 Authentication failure handling testing rational

  To test authentication failure handling, the OS will be configured to lock the account after specified number of unsuccessful login attempts. The use will attempt to login using incorrect login information. The account should be locked after the specified number of login attempts is reached. If the test is successful it will demonstrate that OS supports authentication failure handling. After verifying that authentication failure handling worked review the log files do determine if reaching of the threshold for the unsuccessful authentication attempts was recorded. If the record is present it will satisfy the audit requirement for FIA_AFL.1 because it will demonstrate that the operating system supports the ability to set the limit of unsuccessful authentication attempts. FIA_AFL.1 requires that the number of unsuccessful attempts can be set and when that number is reach the account should be locked. Authentication failure handling test satisfy both aspects of the requirement, the number of unsuccessful authentication attempts and account lockout when that number is reached.

  - FMT_MOF.1 Management of security function behavior testing rational

  To test management of security function behavior an unprivileged user will attempt to disable, enable, and modify audit functionality. The unprivileged user should not be successful in disabling, enabling, or modifying audit functionality. If the test is successful it will demonstrate that OS supports management of security function behavior, because only a privileged user (administrator) can disable, enable, or modify audit functionality. FMT_MOF.1 requires that the ability to disable, enable, and modify audit functionality shall be restricted to administrators. Management of security functional behavior test satisfies the requirements for FMT_MOF.1 by

demonstrating that only administrators have the rights to disable, enable, and modify audit functionality.

- FMT_MTD.1:1 Management of TSF data – I&A Data testing rational

To test management of I&A data an unprivileged user will attempt to add, remove, edit user accounts. The unprivileged user should not be successful in changing any aspects of user account information. If the test is successful it will demonstrate that OS supports management of I&A data. FMT_MTD.1:1 requires that the OS restrict the ability to initialize and modify identification and authentication data to the administrator. Management of TSF data – I&A data test satisfies this requirement by demonstrating that only an administrator has the rights to add, remove, edit user accounts.

- FMT_MTD.1:3 Management of TSF data – I&A Attempts testing rational

To test management of I&A attempts an unprivileged user will attempt to change the number of unsuccessful attempts needed to lock an account. The unprivileged user should be unsuccessful in changing that number. If the test is successful it will demonstrate that OS supports management of I&A attempts. FMT_MTD.1:3 requires that the initialization of modification of the number of unsuccessful authentication attempts shall be restricted to an administrator. Management of TSF data –I&A attempts test satisfies the requirements for FMT_MTD.1:1 by demonstrating that only an administrator can initialize or modify the number of unsuccessful authentication attempts.

- FMT_MTD.1:4 Management of TSF data – Trust Anchors testing rational

To test management of I&A trust anchor an unprivileged user will attempt to add/remove trust anchors to the certificate database. The unprivileged user should be unsuccessful in adding or removing trust anchors. If the test is successful it will demonstrate that OS supports management of trust anchors. FMT_MTD.1:4 requires that the addition and deleting of trust anchor shall be restricted to users. The Management of TSF data – trust anchors test satisfies the FMT_MTD.1:4 requirement by demonstrating that only authorized users have the rights to add and delete trust anchors.

- FMT_MTD.1:5 Management of TSF data – Time testing rational

To test management of I&A time an unprivileged user will attempt to change system time. The unprivileged user should be unsuccessful in changing system time. If the test is successful it will demonstrate that OS supports management of time. FMT_MTD.1:5 requires that the initialization and modification of system time shall be restricted to administrators. The Management of TSF data – time test satisfies the FMT_MTD.1:5 requirement by demonstrating that only administrators have the rights to modify system time.

- FTA_SSL.1 TSF-initiated session locking testing rational

To test TSF-initiated session locking, the OS will be configured to lock the session after a period of inactivity. The test is successful if after the specified time the session is locked, the display is clear, no activity is permitted, and only the user whose session was running and an administrator is able to unlock the session. If the test is successful it will demonstrate that OS supports TSF-initiated session locking. The FTA_SSL.1 requires that a session to be locked after a period of time, making the current contents of the session unreadable and disabling any activity of user's data access/display devices other then unlocking the session. The TSF-

initiated session locking test satisfies the requirement for TSF-initiated session locking by demonstrating that the session was locked after a specified time period, the display was cleared (does not display any current contents), and not activity is permitted other then unlocking the session.

- FTA_SSL.2 User-initiated session locking testing rational

To test User-initiated session locking, user will lock the session. The test is successful if the session is locked and only the user whose session was running and an administrator is able to unlock the session. If the test is successful it will demonstrate that OS supports User-initiated session locking. The FTA_SSL.1 requires that a session can be locked by a user, making the current contents of the session unreadable and disabling any activity of user's data access/display devices other then unlocking the session. The TSF-initiated session locking test satisfies the requirement for TSF-initiated session locking by demonstrating that the session was locked by the user, the display was cleared (does not display any current contents), and no activity is permitted other then unlocking the session.

- FTA_TAB.1 Default TOE access banners testing rational

To test default TOE access banners, the OS will be configured to display a banner when logging into the system. The test will be successful if the banner is displayed when a user is logging into the system. If the test is successful it will demonstrate that OS supports default TOE access banners. FTA_TAB.1 requires that before establishing a user session a warning message shall be displayed regarding the unauthorized use of the system. The default TOE access banner test satisfies the requirement for default TOE access banner by demonstrating that and OS can be configured to display desired message regarding unauthorized use of the system.

- Testing rational for FAU_SEL.1-NIAP-0407

To test the audit requirement for FAU_SEL.1-NIAP-0407 the audit configuration will be modified. The security log will be reviewed to verify that the changes to the audit configuration are noted in the audit record, identifying the identity of the Security Administrator performing the configuration change. If the record is present and the identity of the Security Administrator is identified it will demonstrate the support for audit requirement for FAU_SEL.1-NIAP-0407. The FAU_SEL.1-NIAP-0407 audit requirement requires that the audit record show all modifications to the audit configuration and that the identity of the security administrator performing the function will be identified in the log. The test for FAU_SEL.1-NIAP-0407 audit requirement satisfies this requirement by showing changes to the audit configuration in the log and by identifying the identity of the administrator performing the function.

OE.CRYPTOGRAPHY states that the TOE shall use NIST FIPS 140-2 validated cryptographic services provided by the IT Environment. This objective is satisfied by the following requirements:

- **FCS_CRM_FPS_(EXP).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.

- **ADV_HLD_(EXP).6** helps to meet this objective by describing the cryptographic module provided by the IT environment. This description will help to verify FIPS 140 series Level 1 validation.

OE.DISPLAY_BANNER states that the IT Environment will display an advisory warning regarding use of the TOE. This objective is satisfied by the following requirements:

- **FTA_TAB.1** meets this objective by requiring the IT Environment to display an administrator defined banner before a user can establish an authenticated session. This banner is under complete control of the administrator in which they specify any warnings regarding unauthorized use of the TOE and remove any product or version information if they desire.

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 objective covers the SOF analysis, which analyzes the strength of function of identification and authentication functions.

OE.MANAGE states that the IT Environment will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. This objective is satisfied by the following requirements:

- **FMT_MOF.1** requires that the ability to use particular TOE capabilities be restricted to the Administrator.
- **FMT_MSA.1** requires that the ability to perform operations on security attributes be restricted to particular roles.
- **FMT_MSA.3-NIAP-0429** requires that default values used for security attributes are restrictive, and that the Administrator has the ability to override those values.
- **FMT_MTD.1:1, FMT_MTD.1:2, FMT_MTD.1:3, FMT_MTD.1:4, and FMT_MFT.1:5** require that the ability to manipulate IT Environment and TOE data is restricted to Administrators and authorized users.
- **FMT_SMF.1** requires that appropriate administrators manage the audit and other functions.
- **FMT_SMR.1** defines the specific security roles to be supported to perform the functions listed in the list above.

OE.MEDIATE states that the IT Environment will protect user data in accordance with its security policy. This objective is satisfied by the following requirements:

- **FDP_ACC.1** defines that an Access Control policy that will be enforced on a list of subjects acting on the behalf of users attempting to gain access to a list of named objects. All the operations between subject and object covered are defined by the policy. The “subjects” are generally the IT Environment's “Agents.” The “named objects” are things that the IT Environment is protecting for itself and for the TOE
- **FDP_ACF.1-NIAP-0407** defines the Security Attribute used to provide Access Control to objects based on the following above Access Control policy and access control rules based on those security attributes.

OE.NO_EVIL states that sites using the TOE will ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance. This objective is supported by:

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

OE.GOOD_USER states that sites using the TOE will ensure that TOE users are non-hostile and follow all user guidance. This objective is supported by:

- The User Guides as defined under assurance requirement AGD_USR.1.

OE.PHYSICAL states that the non-IT environment will 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 objective is 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.RESIDUAL_INFORMATION which states that the IT Environment will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. This objective is satisfied by the following requirements:

- **FDP_RIP.2** is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.

OE.SELF_PROTECTION which states that the IT Environment will maintain a domain for its own execution that protects it and its resources from external interference, tampering, or unauthorized disclosure. This objective is satisfied by the following requirements:

- **FPT_RVM.1** ensures that the IT Environment makes policy decisions on all interfaces that perform operations on subjects and objects that are scoped by the policies. Without this non-bypassability requirement, the IT Environment could not be relied upon to completely enforce the security policies, since an interface(s) may otherwise exist that would provide a user with access to TOE resources (including TSF data and executable code) regardless of the defined policies. This includes controlling the accessibility to interfaces, as well as what access control is provided within the interfaces.
- **FPT_SEP.1** was chosen to ensure the IT Environment provides a domain that protects itself from untrusted users. If the IT Environment cannot protect itself it cannot be relied upon to enforce its security policies.

OE.TIME_STAMPS states that the IT Environment will provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. This objective is satisfied by the following requirements:

- **FPT_STM.1** requires that the IT Environment provide time stamps for its own use and for the TOE use.
- **FMT_SMF.1** requires that the IT Environment provide an administrator with the capability to modify system time.
- **FMT_MTD.1:5** requires that the IT Environment restrict the capability to modify system time to an administrator.

OE.TIME_TOE states that The IT Environment will provide reliable time for the TOE use. This objective is satisfied by the following requirements:

- **FPT_STM.1** requires that the IT Environment provide time stamps for its own use and for the TOE use.

OE.TOE_ACCESS states that the IT Environment will provide mechanisms that control a user's logical access to the TOE. This objective is satisfied by the following requirements:

- **FIA_AFL.1** provides a detection mechanism for unsuccessful authentication attempts by the users. The requirement enables an administrator settable threshold that prevents unauthorized users from gaining access to authorized user's account by guessing authentication data by locking the targeted account. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE.
- **FIA_ATD.1** defines the attributes of users, including a user ID that is used to by the IT Environment to determine a user's identity and enforce what type of access the user has to the IT Environment (e.g., the IT Environment associates a user ID with any role(s) they may assume).
- **FIA_UID.2** requires that a user be identified to the IT Environment in order to do anything.
- **FIA_UAU.2** requires that a user be authenticated by the IT Environment in order to do anything.
- **FIA_UAU.7** provides that the authentication data provided by the user is not echoed back in plaintext, thus serving to protect that data.
- **FTA_SSL.1 and FTA_SSL.2** components deal with automatic session locking and termination, either initiated by the IT Environment or a user. They protect from an unauthorized entity to use the unattended session.

OE.TOE_PROTECTION states that the IT Environment will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification. This objective is satisfied by the following requirements:

- **FPT_SEP_ENV_(EXP).1** ensures that the IT Environment provides a domain that protects TSF from untrusted users. If the TSF cannot be protected, it cannot be relied upon to enforce its security policies.
- **ADV_HLD_(EXP).6** will describe how the underlying operating system provides discretionary access control (DAC). DAC and FPT_SEP.1 provided by the underlying OS will protect the TOE and TOE resources from external interference, tampering, or unauthorized disclosure and modification.

8.2.1.2 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 and user overrides revocation checking. This objective is met by:

- FDP_DAU_CPV_(EXP).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_CPI_(EXP).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_(EXP).1, which requires that the TSF accept a certificate only if the specified checks succeed, including that the certificate is not expired as of TOI.

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_CPO_(EXP).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 which includes: certificate, subject alternative names, extendedKeyUsage.

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_(EXP).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_CPI_(EXP).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_(EXP).2, Intermediate certificate processing – basic, which requires that the TSF accept an intermediate certificate only when 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_(EXP).1, Certificate processing – basic, which requires that the TSF accept certificates only with verifiable signatures.

O.Valid_Certificate states that the TSF shall use certificates that are valid, i.e., not revoked, unless user overrides revocation checking or certificate contains a no-check extension. This objective is met by:

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

8.2.1.3 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_CPI_(EXP).2, Certification path initialisation – basic policy, which requires that the TSF shall use the initial-certificate-policies provided by user role.
- FDP_DAU_CPO_(EXP).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.

8.2.1.4 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_CPI_(EXP).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 the user role.
- FDP_DAU_CPV_(EXP).3, Intermediate certificate processing – policy mapping, which requires that the TSF use the intermediate certificate to update specified state variables.
- FDP_DAU_CPO_(EXP).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_CPI_(EXP).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 the user role.
- FDP_DAU_CPV_(EXP).3, Intermediate certificate processing – policy mapping, which requires that the TSF use the intermediate certificate to update specified state variables.
- FDP_DAU_CPO_(EXP).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.

8.2.1.5 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_CPI_(EXP).4, Certification path initialisation – names, which requires that the TSF initialize the following: permitted-subtrees = ∞, excluded-subtrees = ∅
- FDP_DAU_CPV_(EXP).4, Intermediate certificate processing – name constraints, which requires that the TSF accept a certificate only if the conditions specified by the requirement, including verification of authorization, is satisfied.
- FDP_DAU_CPV_(EXP).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

8.2.1.6 PKI Signature Generation Package Rationale

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

- FDP_ETC_SIG_(EXP).1 Export of PKI Signature, which requires that the TSF use the user selected private to key perform digital signature and that the TSF include additional information specified by the ST author with the digital signature to facilitate

signature verification. This additional information includes: hashing algorithm, and signature algorithm.

8.2.1.7 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_(EXP).1, Import of PKI Signature, which requires that the TSF use the following information from the signed data: hashing algorithm and signature algorithm 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_(EXP).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 shall verify that the following additional checks are made: the keyUsage extension output from the Certification Path Validation has the *nonRepudiation or digitalSignature* bit set.

8.2.1.8 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_(EXP).1, Export of PKI Encryption – Key Transfer Algorithms, which requires that the TSF include the following information with the encrypted data: key encryption algorithm, data encryption algorithm, decryptor key identifier. The TSF shall invoke a cryptographic module with the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, and 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_(EXP).1, Export of PKI Encryption – Key Transfer Algorithms, which requires that the TSF include the following information with the encrypted data: key encryption algorithm, data encryption algorithm, decryptor key identifier. The TSF shall invoke a cryptographic module with the following information from Certification Path Validation to create encrypted data: subject public key algorithm, subject public key, and subject public key parameters.
- FDP_DAU_ENC_(EXP).1, PKI Encryption Verification – Key Transfer, which requires that the TSF apply verification checks for key transfer that the keyUsage output from Certification Path Validation contains keyEncipherment bit set.

8.2.1.9 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_(EXP).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. The ST author has selected the following information for this function: key encryption algorithm, data encryption algorithm, and decryptor key identifier.

8.2.1.10 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_(EXP).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_(EXP).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_(EXP).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_(EXP).1, Basic OCSP Client, which requires that a user role be able to override the time checks on the OCSP response.

8.2.1.11 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_(EXP).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 explicitly stated requirement.

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

- FDP_DAU_CRL_(EXP).1, Basic CRL checking, which requires that the TSF accept revocation information local cache, repository, location pointed to by the CRL DP in public key certificate of interest, or user.

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

- FDP_DAU_CRL_(EXP).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_(EXP).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_(EXP).1, Basic CRL checking, which requires that the TSF accept the CRL as current if user role overrides time checks.

8.2.2 Assurance Requirement Rationale

EAL 4 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 behavior. Assurance is additionally gained through an informal model of the TOE security policy. EAL 4 represents a meaningful assurance by requiring design description, a subset of the implementation, and mechanisms and/or procedures that provide confidence that the TOE will not be tampered with during development or delivery. EAL 4 is augmented with ALC_FLR.2 to track and correct the reported and found security flaws in the product and also to provide flaw reporting procedures to the product users.

Due to the requirements levied by the PKE Protection Profile, it was necessary to define three extended security assurance requirements in section 5.4.

ASE_PPC_(EXP).2 addresses the need to verify the requirements placed on underlying Operating System Security Target by the PKE Protection Profile.

ADV_HLD_(EXP).6 addreses the requirements placed by the PKE Protection Profile on meeting the FCS_CRM_FSP_(EXP).1, FPT_SEP_ENV_(EXP).1, FTP_TST_SOF_(EXP).1 security functional requirements.

ATE_FUN_(EXP).3 addresses the need to satisfy security functional requirements which were left unsatisfied by the underlying Operating System Security Target. These SFRs will be met by testing as specified in the PKE Protection Profile.

8.2.3 Strength of Function Rationale

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

8.2.4 Security Functional Requirements Dependencies Rationale

Table 8.6 – Functional Requirements Dependencies

Requirement	Dependencies
IT Environment	
FAU_GEN.1-NIAP-0407:1	FPT_STM.1
FAU_GEN.2-NIAP-0410:1	FAU_GEN.1 (met by FAU_GEN.1-NIAP-0407:1) FIA_UID.1 (met by FIA_UID.2)
FAU_SAR.1	FAU_GEN.1 (met by FAU_GEN.1-NIAP-0407:1)
FAU_SAR.2	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1
FAU_SEL.1-NIAP-0407	FAU_GEN.1 (met by FAU_GEN.1-NIAP-0407:1) FMT_MTD.1
FAU_STG.1-NIAP-0429	FAU_GEN.1 (met by FAU_GEN.1-NIAP-0407:1)
FAU_STG.NIAP-0429-1	FAU_STG.1 (met by FAU_STG.1-NIAP-0429) FMT_MTD.1
FCS_CRM_FPS_(EXP).1	None
FDP_ACC.1	FDP_ACF.1 (met by FDP_ACF.1-NIAP-0407)

FDP_ACF.1-NIAP-0407	FDP_ACC.1 FMT_MSA.3 (met by FMT_MSA.3-NIAP-0429)
FDP_RIP.2	None
FIA_AFL.1	FIA_UAU.1 (met by FIA_UAU.2)
FIA_ATD.1	None
FIA_UAU.2	FIA_UID.1 (met by FIA_UID.2)
FIA_UAU.7	FIA_UAU.1 (met by FIA_UAU.2)
FIA_UID.2	None
FIA_USB.1	FIA_ATD.1
FMT_MOF.1	FMT_SMF.1; FMT_SMR.1
FMT_MSA.1	FMT_SMF.1; FMT_SMR.1 [FDP_ACC.1 Subset access control or FDP_IFC Subset information flow control] (satisfied by FDP_ACC.1)
FMT_MSA.3-NIAP-0429	FMT_MSA.1; FMT_SMR.1
FMT_MTD.1:1 through 5	FMT_SMF.1; FMT_SMR.1
FMT_SMF.1	None
FMT_SMR.1	FIA_UID.1 (met by FIA_UID.2)
FPT_RVM.1	None
FPT_SEP.1	None
FPT_SEP_ENV_(EXP).1	None
FPT_STM.1	None
FPT_TST_SOF_(EXP).1	None
FTA_SSL.1	FIA_UAU.1 (met by FIA_UAU.2)
FTA_SSL.2	FIA_UAU.1 (met by FIA_UAU.2)
FTA_TAB.1	None
CPV – Basic Package	
FDP_CPD_(EXP).1	None
FDP_DAU_CPI_(EXP).1	FCS_COP.1 (met by FCS_CRM_FPS_(EXP).1) FPT_STM.1
FDP_DAU_CPV_(EXP).1	FCS_COP.1 (met by FCS_CRM_FPS_(EXP).1) FPT_STM.1, [FDP_DAU_OCS_(EXP).1 or FDP_DAU_CRL_(EXP).1]
FDP_DAU_CPV_(EXP).2	FDP_DAU_CPV_(EXP).1
FDP_DAU_CPO_(EXP).1	FDP_DAU_CPV_(EXP).1
CPV – Basic Policy Package	
FDP_DAU_CPI_(EXP).2	FDP_DAU_CPI_(EXP).1 (See Note 1)
FDP_DAU_CPO_(EXP).2	FDP_DAU_CPO_(EXP).1 (See Note 1)
CPV – Policy Mapping Package	
FDP_DAU_CPI_(EXP).3	FDP_DAU_CPI_(EXP).2 (See Note 2)
FDP_DAU_CPV_(EXP).3	FDP_DAU_CPV_(EXP).2 (See Note 3)

FDP_DAU_CPO_(EXP).3	FDP_DAU_CPO_(EXP).2 (See Note 2)
CPV – Name Constraints Package	
FDP_DAU_CPI_(EXP).4	FDP_DAU_CPI_(EXP).1 (See Note 1)
FDP_DAU_CPV_(EXP).4	FDP_DAU_CPV_(EXP).1 (See Note 1)
FDP_DAU_CPV_(EXP).5	FDP_DAU_CPV_(EXP).2 (See Note 1)
PKI Signature Generation Package	
FDP_ETC_SIG_(EXP).1	FCS_CRM_FPS_(EXP).1
PKI Signature Verification Package	
FDP_ITC_SIG_(EXP).1	None
FDP_DAU_SIG_(EXP).1	FCS_CRM_FPS_(EXP).1 FDP_DAU_CPO_(EXP).1 (See Note 1)
PKI Encryption using Key Transfer Algorithms Package	
FDP_ETC_ENC_(EXP).1	FCS_CRM_FPS_(EXP).1 FDP_DAU_CPO_(EXP).1 (See Note 1)
FDP_DAU_ENC_(EXP).1	FDP_DAU_CPO_(EXP).1 (See Note 1)
PKI Decryption using Key Transfer Algorithms Package	
FDP_ITC_ENC_(EXP).1	FCS_CRM_FPS_(EXP).1
Online Certificate Status Protocol Client Package	
FDP_DAU_OCS_(EXP).1	FCS_CRM_FPS_(EXP).1 FPT_STM.1
Certificate Revocation List (CRL) Validation Package	
FDP_DAU_CRL_(EXP).1	FCS_CRM_FPS_(EXP).1 FPT_STM.1

Note 1: The dependency is satisfied by including the CPV – Basic Package

Note 2: The dependency is satisfied by including the CPV – Basic Policy Package

Note 3: 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_(EXP).1	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary

		Specification)
2	FDP_DAU_CPI_(EXP).1	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
3	FDP_DAU_CPV_(EXP).1	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
4	FDP_DAU_CPV_(EXP).2	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
5	FDP_DAU_CPO_(EXP).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_CPI_(EXP).2	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
7	FDP_DAU_CPO_(EXP).2	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the ST)
	3. CPV – Policy Mapping Package	
8	FDP_DAU_CPI_(EXP).3	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
9	FDP_DAU_CPV_(EXP).3	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
10	FDP_DAU_CPO_(EXP).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_CPI_(EXP).4	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
12	FDP_DAU_CPV_(EXP).4	Certification Path Processing, CRL Processing and OCSP Processing (See Section 6.1 in the TOE Summary Specification)
13	FDP_DAU_CPV_(EXP).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_(EXP).1	Signature Generation Functionality (See Section 6.2 in the TOE Summary Specification)
	6. PKI Signature Verification Package	
15	FDP_ITC_SIG_(EXP).1	PKI Signature Verification Functionality (See Section 6.3 in the TOE Summary Specification)
16	FDP_DAU_SIG_(EXP).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_(EXP).1	PKI Encryption using Key Transfer Algorithms Functionality (See Section 6.4 in the TOE Summary Specification)
18	FDP_DAU_ENC_(EXP).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_(EXP).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_(EXP).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_(EXP).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.

Curl

Curl is a tool for transferring files with URL syntax, supporting FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, FILE and LDAP. curl supports SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, cookies, user+password authentication (Basic, Digest, NTLM, Negotiate, kerberos...), file transfer resume, proxy tunneling and a busload of other

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.

Network Security Services

Network Security Services (NSS) is a set of libraries designed to support cross-platform development of security-enabled client and server applications.

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
GCC	GNU Compiler Collection
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
NSS	Network Security Services
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
TOI	Time of interest
TSC	**TSF** Scope of Control
TSF	**TOE** Security Function
TSP	Time Stamp Protocol (Internet X.509 Public Key Infrastructure)
USMC	United States Marine Corps